HPI Future SOC Lab: Proceedings 2015

Christoph Meinel, Andreas Polze, Gerhard Oswald,
Rolf Strotmann, Ulrich Seibold, Bernhard Schulzki (Eds.)

HPI Future SOC Lab:
Proceedings 2015

Christoph Meinel | Andreas Polze | Gerhard Oswald | Rolf Strotmann |
Ulrich Seibold | Bernhard Schulzki (Eds.)

HPI Future SOC Lab

Proceedings 2015

Hasso-Plattner-Institut für Softwaresystemtechnik GmbH

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de/ abrufbar.

Hasso-Plattner-Institut 2017
https://hpi.de/

Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam
Tel.: +49-(0)331 5509-0 7 / Fax: +49-(0)331 5509-325
E-Mail: hpi-info@hpi.de

Online veröffentlicht auf dem Publikationsserver der Universität Potsdam
URN urn:nbn:de:kobv:517-opus4-102516
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-102516

Contents

Solving of LP and CWLP Problems Using SAP HANA

Karl Kurbel
European University Viadrina
Große Scharrnstraße 59
15230 Frankfurt (Oder), Germany
kurbel.bi@europa-uni.de

Dawid Nowak
European University Viadrina
Große Scharrnstraße 59
15230 Frankfurt (Oder), Germany
danowak@europa-uni.de

Abstract

This paper explores the capabilities of SAP HANA for solving optimization problems, in particular linear programming and mixed-integer programming problems. The study contrasts a tightly integrated solution approach (GENIOS) with an external solver approach (R server) and with self-implemented optimization heuristics. All solution approaches are integrated into a test environment in HANA, and compared with respect to performance and solution quality. Based on a series of test cases, performance indicators are evaluated and factors influencing the performance are discussed.

1 Introduction and motivation

Today's business software, such as enterprise resource planning (ERP) and supply chain management (SCM) systems, provides solutions to many planning and decision problems [1]. A number of exact and heuristic optimization algorithms have been incorporated into the software (for example, mixed-integer linear programming for supply network planning in SAP SNP [2]).

Optimization models representing real-life situations can be very large, requiring a lot of computing power and efficient algorithms.

SAP HANA as an in-memory database exhibits very good performance where data access and data processing are concerned. Due to efficient techniques such as parallel processing and column-oriented data storage [3], SAP HANA outperforms typical databases [4] [5].

The questions we are investigating in this paper is how optimization models can be integrated into HANA and whether optimization can also benefit from HANA's processing power.

The next section discusses different approaches to solving optimization problems in HANA. Section 3 focuses on the solution architecture and the performance measures used for the test runs. The fourth section briefly describes the testing environment and the test cases used. Section 5 discusses the test results, while the concluding section wraps up the findings and gives an outlook to further research.

2 Solution approach

From release SPS 08 on, SAP HANA can be equipped with its own solver, GENIOS (GENeric Integer Optimization System) [6]. GENIOS is capable of solving continuous and mixed-integer linear programming (MILP) problems. It is a part of AFL (Application Function Libraries) [7, p. 6]), which extends HANA's functionality by predefined functions. These functions can be called by user-defined stored procedures. They are native to the data engine underlying SAP HANA's index server, thus offering the best possible performance for data access and data processing [8, p. 14].

Bearing in mind the architectural concept of SAP HANA [8, pp. 13-15] and the results of previous studies [9], two approaches seem to be worth trying in order to make a comparison with GENIOS.

Since HANA's power is exploited best when "everything" happens inside HANA, the first option is to create a solver for optimization problems with the means that HANA provides for software development. This means, in the first place, to use the native programming language, SQLScript [10].

The second option is a linear programming package that is available on R servers. Although an R server is external to HANA, procedures in R can be written and invoked inside HANA, just like SQLScript procedures [11]. An R server is a statistical server providing advanced calculation functionalities. These functionalities can be easily extended through downloadable function libraries and R scripts. One of them is the *lpSolve* package, which provides an application programming interface (API) for building and solving linear programs [12].

The types of problems we are considering in this study are *linear programming (LP)* and *mixed-integer linear programming (MILP)* problems. A representative of the latter type is the so-called *capacitated warehouse location problem (CWLP).* The CWLP embraces the

decision in which locations warehouses with limited capacities are to be built, and the decision which customers are to be served by which warehouses so as to satisfy all customers' demands while minimizing total fixed and transportation costs.

3 Solution architecture

For continuous linear programming problems, the solvers of GENIOS and *lpSolve* were used. Mixed-integer problems were narrowed down to the capacitated warehouse location problem and tackled with the help of GENIOS and self-implemented heuristics, namely *Add* and *Drop* [13]. These special heuristics for CWLP were implemented as stored procedures in HANA using SQLScript. An overview of the solution architecture is given in Figure 1.

3.1 Integration concept and functionality

Optimization models are usually stored in a common format (such as MPS [14]). Due to the fact that SAP HANA (SPS version 08) has no features to directly read such a format, it was necessary to write procedures for importing and parsing the problem data according to each of the solvers' requirements.

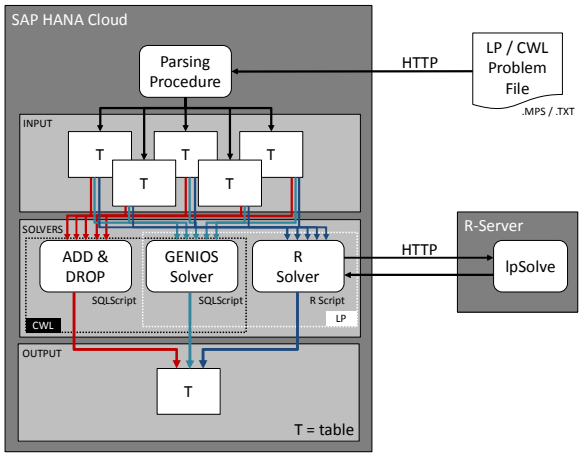


Figure 1. Solution architecture

One such procedure converts a textual input format (i.e., .mps, .txt) into a table-based format that the various solvers (GENIOS, lpSolve, custom algorithms) can use. All solvers rely on a common data model. This is the data model underlying GENIOS [6], which deemed appropriate for the other solvers as well. In this model, the data of the optimization problem are distributed into five input tables, as shown in Figure 1 ("T" indicates a table).

Input data are imported at the moment when the solver procedure is invoked. Next the optimization task is carried out. Afterwards, the final result (objective function value) and the performance parameters are sent to the output table.

3.2 Performance measuring concept

In order to measure the solvers' performance and the accuracy of the test results, the following indicators were applied:

- *Function runtime* – time needed for calculation of the solution (i.e. the time the data spends in the optimization algorithm)

- *Execution time* – time from procedure invocation to the end of the procedure, resulting in output of the optimization result; includes time needed to access and transfer the problem data

- *Data load time* – difference between execution time and function runtime

- *Number of iterations* – number of iterations the optimization algorithm needs to compute the final solution

- *Solution value* – final value of the objective function for the particular problem being tested

Whereas the first indicator focuses on the time the optimization algorithm needs, the second one additionally considers the time needed for data transfer from and to the database tables. The third one – total number of iterations needed for calculation of the final solution – helps to understand the algorithm's performance and the problem complexity (i.e. numerical stability/instability). The fourth indicator, solution value (final value of the objective function) allows to compare the achieved solution with the optimal solution provided in the test case library.

4 Characteristics of problem set and test environment

In order to be able to validate the test results, collections of optimization problems suggested in the literature were used. For the linear programming tests, 80 real-life problems of different size and complexity were taken from the NetLib library [15]. NetLib is a set of LP problems available on the Internet, which is often used for benchmarking LP solvers.

Table 1 outlines the LP problems chosen for the tests. The columns show the name of the problem (in NetLib); the numbers of constraints, variables and non-zero elements; and the known optimum of the objective function. The number of non-zeros is important because it indicates the amount of input data to be parsed into the input tables and then read during execution of the solution procedure.

Similarly, a set of 40 reference problems from a well-known library of CWLP models [16], proposed by Beasley [17], were selected for the test runs using the MIP solvers (cf. table 2).

Model data of all of the problems displayed in the tables 1 and 2 were retrieved from the libraries and saved locally as text files.

Name	Cons	Var	Nonzeros	Optimum
AFIRO	27	32	88	-464.75
SC50B	50	48	119	-70.00
SC50A	50	48	131	-64.58
SC105	105	103	281	-52.20
KB2	43	41	291	-1,749.90
ADLITTLE	56	97	465	225,494.96
BLEND	74	83	521	-30.81
SC205	205	203	552	-52.20
SCAGR7	129	140	553	-2,331,389.82
SHARE2B	96	79	730	-415.73
RECIPELP	91	180	752	-266.62
VTP-BASE	198	203	914	129,831.46
LOTFI	153	308	1,086	-25.26
SHARE1B	117	225	1,182	-76,589.32
BOEING2	166	143	1,339	-315.02
BORE3D	233	315	1,525	1,373.08
SCORPION	388	358	1,708	1,878.12
CAPRI	271	353	1,786	2,690.01
SCAGR25	471	500	2,029	-14,753,433.06
SCTAP1	300	480	2,052	1,412.25
BRANDY	220	249	2,150	1,518.51
ISRAEL	174	142	2,358	-896,644.82
SCFXM1	330	457	2,612	18,416.76
BANDM	305	472	2,659	-158.63
FINNIS	497	614	2,714	172,791.07
STANDATA	359	1,075	3,038	1,257.70
STANDGUB	361	1,184	3,147	1,257.70
SCSD1	77	760	3,148	8.67
GFRD-PNC	616	1,092	3,467	6,902,236.00
BEACONFD	173	262	3,476	33,592.49
STANDMPS	467	1,075	3,686	1,406.02
BOEING1	351	384	3,865	-335.21
SCRS8	490	1,169	4,029	904.30
DEGEN2	444	534	4,449	-1,435.18
AGG2	516	302	4,515	-20,239,252.36
TUFF	333	587	4,523	0.31
AGG3	516	302	4,531	10,312,115.94
SEBA	515	1,028	4,874	16,189.30
SHELL	536	1,775	4,900	1,208,825,346.00
PILOT4	410	1,000	5,145	-2,581.14
SCFXM2	660	914	5,229	36,660.26
SCSD6	147	1,350	5,666	50.50
SHIP04S	402	1,458	5,810	1,798,714.70
PEROLD	625	1,376	6,026	-9,380.76
BNL1	643	1,175	6,129	1,977.63
FFFFF800	524	854	6,235	555,679.56
GANGES	1,309	1,681	7,021	-109,585.74
SCFXM3	990	1,371	7,846	54,901.25
SCTAP2	1,090	1,880	8,124	1,724.81
SHIP04L	402	2,118	8,450	1,793,324.54
PILOT-WE	722	2,789	9,218	-2,720,107.53
SIERRA	1,227	2,036	9,252	15,394,362.18
SHIP08S	778	2,387	9,501	1,920,098.21
MAROS	846	1,443	10,006	-58,063.74
SCTAP3	1,480	2,480	10,734	1,424.00
FIT1P	627	1,677	10,894	9,146.38
SHIP12S	1,151	2,763	10,941	1,489,236.13
25FV47	822	1,571	11,127	5,501.85
SCSD8	397	2,750	11,334	905.00
PILOTNOV	975	2,172	13,129	-4,497.28
CZPROB	929	3,523	14,173	2,185,196.70
FIT1D	24	1,026	14,430	-9,146.38
PILOT-JA	940	1,988	14,706	-6,113.14
BNL2	2,324	3,489	16,124	1,811.24
SHIP08L	778	4,283	17,085	1,909,055.21
CYCLE	1,903	2,857	21,322	-5.23
SHIP12L	1,151	5,427	21,597	1,470,187.92
DEGEN3	1,503	1,818	26,230	-987.29
80BAU3B	2,263	9,799	29,063	987,224.19
GREENBEB	2,392	5,405	31,499	-4,302,260.26
GREENBEA	2,392	5,405	31,499	-72,555,248.13
D2Q06C	2,171	5,167	35,674	122,784.21
WOODW	1,098	8,405	37,478	1.30
DFL001	6,071	12,230	41,873	11,266,396.05
PILOT	1,441	3,652	43,220	-557.49
D6CUBE	415	6,184	43,888	314.92
WOOD1P	244	2,594	70,216	1.44
PILOT87	2,030	4,883	73,804	301.71
FIT2D	25	10,500	138,018	-68,464.29
MAROS-R7	3,136	9,408	151,120	1,497,185.17

Table 1. LP test cases used for evaluation

Name	Cons	Var	Tableau	Optimum
CAP41	16	50	800	1,040,444.38
CAP42	16	50	800	1,098,000.45
CAP43	16	50	800	1,153,000.45
CAP44	16	50	800	1,235,500.45
CAP51	16	50	800	1,025,208.23
CAP61	16	50	800	932,615.75
CAP62	16	50	800	977,799.40
CAP63	16	50	800	1,014,062.05
CAP64	16	50	800	1,045,650.25
CAP71	16	50	800	932,615.75
CAP72	16	50	800	977,799.40
CAP73	16	50	800	1,010,641.45
CAP74	16	50	800	1,034,976.98
CAP81	25	50	1,250	838,499.29
CAP82	25	50	1,250	910,889.56
CAP83	25	50	1,250	975,889.56
CAP84	25	50	1,250	1,069,369.53
CAP91	25	50	1,250	796,648.44
CAP92	25	50	1,250	855,733.50
CAP93	25	50	1,250	896,617.54
CAP94	25	50	1,250	946,051.33
CAP101	25	50	1,250	796,648.44
CAP102	25	50	1,250	854,704.20
CAP103	25	50	1,250	893,782.11
CAP104	25	50	1,250	928,941.75
CAP111	50	50	2,500	826,124.71
CAP112	50	50	2,500	901,377.21
CAP113	50	50	2,500	970,567.75
CAP114	50	50	2,500	1,063,356.49
CAP121	50	50	2,500	793,439.56
CAP122	50	50	2,500	852,524.63
CAP123	50	50	2,500	895,302.33
CAP124	50	50	2,500	946,051.33
CAP131	50	50	2,500	793,439.56
CAP132	50	50	2,500	851,495.33
CAP133	50	50	2,500	893,076.71
CAP134	50	50	2,500	928,941.75
CAPA (8000)	100	1,000	100,000	19,240,822.45
CAPA (10000)	100	1,000	100,000	18,438,046.54
CAPA (12000)	100	1,000	100,000	17,765,201.95

Table 2. CWL test cases used for evaluation

The hardware configurations used for the test runs have the following characteristics:

- HANA server: 64 virtual cores (2 GHz), 1 TB RAM
- R server: 4 virtual cores (2 GHz), 16 GB RAM.

This infrastructure was shared between all users of the HANA instance pool.

5 Results

The test problems outlined in section 4 were uploaded, parsed, and tested using the solvers implemented in HANA. The results were then exported to Excel in order to perform a more detailed analysis.

Since the infrastructure was shared between many users assigned to the same HANA instance pool, it cannot be assumed that the hardware resources were exclusively allocated to a particular test run. In order to obtain more reliable results, variances in resource allocation must be leveled. Therefore, each test case was repeated several times (at different times and days) and then the mean value was taken.

5.1 LP results

For the linear programming problems, test results are shown in table 3. In particular, the execution time, the

function runtime (i.e. time the algorithm needs), and the data-load time of each problem are listed.

Three test cases could not be solved on the R server, but were returned with execution time overrun. In all other cases, both solvers returned the same optimal solution as documented in the NetLib library.

Execution times for GENIOS are much shorter than execution times for *lpSolve* (on average three times shorter). The most likely explanation is that GENIOS is tightly integrated with the database [8, p. 14] and therefore can access problem data very quickly. In contrast to this, *lpSolve* requires problem data to be sent to the R server via a network first, before the problem can be solved.

This explanation is supported by an analysis of the data-load times. Comparing these times, on average 290 milliseconds more per test run were needed for data transfer between the R solver and the database than between GENIOS and the database.

Regarding the function runtime, the dominance of GENIOS is not as clear. GENIOS still outperformed R, yet only in 61 test cases. In 19 test cases, the function runtime of the R solver was smaller than the function runtime of the GENIOS solver. However, these shorter runtimes can only compensate the difference in execution times by approximately 15%.

An important observation is that most NetLib problems are numerically instable. This requires additional operations and checking, resulting in a significant amount of runtime, regardless of the problem size.

5.2 CWL results

The two CWLP heuristics used, *Add* and *Drop*, do not necessarily find an optimal solution. Therefore, we compared not only their performance measures with GENIOS but also their solution quality.

The test results are shown in table 4. The column "Abs. deviation" shows how far away from the known optimum (as documented by academia [16]) the solution found by the particular solver is.

Regarding *solution quality*, GENIOS remains unrivaled, being able to find the optimum in every test run. Looking at the other two solvers, more accurate results are returned by *Add*, with a mean deviation from the optimum of 0.19%. Results provided by *Drop* are mostly inaccurate – with the mean deviation from the optimum of 31%.

Considering *execution times*, GENIOS outperforms the other solvers. It is 16 times faster than *Drop* and more than 164 times faster than *Add*. This huge difference is mostly due to the fact that our implementations are not very sophisticated. Therefore, a large number of iterations are needed to approximate the optimum. The *Add* procedure is slower than *Drop*, but its solution quality is better.

Name	GENIOS			R (lpSolve)		
	Execution time [ms]	Function runtime [ms]	Data-load time [ms]	Execution time [ms]	Function runtime [ms]	Data-load time [ms]
AFIRO	40	9	31	387	27	360
SC50B	60	54	6	352	31	321
SC50A	42	20	22	355	31	324
SC105	41	10	31	324	32	292
KB2	90	20	70	531	17	514
ADLITTLE	43	10	33	364	32	332
BLEND	70	53	17	367	39	328
SC205	47	20	27	358	42	316
SCAGR7	60	40	20	297	36	261
SHARE2B	54	30	24	343	37	306
RECIPELP	104	40	64	504	39	465
VTP-BASE	102	50	52	540	55	485
LOTFI	56	30	26	381	54	327
SHARE1B	63	40	23	393	50	343
BOEING2	232	120	112	623	87	536
BORE3D	209	100	109	692	76	616
SCORPION	64	40	24	355	60	295
CAPRI	196	110	86	668	79	589
SCAGR25	89	70	19	478	148	330
SCTAP1	74	50	24	361	53	308
BRANDY	87	80	7	378	54	324
ISRAEL	78	60	18	377	53	324
SCFXM1	103	90	13	442	82	360
BANDM	101	80	21	424	114	310
FINNIS	330	270	60	695	137	558
STANDATA	122	60	62	585	109	476
STANDGUB	127	60	67	615	121	494
SCSD1	72	50	22	363	42	321
GFRD-PNC	142	80	62	623	178	445
BEACONFD	71	40	31	387	44	343
STANDMPS	135	70	65	664	181	483
BOEING1	249	140	109	641	99	542
SCRS8	111	100	11	473	114	359
DEGEN2	154	140	14	588	258	330
AGG2	71	50	21	387	61	326
TUFF	153	80	73	621	134	487
AGG3	85	60	25	393	66	327
SEBA	130	70	60	657	192	465
SHELL	163	110	53	684	230	454
PILOT4	422	370	52	803	319	484
SCFXM2	220	210	10	671	264	407
SCSD6	115	90	25	510	102	408
SHIP04S	103	80	23	494	108	386
PEROLD	609	560	49	1,140	652	488
BNL1	241	230	11	621	239	382
FFFFF800	123	100	23	613	207	406
GANGES	166	160	6	796	323	473
SCFXM3	272	250	22	1,116	607	509
SCTAP2	171	150	21	651	220	431
SHIP04L	133	110	23	553	137	416
PILOT-WE	1,617	1,610	7	2,241	1,485	756
SIERRA	206	140	66	808	343	465
SHIP08S	140	120	20	677	208	469
MAROS	277	230	47	1,326	848	478
SCTAP3	225	210	15	1,084	326	758
FIT1P	531	490	41	1,201	622	579
SHIP12S	145	130	15	1,204	425	779
25FV47	1,320	1,283	37	1,720	1,259	461
SCSD8	328	310	18	893	357	536
PILOTNOV	1,290	1,034	256	1,524	753	771
CZPROB	353	300	53	2,019	1,362	657
FIT1D	405	330	75	722	226	496
PILOT-JA	1,095	1,070	25	2,161	1,408	753
BNL2	527	520	7	1,849	1,205	644
SHIP08L	254	230	24	964	358	606
CYCLE	890	813	77	2,193	1,574	619
SHIP12L	308	280	28	1,816	749	1,067
DEGEN3	1,880	1,821	59	5,042	4,405	637
80BAU3B	1,632	1,610	22	-	-	-
GREENBEB	5,220	5,061	159	11,994	11,405	589
GREENBEA	7,810	7,502	308	16,715	16,178	537
D2Q06C	9,710	9,366	344	11,314	10,438	876
WOODW	515	500	15	4,037	2,836	1,201
DFL001	87,970	74,040	13,930	76,747	76,073	674
PILOT	7,590	7,330	260	-	-	-
D6CUBE	1,730	1,711	19	2,455	1,955	500
WOOD1P	415	380	35	1,951	1,334	617
PILOT87	16,090	15,413	677	-	-	-
FIT2D	1,910	1,893	17	5,267	4,511	756
MAROS-R7	3,320	3,245	75	17,958	15,454	2,504

Table 3. LP test results

Name	GENIOS		ADD			DROP		
	Execution time [ms]	Function runtime [ms]	Abs. deviation [%]	Execution time [ms]	Function runtime [ms]	Abs. deviation [%]	Execution time [ms]	Function runtime [ms]
CAP41	139	88	0.0	16,912	14,616	0.2	1,961	1,804
CAP42	300	294	0.0	16,536	14,312	0.0	1,797	1,688
CAP43	235	229	0.0	16,082	13,867	0.0	1,949	1,833
CAP44	189	168	0.0	17,071	14,900	0.0	2,045	1,898
CAP51	326	323	0.2	12,919	10,688	2.4	1,727	1,607
CAP61	124	107	0.0	15,986	13,785	1.5	1,784	1,657
CAP62	112	107	0.4	15,454	13,248	1.1	1,846	1,744
CAP63	313	303	0.2	11,425	9,195	21.4	1,719	1,599
CAP64	183	161	0.0	11,492	9,295	19.9	1,977	1,865
CAP71	82	61	0.0	15,704	13,488	1.5	1,865	1,757
CAP72	86	66	0.4	15,448	13,169	1.1	1,871	1,745
CAP73	93	72	0.2	9,847	7,643	1.3	1,797	1,683
CAP74	111	71	0.0	9,912	7,705	0.4	1,944	1,824
CAP81	224	204	0.3	42,757	37,881	40.4	3,123	2,982
CAP82	378	371	0.7	39,176	34,266	35.3	3,304	3,162
CAP83	341	349	0.4	37,693	32,801	31.9	3,270	3,124
CAP84	499	497	0.0	37,927	33,001	28.1	3,056	2,930
CAP91	186	163	0.1	38,025	33,097	4.3	3,025	2,890
CAP92	245	226	0.0	33,726	28,804	31.4	3,223	3,078
CAP93	456	454	0.2	28,114	23,240	148.2	3,261	3,055
CAP94	487	477	0.2	26,356	21,382	289.7	3,120	2,980
CAP101	187	174	0.1	37,383	32,304	4.3	3,074	2,943
CAP102	124	113	0.1	33,425	28,258	3.2	3,062	2,923
CAP103	117	91	0.1	23,176	18,190	19.2	3,136	3,029
CAP104	215	199	0.0	18,788	13,791	18.5	2,756	2,648
CAP111	422	425	0.3	131,544	113,290	15.2	8,376	8,180
CAP112	836	833	0.4	120,788	103,380	11.6	7,974	7,768
CAP113	1,134	1,166	0.2	116,278	98,854	9.4	8,066	7,859
CAP114	1,126	1,164	0.2	110,821	93,287	7.6	8,008	7,822
CAP121	346	325	0.1	114,204	96,326	12.3	8,038	7,840
CAP122	562	557	0.0	96,700	79,097	34.4	8,155	7,960
CAP123	774	789	0.2	85,089	67,478	155.7	8,226	8,044
CAP124	1,151	1,178	0.2	73,085	55,519	144.3	8,134	7,948
CAP131	163	148	0.1	111,460	93,824	9.9	7,235	7,035
CAP132	183	164	0.1	93,764	76,211	5.7	7,000	6,821
CAP133	215	203	0.1	70,982	53,512	20.7	7,072	6,907
CAP134	266	253	0.0	51,725	34,245	18.5	7,278	7,111
CAPA (8000)	-	-	0.6	2,566,325	1,873,101	38.5	612,746	605,786
CAPA (10000)	-	-	0.7	1,960,734	1,355,944	33.3	452,523	445,896
CAPA (12000)	-	-	0.6	2,903,135	2,118,931	37.4	687,438	679,630

Table 4. CWL test results

An analysis of the *function runtime* yields similar results. GENIOS is by far the fastest. Its mean calculation time is 340 milliseconds per test run, whereas *Drop* and *Add* need 4,047 milliseconds and 39,404 milliseconds, respectively.

However, it needs to be mentioned that GENIOS was not able to solve larger CWL problems (100,000 non-zeros) – not even within the same time that the heuristics took.

Apart from this limitation, GENIOS delivered more accurate results within significantly shorter times than any of the two heuristics.

6 Conclusions and outlook

Based on our comprehensive test runs, it can be concluded that GENIOS offers significantly better performance than the other optimization approaches. This is true for both the LP and the CWL problems.

The performance gap between GENIOS and *lpSolve* on the R server can be partly explained by the differences in data transfer between the database and the solver. GENIOS is tightly integrated with HANA's index server, resulting in very fast data access. For *lpSolve*, on the other hand, data must be transferred over a network to the R server.

However, looking solely at the function runtime, the difference between GENIOS and *lpSolve* is not so big. For the test problem set, GENIOS still outperformed R, but only in three quarters of the problems. This confirms again that the most crucial factor is the relatively slow data transfer between the HANA database and the R server. Shorter function runtimes in R could not compensate this slowing down.

For the CWL problems, huge differences between GENIOS and the *Add* and *Drop* heuristics were observed, regarding both execution time and solution quality. Whereas the heuristics miss the optimum in two out of three cases, GENIOS computes the exact optimal solution in all cases. *Add* returns better solutions than *Drop*. While *Add* is slower than *Drop*, both heuristics are very slow compared to GENIOS.

The results show that both heuristic approaches were not the best choices. Other techniques most likely would provide better results. However, it is doubtful whether SQLScript based implementations could ever compete with GENIOS.

The main limitation of our results is that all calculations were performed using a shared infrastructure. Hence, the performance results may be partially biased by varying amounts of resources allocated to the test runs. In further research, performance tests should be conducted using a dedicated infrastructure. This would enhance the credibility of the performance measurements and conclusions found in this study.

In addition, more external solvers from well-known optimization packages (such as Cplex, Gurobi, or Lingo) should be tested. Since GENIOS is tightly integrated with HANA, it would be interesting to see how those really "external" servers compare with GENIOS regarding the runtimes for optimization.

References

[1] K. Kurbel: *Enterprise Resource Planning and Supply Chain Management. Functions, Business Processes and Software for Manufacturing Companies*. Springer, Heidelberg, New York, 2013.

[2] M. Hoppe: *Sales and Inventory Planning with SAP APO*. Galileo Press, Boston, 2007.

[3] H. Plattner and A. Zeier: *In-Memory Data Management - Technology and Applications*. Springer, Berlin, Heidelberg, 2012.

[4] H. Plattner: A common database approach for OLTP and OLAP using an in-memory column database. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data (pp. 1-2). ACM.*, Providence, Rhode Island, USA, 2009.

[5] J. Word: *SAP HANA Essentials.* 2nd ed., Epistemy Press, 2012.

[6] SAP AG: *OFL AFL (internal documentation)*. 2014.

[7] SAP AG: *SAP HANA Predictive Analysis Library (PAL)*. 16 February 2015. [Online]. Available: http://help.sap.com/hana/SAP_HANA_Predictive_An

alysis_Library_PAL_en.pdf. [Accessed 20 February 2015].

[8] SAP AG: *SAP HANA Developer Guide - SAP HANA Platform SPS 08*. 21 August 2014. [Online]. Available: http://help.sap.com/hana/SAP_HANA_Developer_Guide_en.pdf. [Accessed 20 January 2015].

[9] K. Kurbel and D. Nowak: Is SAP HANA Useful for Optimization? - An Exploration on LP Implementation Alternatives. In: *Proceedings of the ERP Future 2014 Conference, 17.-18. November 2014*, Dornbirn, Austria, 2014.

[10] SAP AG: *SAP HANA SQLScript Reference*. 2014. [Online]. Available: http://help.sap.com/hana/SAP_HANA_SQL_Script_Reference_en.pdf. [Accessed 20 January 2015].

[11] SAP AG: *SAP HANA R Integration Guide*. 2014. [Online]. Available: http://help.sap.com/hana/SAP_HANA_R_Integration_Guide_en.pdf. [Accessed 20 January 2015].

[12] K. Konis: *R Interface for lp_solve version 5.5.2.0*. 12 November 2014. [Online]. Available: http://cran.r-project.org/web/packages/lpSolveAPI/lpSolveAPI.pdf. [Accessed 20 February 2015].

[13] S.K. Jacobsen: Heuristics for the capacitated plant location model. *European Journal of Operational Research*, 12(3):253-261, 1983.

[14] P. Notebaert and K. Eikland: *MPS file format*. [Online]. Available: http://lpsolve.sourceforge.net/5.5/mps-format.htm. [Accessed 31 March 2015].

[15] Netlib.org: *The NETLIB LP Test Problem Set*. [Online]. Available: http://www.netlib.org/lp/data/. [Accessed 20 January 2015].

[16] J.E. Beasley: *OR-Library - Capacitated warehouse location*. June 2012. [Online]. Available: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/capinfo.html. [Accessed 20 January 2015].

[17] J.E. Beasley: OR-Library: distributing test problems by electronic mail. *European Journal of Operational Research*, 41(11):1069-1072, 1990.

Security Monitoring and Analytics of HPI FutureSoC Lab (Phase I)

Amir Azodi, David Jaeger, Feng Cheng, Christoph Meinel

Hasso Plattner Institute (HPI)
University of Potsdam
14482 Potsdam, Germany
amir.azodi@hpi.de, david.jaeger@hpi.de,
feng.cheng@hpi.de, christoph.meinel@hpi.de

Abstract. Utilizing the resources of the HPI FutureSoC Lab we have worked on resolving a number of challenges in the field of Network Monitoring. As computer networks grow in size and complexity, monitoring them becomes more challenging. In order to meet the needs of IT administrators maintaining such networks, various *Network Monitoring Systems (NMS)* have been developed. Most NMSs rely solely on active scanning techniques in order to detect the topology of the networks they monitor. We propose a passive scanning solution using the logs produced by the systems within the networks. Additionally, we demonstrate how passive monitoring can be used to develop a holistic knowledge graph of the network landscape.

Index terms— Network Monitoring, Network Graph, Event Processing, Event Normalization

1 Introduction

A primary benefit of NMSs is that they allow the administrators to monitor the systems across their network more effectively by being able to observe events of interest, whether they are related to the security, availability, quality of service, or other aspects of keeping a network and its available services operating in an optimal manner. Often administrators need to target systems matching a specific criteria within their network's landscape, for instance when trying to patch newly discovered security vulnerabilities, issuing software updates, or detecting hardware anomalies. License management and regulatory compliance is another area where operating a NMS can be beneficial. Interestingly, although the concept of a generic and automated NMS — using logical network topology graphs — has been around for quite some time[1], software vendors have largely opted for providing vendor specific NMSs. As a result, even for simple tasks such as getting the system uptime information of a particular host, network administrators have had to deal with different NMSs or even rely on simpler methods (i.e. locally/remotely reading a host's system logs). Many software vendors provide their own means of monitoring their systems within a network. Therefore these

NMSs are often not self-contained and can only monitor systems specific to a subset of the network.

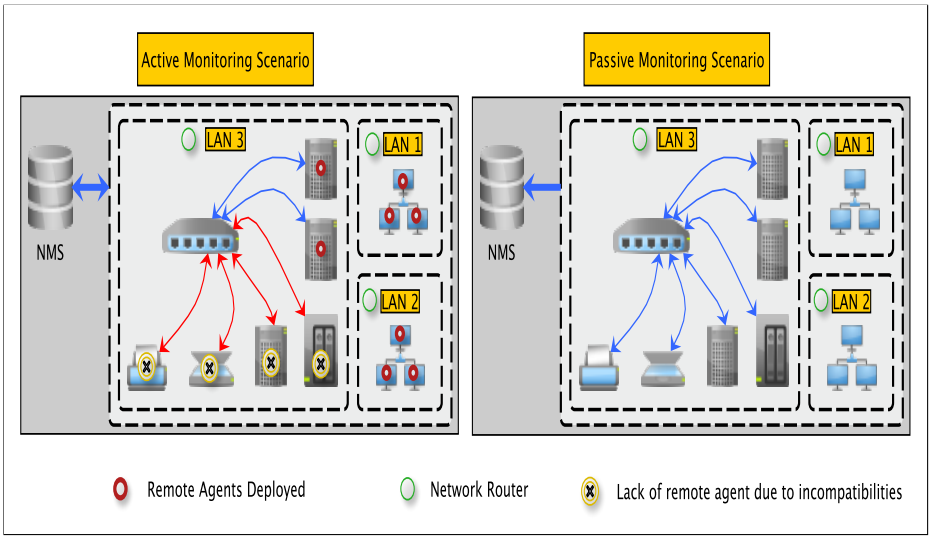


Fig. 1. This figure demonstrates the logical difference between Active and Passive network monitoring mechanisms.

2 Using Processed Events to Generate a Network Graph

Mapping out a network requires an understanding of the underlying *logical* topology of the network. By focusing on passive monitoring we inherently avoid performance and compatibility issues marring active monitoring systems. From the events which are useful in building a network graph (events that include information from the higher layers of the OSI Model), a subset is used for mapping the lower levels of the networks logical structure. We refer to this subset as Network Layer Events (NLE). NLEs are often produced by networking systems such as Routers, Switches, Firewalls, as well as basic network services such as DHCP, DNS, IDS Sensors, Wireless Access points, etc. Events used to populate information about an individual object's applications are referred to as Application Level Events (ALE). ALEs are produced by Operating Systems, Web Servers, FTP Servers, Mail Servers, Authentications Services, etc. By processing a set of events produced by a network firewall, our passive monitoring system can detect an internal network and the presence of hosts within that network. The sample event set used for the purposes of this paper is compromised of 1586 events produced over a time span of 4 hours and 12 minutes by a Cisco ASA firewall system. We used a relatively small event set in order to be able to produce reasonably sized figures due to space constraints.

Using the information extracted from the processed events we attempt to construct a map of the network topology by observing the connections between the communicating hosts. Figure 2 demonstrates a simple example mapping communications between one subnet of the internal network and a subset of the external networks and hosts. Subsequently, by monitoring the application level events an overall specification list for the individual systems is formed. Host specific information is then overlaid on to the network topology map to construct a comprehensive view of the network. In addition to the ALEs, the context information added to the ALEs during their processing by REAMS can also be used to create a more complete image of the system. By analyzing NLEs in our sample log file, we can detect the presence of multiple external networks. As a security feature, we resolved the external IP addresses discovered in the events and match them against a list of known malicious domains found in [2]. We discovered no correlation between the external hosts and the list of known malicious domains. However such tests could be instrumental in discovering network security breaches by network monitoring systems. Table **??** represents the 10.10.10.0/24 subnet discovered (one of the internal subnets) and displays the hosts within the subnet as well as the ports used for communication by those hosts.[1]

We resolve the port numbers against a knowledge base of port number information. Table 1 presents a mapping between the ports and the relevant information. At this point we can observe that there is no service *registered* on port 1028, although it is being used. Additionally it can be seen that in the past certain malware have used this port for communication. In a commercial setting, the NMS could raise an alert to inform the network administrators of the event's occurrence.

Port #	TCP	UDP	Known Service(s)	Status
17	✗	✓	Quote of the Day	Official
22	✓	✗	Secure Shell (SSH)	Official
53	✗	✓	Domain Name System (DNS)	Official
80	✓	✗	Hypertext Transfer Protocol (HTTP)	Official
123	✗	✓	Network Time Protocol (NTP)	Official
161	✗	✓	Simple Network Management Protocol (SNMP)	Official
427	✗	✓	Service Location Protocol (SLP)	Official
443	✓	✗	Hypertext Transfer Protocol over TLS/SSL (HTTPS)	Official
500	✗	✓	Internet Security Association...(ISAKMP)	Official
993	✓	✗	IMAP over TLS/SSL (IMAPS)	Official
1027	✗	✓	Native IPv6 behind IPv4-to-IPv4 NAT	Official
1028	✗	✓	**KiLo, SubSARI**	**Unknown**
1702	✗	✓	Deskshare	Unofficial
1906	✗	✓	Unknown	Unknown

Table 1. Correlating a subset of the communication open ports with likely services.[2]

[1] Some ports and hosts were omitted due to size constraints

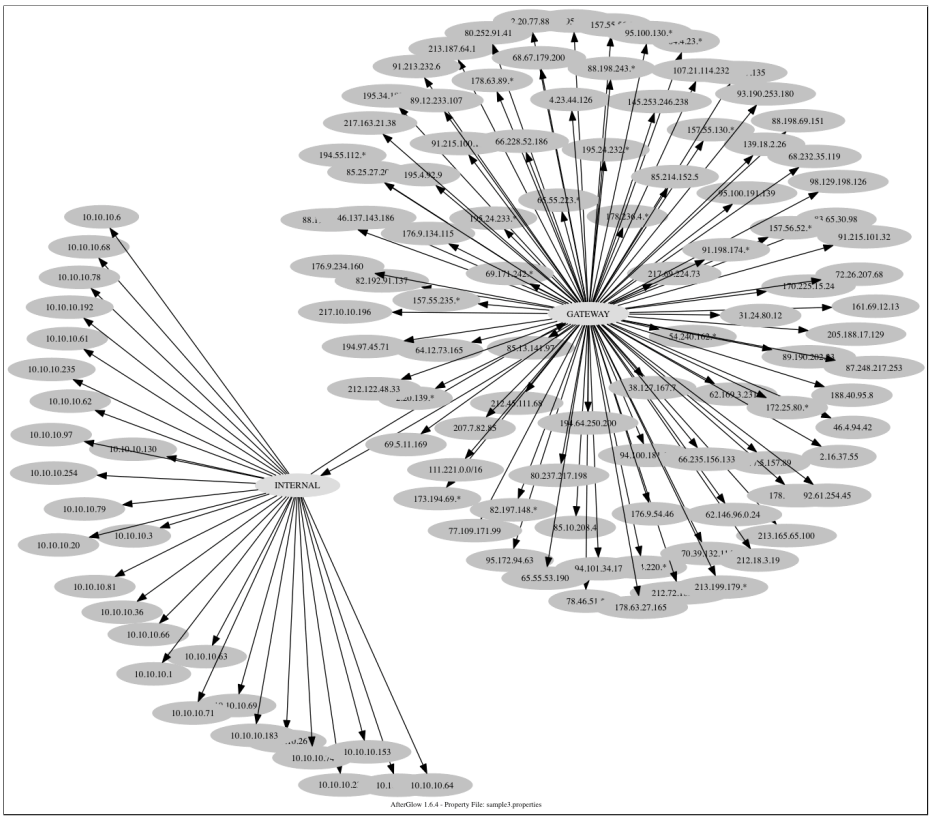


Fig. 2. A simplified view of the network topology using event driven passive network discovery.

References

1. A.B. Bondi. Network management system with improved node discovery and monitoring, January 20 1998. US Patent 5,710,885.
2. The DNS-BH project. Malware prevention through domain blocking (black hole dns sinkhole), 2014. [Online; accessed 11-August-2014].

[2] Well Known Ports: 0 through 1023.
Registered Ports: 1024 through 49151.
Dynamic/Private : 49152 through 65535.

Technical Report

Machine Learning for Security Analytics powered by SAP HANA (Phase II)

Future SOC Lab Project

by

Andrey Sapegin, Marian Gawron, Feng Cheng, Christoph Meinel

Potsdam, Version: March 23, 2015

1 Project Concepts

Nowadays, malicious user behaviour that does not trigger access violation or alert of data leak is difficult to be detected. Using the stolen login credentials the intruder doing espionage will first try to stay undetected: silently collect data from the company network and use only resources he is authorised to access. To deal with such cases, a Poisson-based anomaly detection algorithm is proposed in this report. To prove the proposed approach, we developed a special simulation testbed that emulates user behaviour in the virtual network environment deployed in the Future SOC Lab. The proof-of-concept implementation has been integrated into our prototype of a SIEM system — Real-time Event Analysis and Monitoring System, where the emulated Active Directory logs from Microsoft Windows domain are extracted and normalised into Object Log Format for further processing and anomaly detection. The experimental results show that our algorithm was able to detect all events related to malicious activity and produced zero false positive results. Forethought as the module for our self-developed SIEM system based on the SAP HANA in-memory database, our solution is capable of processing high volumes of data and shows high efficiency on experimental dataset. The results of this project phase will be utilised to analyse real data and confirm the efficiency of this approach on the non-simulated dataset.

1.1 Target scenarios

We decided to analyse the case when a user is authorised to access a resource, but the resource is however never used neither by this user nor by any other users (from the same group) under normal working conditions. This case emulates an example of malicious user behaviour, when the user credentials were stolen and are now used to collect information from the enterprise network avoiding any access violations to stay undetected by an intrusion detection system. Under this case we describe 2 scenarios: (1) normal, which will be used to generate a training dataset, and (2) abnormal, used to generate a testing dataset, where all malicious events should be captured automatically. The details are illustrated in Figures 1.1 and 1.2.

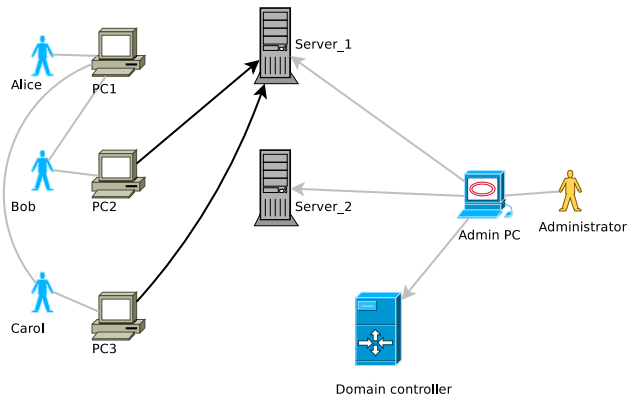


Figure 1.1: Normal behaviour of users in the network

Figure 1.1 describes normal behaviour of users in a small network. Alice, Bob and Carol are usual network users. All users have access to both Server_1 and Server_2,

however, only Bob and Carol regularly log on to the Server_1, and nobody accesses Server_2. Like in the real network, a user first needs to login to their PCs, and only then he could access a server remotely. The users could also login on each other's PCs, e.g. Carol logs on to the PC1 of Alice. Finally, the network administrator has an access to all virtual machines in the network and regularly use all Servers (including Domain Controller) as well as his own PC.

Different to the normal scenario, Figure 1.2 shows an example of malicious user behaviour.

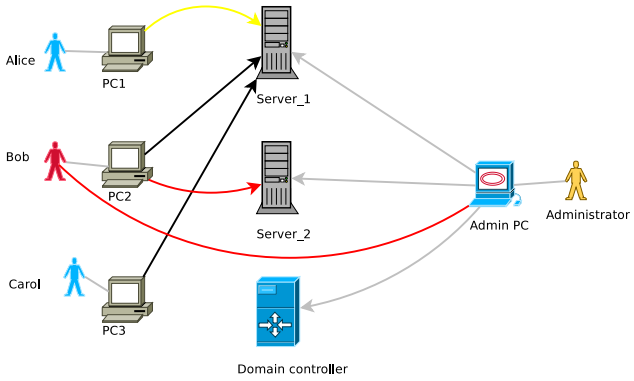


Figure 1.2: Malicious user behaviour of user Bob

In Figure 1.2, Bob connects to the Server_2, which was never used before neither by him, nor by other users, despite the fact, that all users are authorised to access this server. We classify such cases as malicious user behaviour. Bob's local connection to Admin PC should be then classified malicious as well. However, the fact, that Alice connects to the Server_1 should be classified as benign, even though it was also not expected for her. We do not mark such cases as suspicious, since the Server_1 has been regularly accessed by other users from the same group (Bob and Carol). We therefore classify only Bob's behaviour as malicious and will try to capture it during the analysis phase.

Our simulation scenario bases on the testbed with virtual machines running on the VMware ESXi virtualisation server inside the Future SOC Lab network. For all PCs — PC1-PC3, as well as Admin PC — we use virtual machines with Microsoft Windows 7. Server_1 and Server_2 have Microsoft Windows Server 2003 installed, while for the Domain controller we setup Microsoft Windows Server 2012. The Domain controller just provides Active Directory services, DHCP and DNS, meanwhile all virtual machines have Remote Desktop enabled. Using this testbed, we simulate logins on the user PCs as well as connections to the servers and domain controller.

To simulate user logon and logoff events, we decided to utilise the VNC access functionality of the VMware ESXi hypervisor. We have created a program in Python, that is able to simultaneously connect to different virtual machine consoles and execute user actions, such as logon and logoff, based on the screen capture and recognition functions provided by Python Imaging Library. To simulate connections from user PCs to servers, we use Remote Desktop Protocol. The simulation software is able to unlock the computer (by sending the "Ctrl-Alt-Del" combination and typing in a password) and then start the PowerShell script located on the Desktop of each virtual PC. This script opens the connection via Remote Desktop to the predefined server.

Since every logon event (both local and via RDP) is saved by as the Windows Event and reported to the domain controller, we are able to realistically reproduce Active Directory logs for our scenario of user behaviour, including both local logon/logoff events and connections to server.

2 Poisson-based anomaly detection

To identify anomalies in the user behaviour, we first divide the data into time intervals. We choose the appropriate time interval heuristically, since it depends on the time range of a dataset, the number of users and on how often they perform logon events. For our simulated data we consider 15 minutes as optimal time interval.

After selection of the time interval, we calculate number of logon events per time interval for each user group on each workstation to check the probability of this number of logon events according to the Poisson's formula:

$$P(x) = \frac{e^{-\lambda}\lambda^{x}}{x!} \tag{2.1}$$

where x is number of logon events per time interval. Please see Algorithm 1 for details.

Algorithm 1 Anomaly_detection_groups($threshold_{group}$)

```
1: for all {user_group,workstation} from training_data do
2:    λ_group = average number of logon events per time_interval
3: end for
4: for all time_interval in testing_data do
5:    for all {user_group,workstation} from testing_data do
6:       logons = number of logon events
7:       if PoissonsProbability({user_group,workstation}, logons, λ_group) <
         threshold_group then
8:          mark {user_group,workstation,time_interval} as suspicious
9:       end if
10:   end for
11: end for
```

The Algorithm 1 first calculates the value of λ_{group} for each {user_group,workstation} pair. For a Poisson's distribution, this value could be calculated as average number of logon events per time interval (line 2). The λ_{group} value represents a model of normal user group behaviour, based on the training data. On the lines 4-11 we apply our model on testing data to identify events, that differ from modelled normal behaviour of user groups. Using the same time interval as for training data, we count number of logon events for each {user_group,workstation} pair in the testing data (line 6) and calculate its probability (according to the Formula 2.1). On the line 7 we compare the probability of particular number of logon events for a user group on some workstation within a particular time interval with the threshold value. If the probability is less than threshold, we mark a triplet user_group,workstation,time_interval as *suspicious* on the line 8.

For all revealed *suspicious* events we utilise Algorithm 2 to identify which users from the highlighted groups caused it.

Algorithm 2 Anomaly_detection_users($threshold_{user}$)

1: **for all** {user,workstation} from training_data **do**
2: λ_{user} = average number of logon events per time_interval
3: **end for**
4: **for all** time_interval in testing_data **do**
5: **for all** {user,workstation} from testing_data where {user_group,workstation,time_interval} is *suspicious* **do**
6: **if** PoissonsProbability({user,workstation},λ_{user}) < $threshold_{user}$ **then**
7: **mark** {user,workstation,time_interval} as *anomaly*
8: **end if**
9: **end for**
10: **end for**

Similar to Algorithm 1, the Algorithm 2 calculates λ_{user} based on training data. First, we find an average number of logon events per time interval for each {user,workstation} pair (line 2). Next, from testing data, we select all users related to *suspicious* events identified by Algorithm 1 and check the probability of number of logon events per time interval for each user and each workstation (lines 4-10) using Formula 2.1. If the probability is less than threshold, we mark all events related to the triplet {user,workstation,time_interval} as **anomalies**.

Using a two-step probability check – first for user groups, and then for particular users from suspicious groups – we avoid a false positive anomaly alerts for situations when the user performs some action (login on the workstation), which is not usual for him personally, but usual for his user group.

3 Anomaly detection results

Let's review the generated dataset and check if our algorithm is able to detect cases of simulated malicious user behaviour. We present distribution of logon-related events from the training dataset in Figure 3.1, while Figure 3.2 shows the same distribution for the testing dataset.

Figures 3.1 and 3.2 describe the distribution of simulated logon events in the testbed network. Each circle points to the logon of the user (on the x-axis) on the workstation or server (on the y-axis). The logarithmically scaled size of the circles reflects number of events for each particular user and workstation/server. Both Figures correspond to original simulation scenarios, which provides an extra proof, that no logon activities were lost during data filtering.

After our Poisson-based anomaly detection algorithm learns user behaviour model from training dataset, the model is applied on testing dataset to find anomalies, that do not fit into the learned model. We present the identified anomalies on Figure 3.3.

Figure 3.3 shows results of anomaly detection using the same representation as Figures 3.1 and 3.2. Compared to Figure 3.2, it presents the subset of events (including users and workstation), that were marked as anomalous by our algorithm. The results prove that our algorithm has successfully captured all logon-related events for user

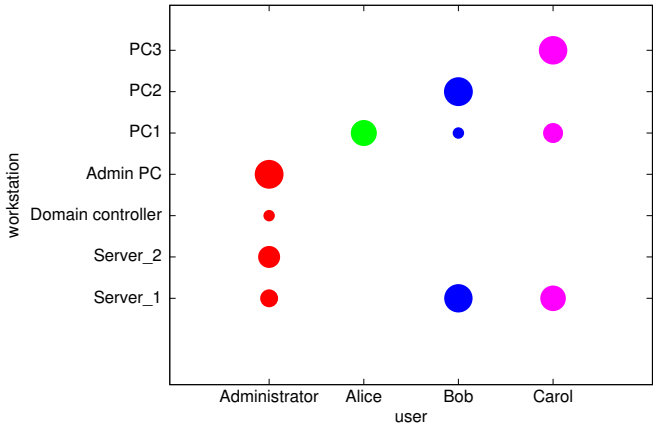


Figure 3.1: Distribution of user logon events in the training dataset

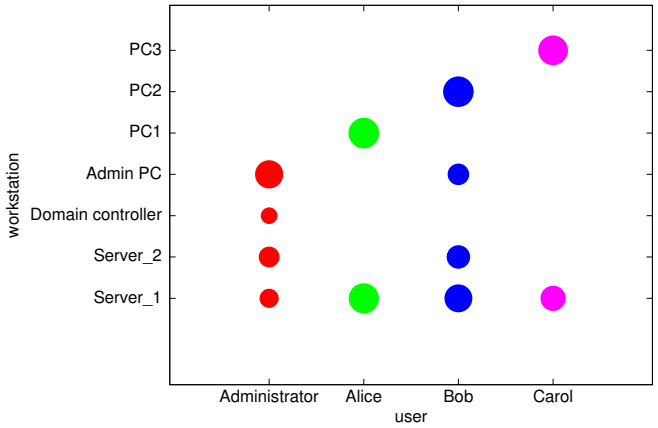


Figure 3.2: Distribution of user logon events in the testing dataset

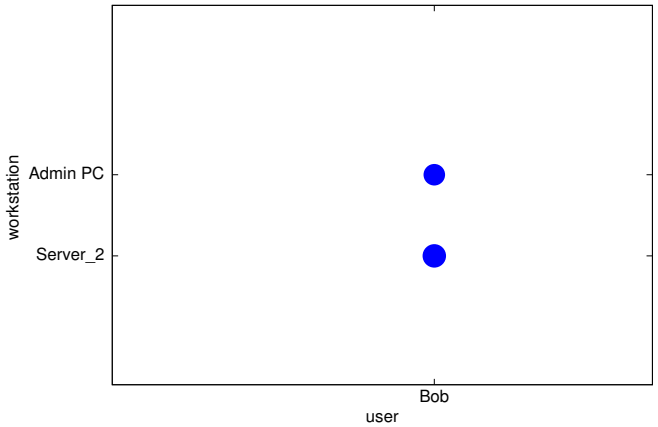


Figure 3.3: Detected user behaviour anomalies

Bob on both Server_2 and Admin PC. All these events are malicious according to our scenario. Moreover, no other event was classified as anomaly, which implies that our

approach to perform two-step probability check — first for user groups and then for each user individually — helped to avoid producing any false positive results.

4 Conclusion

We have developed an effective approach to detect malicious user activity even when such activity does not violate any access or data protection policies. Our results show that our algorithm is able to detect all existing malicious events without producing any false positive alarms. Such precision was possible due to the performed two-step probability check. By checking the Poisson probability first for each {user_group,workstation} pair, and later – for {user,workstation} pairs, we are able to correctly classify legitimate cases when the resource accessed by the group of users was not accessed by one of the users from that group for a long time period.

Our concepts were checked using the proof-of-concept implementation, that utilises simulated dataset. The simulation tool emulates user activity to produce real events on the domain controller within our testbed in the Future SOC Lab. The use of a simulated dataset does not only allow to precisely estimate efficiency of our algorithm, but also to emulate normal user behaviour in cases, when cleaning up the training dataset could be considered too complex. Even though for cases, when real training dataset contains undetected malicious activity, so that our algorithm is unable to detect them later in the data, all new issues of malicious user behaviour, that were not captured into training dataset, will be detected.

We hope that the future integration of our Poisson-based anomaly detection approach into a SIEM system prototype like REAMS will significantly improve its ability to detect cases of malicious user behaviour and achieve higher level of security in the monitored network.

Sales Planning and Forecast

Frank Morelli
Pforzheim University of Applied Sciences
Tiefenbronnerstr. 65
75175 Pforzheim
frank.morelli@hs-pforzheim.de

Lukas Stahl
Pforzheim University of Applied Sciences
Tiefenbronnerstr. 65
75175 Pforzheim
lukas.stahl@gmx.de

Stefan Kerl
PIKON International Consulting Group
Kurt-Schumacher-Str. 28 - 30
66130 Saarbrücken
stefan.kerl@pikon.de

Abstract

This research project presents a sales planning and forecast solution based on proposals generated by SAP HANA. Major goal is to explore potential benefits derived by using this IT technology in comparison to classical sales planning in practice from a business´ perspective. In a concrete scenario for a fictitious SME company, a human sales planner interacts with SAP HANA and Predictive Analysis Library (PAL) features to create an accurate plan. Within the involved system landscape an automatic transformation of real-time ERP data into high level information in SAP BW on HANA takes place and external data is added. The prototype demonstrates the modus operandi of the interaction between a human sales planner and the active management support from SAP HANA based on the PAL methods.

2 Introduction

Central concern of operative planning activities within a company is to make strategic targets work. In contrast to the field of strategic planning it is based on a system of interrelated subplans typically combined with short-term activities. Sales planning, as a part of the integrated sales and operations planning process, continually achieves focus, alignment and synchronization among all functions of the sales organization. It is based on an (updated) forecast referring to past sales and trend analysis. The result has to be aggregated sales plan data in one uniform view comprising all relevant components and participants (e.g. groups, business units). Thus current planning and forecasting applications face the problem to deliver exact and detailed plan data on the one hand and reduce the user effort during the planning process to a minimum on the other.

This research project aims to implement and evaluate an IT solution employing the technical capabilities of SAP HANA optimizing a sales planning process related to detail level and accuracy of proposed plan data. It is based on an active support paradigm for an IT system within the field of analysis, data mining, forecast, projections, and simulations. The SAP system is designed to generate proposals which can be used by a human sales planner (in best case) without any adaption on a high (aggregated) and appropriate level. The human-computer-interaction is located on the level of SAP BW on HANA. A proposal of plan data is generated based on actuals of the past (loaded from an SAP ERP system) combined with external data by methods of the Predictive Analysis Library (PAL). Customer planning function types trigger the predictive analysis process and provide the result into the planning application. After check and adaption of proposed plan data a disaggregation to the required detail level takes place in real time. Saved plan data is therefore immediately available for further analysis (in other business areas like finance or production) which could lead again to adaption of plan data (closed loop).

Plan frequency and planning horizon depend on the specifics of each industry. Therefore a concrete branch, the discrete manufacturing, has been chosen. Companies within the discrete manufacturing industry can be categorized by the production of distinct and therefore countable items. As the fictive Global Bike Company (GBI) is typically utilized within universities having integrated SAP content in their programs (members of the SAP UA program), the corresponding structure and data seem to be a suitable example. In general it can be compared to the IDES, the “International Demonstration and Evaluation System”, created once by SAP to demonstrate various business scenarios executed in an SAP ERP system.

3 Business Scenario

The following subchapters describe the research project from a business´ perspective. They cover a characterization of the situation within a company intended to apply a sales planning application within SAP HANA including Predictive-Analytics-Methods of the PAL, a short explanation of the methodical

approach which was followed and a description of the created data model.

3.1 Initial Situation and Hypotheses

As a concrete example the fictional company Global Bike Inc. (GBI) has been chosen. It contains application data with many realistic characteristics for a bicycle-producing company and business processes that are designed to reflect real-life business requirements (e.g. order-to-cash processes and procurement-to-pay handling).

GBI has a complete story attached to it and comprises two companies located in the US and in Germany. The material spectrum includes trading goods, raw materials, semi-finished goods and finished goods. The actual detailed content, which requires license and use of SAP software to function, is available for SAP UA members. Within the research project the existing data structure has been enhanced for scenarios of make-to-stock production (production with no customer relation) as well as make-to-order processes. Talking hereinafter about GBI, the specific case created for this research project is meant.

The GBI produces eight different types of bicycles for the German market and sells exclusively to retailers. There are about 45 customers, each located in a different city within Germany. The overall market share stands permanently at about 30%. In order to satisfy customer's volume requirements and to optimize the company's inventory, the sales department wants to implement a forecasting solution whose results directly flow into the monthly sales planning process.

To examine how the PAL works and to demonstrate the interaction between the forecasting proposals of the PAL and the sales planning in SAP HANA, some central assumptions were made: First, the sale of different types of bicycles depends on the geographic location of the customers. And secondly, there is a correlation between actual sales and the corresponding weather situation. For the weather, past weather reports for the relevant period of time were used.

3.2 Methodical Approach

"Predictive Analytics as a subfield of Business Intelligence describes a type of preparation and evaluation of data to support the future decision making within all enterprise levels. The data mining is extended by the determination of forecast values to provide information about the future in order to improve the decision making." [1]

The basic idea of Predictive Analytics is to gain information and knowledge about the customer to improve the internal processes within a company. By applying mathematical algorithms and by using statistical instruments, the available internal and external data and information are combined. As a result, patterns in data are derived in order to forecast future probabilities by an acceptable level of reliability.

Figure 1, as the developed and used process model for the business case, shows the single steps to pass a process of Predictive Analytics and finally to provide the result into the planning application.

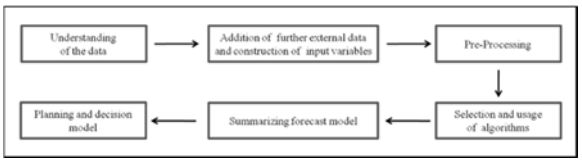


Figure 1: Methodical Approach of a Predictive Analytics Process

3.3 Data Model

Before building the prototype of the SAP BW on HANA system, an appropriate and consistent data model needed to be conceptualized. First of all, the actual sales data for the period from January 2008 to December 2014 were generated by the usage of a self-programmed ABAP tool. Starting point for the generation of the actual sales data was the total number of bicycles sold in Germany per year. All the market information has been drawn from official statistics of the Federal Statistical Office Germany. As a pre-assumption a solid market share of the GBI was set whereby the number of sales units per year was derived. This quantity was broken down into periods, customers and products. Hereby following influencing factors were taken into consideration:

- Number of inhabitants per state and city
- Percentage of bicycle traffic installations per state (fixed over the whole period)
- Actual monthly weather in the customer's city
- Geographical location of the customer's city

Following the breakdown of the yearly sales units per customer to the material groups (type of bicycles), the distribution to the single months took place through dividing the yearly amount by 12. Then weather and geographical position correction factors were applied to simulate the local physical characteristics of each customer and the consequential buying patterns with regard to the single material groups. In terms of the weather, a monthly rating per city was made, categorizing the weather in "good", "medium" and "bad" based on the corresponding actual temperature and rainfall. The geographical location of the cities was categorized into the four categories "urban flat", "urban mountainous", "rural flat", and "rural mountainous" based on the corresponding number of habitants and geographical altitude of each customer's city. By taking the average monthly weather and the geographical location of the customer's city into account, it was determined for each

material group how strong the number of sales react on the factors. In case of a reaction on weather and geographical location factors with regard to a specific material group (e.g. mountain bike), the normal sales changes positively or negatively by up to 30%. At this point, assumptions regarding the correlations were made. In the last step, the monthly sales numbers per material group were distributed to the individual articles (10 per material group). The first three articles within a material group assigned 70 % of the amount per material group, the next three further 20 % and the last four articles assigned to the remaining 10 %. The distribution was finally corrected by a random factor of +/- 1 %.

As soon as the data had been generated according to the procedure described above, the data was imported via an interface to the Profitability Analysis (CO-PA) module of the SAP ERP system. This was made as a substitute for the conventional way of creating CO-PA data out of invoices from the SD module in order to reduce the complexity of data generation within the research.

4 Planning Application

4.1 Concept

The main idea of the applied sales planning in this research project is to use the new capabilities of in-memory technology provided by SAP HANA within the planning and forecast process such as PAL, automatic disaggregation, in-memory and real-time access to SAP ERP data.

In the designed period of the planning application, the influence factors from a business´ perspective has to be determined from the very beginning. In this case, internal data from the company (e.g. historical sales volumes of previous years) and external data (e.g. past weather reports) will be considered in the data model. Depending on types and origins of input data, certain technical preparations have to be made and appropriate statistical methods provided by the PAL need to be chosen. As output from PAL algorithms highly significant forecast data on a detailed level will be delivered. This is expected to be the solid basis for further planning steps such as sales quantity and revenue planning.

For the sales planning type, the rolling sales planning for the next 24 months was chosen. The rolling sales planning works as an instrument to closely and dynamically intermesh the single sales plans over several planning periods.

As front-end tool, SAP BusinessObjects Analysis for Office is used for the implemented planning application. This application includes three different planning workbooks, one for the forecast proposals, the second for the sales quantity plan, and the third one for the sales revenue plan.

4.2 Functions

Generally, the planner can choose the planning detail level e.g. customer, material group, article and the monthly or yearly time period. The result area of the forecast workbook shows the actual quantities of the last seven years as reference.

The forecast comprises two different proposal functions: on the one hand it determines proposals only based on historical data of seven business years with a Time Series Algorithm (Exponential Smoothing) and on the other hand it is able to calculate proposals based on the weather expectations of the planner for the next month by means of a Regression Analysis. To generate forecast proposals, the Forecast Smoothing Algorithm was chosen. In the first step this algorithm automatically selects and calculates the optimal algorithm and corresponding parameters out of a set of smoothing functions, including Single Exponential Smoothing, Double Exponential Smoothing and Triple Exponential Smoothing. Exponential Smoothing is generally a time series technique that can be used with any discrete set of repeated measurements. The second step results in a concrete forecast proposal based on the algorithm found in step one, which is delivered on the detailed level and transferred to the planning environment of SAP BW on HANA.

In order to simulate the possible weather effects on the sales quantity, the planner chooses the month to be adapted, one of the weather categories (1=good, 2=medium, 3=bad) and the minimum coefficient of determination (between 0 and 1).This triggers a Regression Analysis (Linear Regression - LR) resulting in a regression model which displays indicated correlations between the specified parameters (sales quantity and weather category). Based on this the forecasting with LR Algorithm proposes a precise optimum forecast quantity if the minimum coefficient of determination is reached. Afterwards the planner can manually change the forecast values at his own discretion. In this case the planner changes for example the monthly overall sales forecast for one specific customer and one material group. Then, the proposed values of each article automatically change analogously to the previous weighted distribution.

As soon as the planner has finished the forecast steps, he saves the plan data and switches over to the sales quantity planning workbook. The planner needs to transfer the proposed forecast 1:1 by using a copy function. The forecast and the copied sales quantity are displayed in the result area of the sales quantity workbook. If necessary, the planner can make manual adjustments once again. If the adaptation takes place on a higher level, the plan data will be disaggregated to the detailed level. At this point of the

planning process, the planner is able to compare the proposed forecast generated by the system with the actuals and plan quantities. While saving in sales quantity workbook the data is also transferred to and visible in the sales revenue workbook. During the transfer, a calculation of the revenues takes place.

For the last step, the planner opens the sales revenue workbook. In addition to the sales quantities, the monthly prices for each article and the calculated plan revenues are displayed. The prices are maintained on material level within the ERP System and visualized in real-time. If required, the revenue can be adjusted and again, the respective effected changes in all levels take place analogously to the previous calculated weighted distribution. If the revenue is manually adjusted, sales quantities will be adjusted correspondingly. Revenues will be updated as well, if the quantity has been changed.

As result of the planning process, planning quantities and revenues are available on detailed level for further analysis and usage in other company processes.

5 Prototype

The research prototype includes the four systems SAP ERP on HANA, SAP BW on HANA, SAP BusinessObjects Server and a database for the external data. The first component of the prototype is a database where the information about the weather as well as the market data is extracted from. Within this database the external data is prepared and adapted to the needs of the business scenario. This database is connected via SAP BusinessObjects (BO) Data Services to both SAP systems, ERP on HANA and BW on HANA. Thus the data provisioning for generating the actuals (ABAP program on SAP ERP) as well as the data basis for the planning application (SAP BW on HANA) have been realized.

The implementation of the ABAP program for the generation of actuals is based on parameters (e.g. market share) to control the creation of actuals. It selects the external data from tables filled by BO Data Services and creates accordingly to the defined logic (see chapter 3.3) the actuals of the past. As a result the actuals are stored in a new database table. This table is used by the interface for CO-PA data creation. As preparatory work the master data in SAP ERP on HANA for 45 customers, 80 products and 8 material groups were created, as well as the operating concern “GL00” was customized and the BW Extractor for CO-PA transaction data was generated with all required fields. To deliver product prices in real-time, a calculation view was developed in SAP ERP on HANA by means of the SAP HANA Studio.

In SAP BW on HANA, a data model with new BW Objects was set up (see figure 2):

Starting in SAP ERP CO-PA, data was extracted via the SAP standard ETL process into an Advanced Datastore Object (Type Infocube). An Advanced Datastore Object is a new Infoprovider which combines functionalities of all classic providers like Infocubes or Datastore Objects in a single provider type. The Calculation View for prices was integrated by Smart Data Access (SDA) in combination with an Open ODS View. These are new components as well, that allow easily building up a virtual data model without persistent data storage in BW. During query runtime prices are directly read from SAP ERP. The plan data (generated by system or by manual user input) is stored into three real-time Infocubes. The separate basis providers have been combined in three Composite Provider of type union. On top of the composite providers, four aggregation levels were set up to build planning objects such as planning queries with SAP BEx Query Designer, planning workbooks (SAP BO Analysis for Office) and planning functions, planning filters or planning sequences. Within these planning functions (two of type Exit) the logic for proposal generation was implemented by using ABAP Managed Database Procedures (AMDP), which is the bridge between ABAP application server and HANA database. The Exit planning functions were implemented so that they can be executed completely in-memory. Planning functions were called from the planning workbook to determine for which customer / product combination (depending on user input) a proposal should be generated. Inside planning function, HANA stored procedures were called with following algorithms of the PAL: Forecast Smoothing (Time Series Algorithm) and Linear Regression (Regression Algorithm).

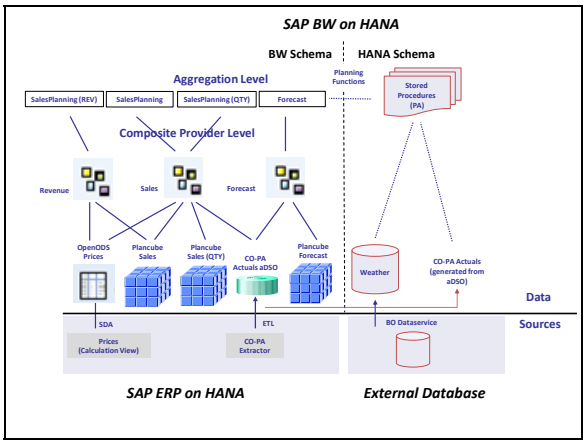


Figure 2: Data Model for the Research Project

The Forecast Smoothing Algorithm executes Single, Double and Triple Smoothing Algorithm several times to determine the optimal parameters for a current run. The result of this execution is a forecast of sales quantity on a detailed level.

The second algorithm (Linear Regression) takes the weather factor into account. Sales quantity in combination with weather category on level customer / article / period is passed to the algorithm. If the cal-

culated coefficient of determination reveals to be greater than the user defined value (set at runtime by planner), a forecast based on that regression is calculated. To proof the significance of result the f-value delivered by the regression algorithm is checked.

To allow the sales planner to change generated forecast manually, the data is copied from the forecast cube to the sales quantity cube by a SAP standard copy function (completely HANA optimized).

After possible manual adaption of sales quantities a last step of calculating / adapting sales revenue is processed by first copying data from the sales quantity cube to the final sales cube. After the data has been copied, the human planner can change quantities as well as revenues. Due to technical problems, instead of an inverse formula within the query definition, two planning functions (Type FOX Formula) were created to re-calculate quantities after revenue adaption and to re-calculate revenues after quantity changes, so that quantities, prices and revenues always fit together.

6 Conclusions

The major goal of the research project was to explore potential benefits of SAP HANA in combination with the PAL in comparison to classical sales planning in practice from a business´ perspective. According to this, the following statements summarize the insights of the research team:

One of the findings during the research project has been, that for figuring out which algorithm offered by the PAL is the most fitting one (regarding the specific business case), the user has to have professional and methodical know-how about the appropriate statistical and mathematical analysis as a prerequisite.

With SAP HANA, companies can use the PAL as a provided library which enables the planner to use optimized statistical algorithms for forecasts.

In general, planning processes are all very data and performance intensive. A way to increase the planning quality in all levels of details, is to simulate different scenarios with different planning processes and plan numbers. One of the main advantages of sales planning using SAP HANA is the high performance regarding the data processing of the used algorithms within the forecast calculation. Since the efforts in terms of installing the PAL algorithms and meeting the technical prerequisites has been made in advance, the effort during the original planning process is reduced significantly.

The usage of SAP HANA enables to integrate external data in the planning process by binding different databases employing SAP BO Data Services or including external data in real-time using virtual data models. This can increase the accuracy of plan data and therefore the efficiency of the company's performance processes.

Planning with an in-memory technology like SAP HANA leads to a substantial enhancement with regard to the performance during planning, especially in terms of disaggregation or aggregation of data. Consequently, the velocity as well as the dynamic and flexibility during planning increases. Mapping real-time data to simulate different scenarios in order to support management decisions helps the company to be competitive in a dynamic market environment.

Due to the high performance, companies, if necessary, are able to store mass data online instead of a time-delayed back-end-execution.

In this respect a “non-disruptive” and improved sales planning process can be implemented by using SAP HANA in combination with the PAL.

References

Comp. Acknowledgements

[1] C. Felden / C. Koschtial / J. Buder (2012), p.522. In: P. Mertens / S. Rässler (eds.,2012): Prognoserechnung, 7th edition, Springer, Heidelberg a.o.

The authors would like to thank to the active collaboration with SAP, and especially to the experts of PIKON Deutschland AG (Andreas Adam, Joschka Argast, Sebastian Broschart, Benjamin Duppe, Oliver Dworschak, Yuanyuan Gao, Jörg Hofmann, André Klos, Sarah Leichtweis, Fabian Mosbach, Carsten Promper, Christian Schlömer).

Project OliMP: In-Memory Planning with SAP HANA

Mariska Janz, Abdulmasih Hadaya and Ivaylo Ivanov
Department of Very Large Business Applications (VLBA)
Carl von Ossietzky University of Oldenburg
{mariska.janz, abdulmasih.hadaya, ivaylo.ivanov}@uni-oldenburg.de

Abstract

While dealing with planning and optimization it is important not only to perform strategic decisions, but also to adequately react to environmental changes. On operational planning and optimization level, the feedback must be provided in real-time and with high certainty. To do it in current situation, the in-memory solution should be used to accelerate the speed of the analysis and provide near real-time response. In details, beside collecting and performing ETL, the project is split into 3 major parts: (a) Posthumous Analysis; (b) Planning and Simulation; (c) Predictive Analytics.

1. Introduction

In-memory computing allows the processing of very large amounts of real-time data in the main memory of the server, so that results from analysis and transactions are immediately available.

How can the use of in-memory planning and forecasting tools provide a better simulation of future effects of today taking decisions in business questions? How can responsible actors be supported to take transparent decisions rather than subjective or feeling decisions? Can the technological advancement generate economic/business added value?

The goal of our project group "In-Memory Planning with SAP HANA"[1] is to answer the questions above and to obtain the following working skills:

- Software development
- Use of software development tools
- Writing reports and project documentation
- Teamwork and personal soft skills
- Use of SAP HANA to obtain fast in-memory computing
- Use of predictive and analytical tools like SAP Predictive Analysis

In the following, we will describe our findings and results from the seminar phase, our project idea, the next steps we plan to undertake and the Future SOC Lab resources we are using.

2. Project Idea

The research project "In-Memory Planning with SAP HANA" is focusing on operational planning and optimization. It is our agenda to improve the process of business planning with predictive data analysis. This can be done by using In-Memory-Technology [4]. This new technology is able to create much faster prediction when dealing with many data sources which is especially interesting for enterprises. In our special case of research, we deal with the estimation and calculation of energy consumption. Although this can be done by just using historical data, a more accurate result can be achieved by including further data. Concerning the scientific side of the project, it needs to be clarified how the predictions change with involvement of different combinations of data. This is done by using the SAP HANA-In-Memory-Database with SAP AFM [10].

3. Used Future SOC Lab resources

The Future SOC Lab provided us our own SAP HANA instance. We have used it for evaluating and testing first approaches to earn initial practical experience in how to handle the appliance. We implemented specific software tool using Java that allowed as to import data from the following sources: (a) Deutscher Wetterdienst (DWD) [1]; (b) European Network Of Transmission System Operators For Electricity (ENTSO-E) [3]; (c) European Energy Exchange (EEX) [2]; and (d) Calendar Data.
The delivered resources are significantly involved in the development process and needed for further tasks.

[1] For more information about our project group, see Appendix A.

Future SOC Lab delivers the indispensable basis for our project group work.

4. Hypotheses

The main hypothesis is that aggregating more data and using it as an input should lead to better forecast results. Three sub-hypotheses were built to prove or disprove the main hypothesis. The base hypothesis takes no factors into account and is based on historical data of energy consumption. The hypotheses from 1 to 3 consider additional factors that influence the energy consumption such as weather and energy prices with hypothesis 1 having the least factors and hypothesis 3 having the most factors. All hypotheses are based on building a model using data from 2009 till 2013 to deliver a forecast for January 2014 on an hourly basis. The first hypothesis considers calendar events next to the historical data of the energy consumption. Calendar events are thought to play a role regarding energy consumption. Official holidays and occasions could relate the more or less energy consumption. The second hypothesis adds the air temperature as a factor to the model. Weather conditions and air temperature in specific are also factors that probably affect the energy consumption. The third hypothesis adds the energy prices to the previous factors. Energy prices are thought to be a major factor that helps -when combined with the other factors- to build a model that has enough input to predict the future of the energy consumption.

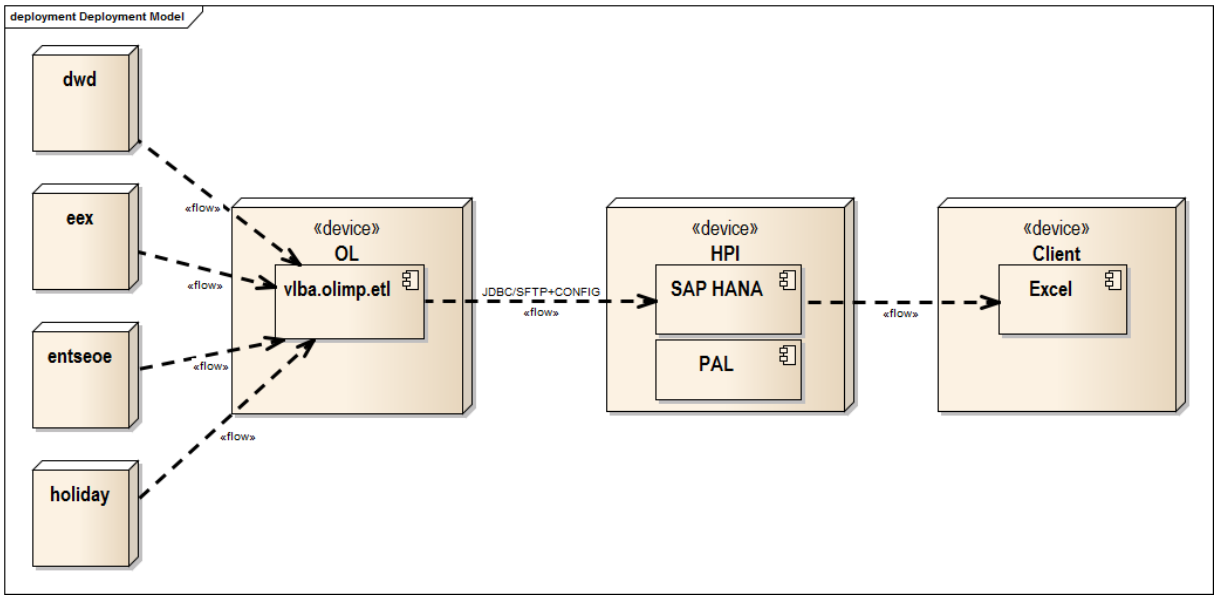


Figure 1. System architecture: Deployment view

5. Working method

In the previous section we saw that the main hypothesis depends on the base hypothesis and three sub-hypotheses to be proved or disproved. The next step would be to evaluate these sub-hypotheses and use the results as input to evaluate the main one. For that, analysis capabilities and prediction tools from SAP PAL were used to test the sub-hypothesis. For the base hypothesis the time series algorithms which predict the future of a factor depending on its past values and behavior were used [8, S. 225]. Different settings and parameters for each algorithm were implemented to enable different weighing of the trends in the data. The algorithms used for the first, second and the third hypothesis analyse the relations between the factors based on different mathematical models such as exponential regression, linear regression, multiple linear regression and other regression algorithms [8, S.156]. Also here, different settings were implemented to adjust the built model in a way that delivers the best forecast. The evaluation of the forecast was carried out through different measures. These measures give different sights on the forecast data compared to the actual data to assess the errors in the predictions, the goodness of the built models and the suitability of the models to predict further periods [6].

6. Data loading into SAP HANA

The following graphic illustrates the deployment view of the architecture. On the left side are the different data sources in different file formats. An ETL [5] process was used to bring data from various of data

sources to the centralized database, in our particular case in was SAP HANA. The further analysis was conducted mainly by means of the shipped within the SAP HANA prediction library known as Predictive Analysis Library (PAL) [8]. The evaluation of the goodness of the predictions was done on the client side with means spreadsheets processing tools such as Microsoft Excel.

7. Results

The assumption that more data can improve the prediction was true for tests with the algorithms such as Exponential Regression [7], Multiple Linear Regression[7] and Support Vector Machine [9]. But the enhancement that the addition of new factors contributed to the predictions was insignificant. The best prediction result was with the base hypothesis based on historical data of the energy consumption as tested with the algorithm linear regression with damped trend and seasonality. Based on hypothesis 3 no significant predictions compared to the other hypotheses were achieved. The following graphic shows the best run results with different algorithms using curves in different colors to show the actual and the predicted data:

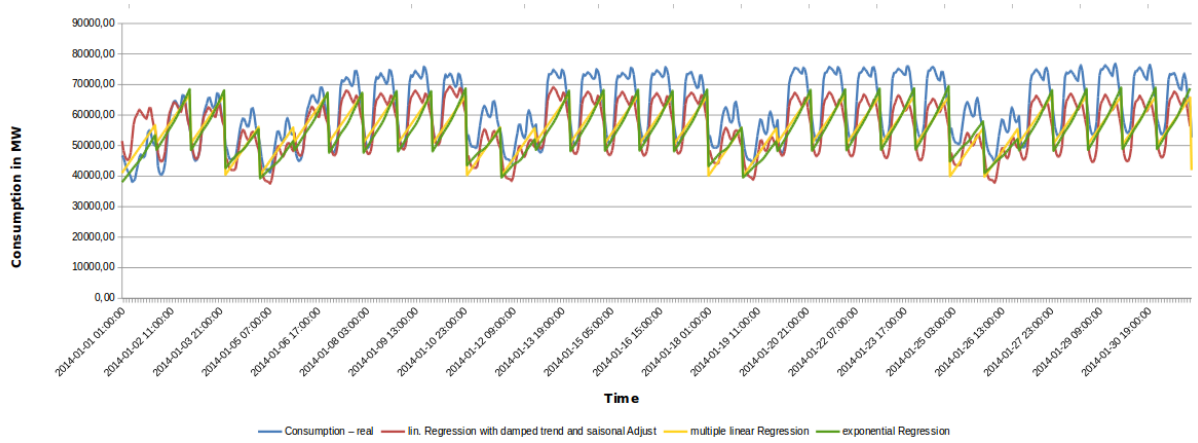


Figure 2. The three hypotheses compared to actual energy consumption

8. Future works

There are a plenty of possibilities for the enhancements on top of the obtained results. The achieved results can be significantly extended by trying out other analysis using other, non used in the current study, machine learning or data mining algorithms like Artificial Neural Networks [11]. As well as we would like to point out, that other data sources could enrich the input leading to better forecasts. New Factors, which might affect the energy consumption rate, could be figured ourt by generating more features and analytical dimensions from newly added, previously unseen, data sets.

References

[1] Deutscher wetterdienst (dwd), url: http://www.dwd.de/.

[2] European energy exchange (eex), url: https://www.eex.com/en/.

[3] European network of transmission system operators for electricity (entso-e), url: https://www.entsoe.eu/pages/default.aspx.

[4] Sap in-memory computing technology changing the way business intelligence is managed.

[5] F. N. Alexander Albrecht. Etl (extract-transform-load).

[6] K. Grace-Martin. Assessing the fit of regression models, May 2005.

[7] R. T. O'Connell and A. B. Koehler. *Forecasting, time series, and regression: An applied approach*, volume 4. South-Western Pub, 2005.

[8] SAP. Sap hana predictive analysis library (pal).

[9] B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[10] . S. SE and an SAP Affiliate Company. *SAP HANA Developer Guide*, volume SAP HANA Platform SPS 08, Document Version: 1.1 - 2014-08-21.

[11] B. Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

Appendix A. The Team and Work Organization

We are 11 students working in a project group at the Business Informatics faculty with five supervisors from the department of Very Large Business Applications (VLBA) at the Carl von Ossietzky University of Oldenburg (Germany). The project group is a compulsory activity for receiving a master's degree at our university.

Our project group called "OliMP[2] - In-Memory Planning with SAP HANA" started at the beginning of April 2014. The project group has a duration of one year and stops in the end of March 2015. Firstly, we did Massive Open Online Courses provided by openSAP and the HPI. The main objective of doing these online courses was to improve our knowledge skills in the SAP HANA technology. Every project member did the following online-courses:

- An Introduction to SAP HANA by Dr. Vishal Sikka
- Introduction to Software Development on SAP HANA by Thomas Jung
- In-Memory Data Management 2013 by Prof. Hasso Plattner

The first phase of our project consisted of specific course works: Every project member wrote a seminar paper about 10-15 pages about a specific topic. These topics are connected to different project perspectives in order to get an all-overview of the working package. Moreover every project member gave a presentation about his topic in order to share his knowledge with the other project members. So, all members of the project group have an equal level of knowledge. The topics of the seminar papers and the project members working on them are as follows:

- **Business-related processes**
  - Processes of planning in an enterprise from technical and organizational point of view and their goals (Igor Perelman)
  - Opportunities and challenges in the classic enterprise planning (Johannes Steffen Scheer)
  - Integrated enterprise planning systems (Farhad El-Yazdin)
- **Basics of SAP HANA and inMemory Computing**
  - Pros and Cons of InMemory-Computing (Rima Adhikari K.C.)
  - Analytical capabilities of SAP HANA: integration with R & Excel? (Eduard Rajski)
- **Planning processes and tools**
  - Statistical methods for updating historical data (Daniel Stratmann)
  - Estimation of the use of predictive methods and tools for SAP (SAP Predictive Analysis) (Jonas Schlemminger)
  - Planning and forecasting tools with their strengths and weaknesses using the example of the systems SEM-BPS, BW-IP and BPC by SAP AG (Abdulmasih Hadaya and Mariska Janz)
- **Development tools and organisation methods of projects**
  - Design Thinking (Ivaylo Ivanov)
  - Classical vs. Agile Software Development: Using Scrum in the project group (Benjamin Hemken)

An important topic in connection with team work and organization was to find an appropriate process model for the project. This project contains risks related to the complexity of the potential solution. Another important fact is that the requirements can be changed during the development process. To be able to react to such changes we use the agile process model SCRUM.

[2] OliMP is a acronym for **Ol**denburger **in**-**M**emory **P**lanung (http://www.ol-imp.de/)

OntQA-Replica: Intelligent Data Replication for Ontology-Based Query Answering

Lena Wiese
Research Group Knowledge Engineering
Institut für Informatik
Georg-August-Universität Göttingen
Goldschmidtstraße 7
37077 Göttingen
wiese@cs.uni-goettingen.de

Abstract

The OntQA-Replica project aims to improve the performance of ontology-based query answering in distributed databases by employing a preprocessing procedure (including a clustering step and a fragmentation step): for efficient query answering, data records that are semantically related are grouped in the same data fragment based on a notion of similarity in the ontology. At the same time, the OntQA-Replica project supports an intelligent data replication approach that minimizes the amount of resources used and enables an efficient recovery in case of server failures. The OntQA-Replica project aims to show scalability and failure tolerance of this intelligent query answering approach by developing novel replication schemes for several fragmentations with overlapping fragments.

1. Introduction

In the era of "big data" huge data sets usually cannot be stored on a single server any longer. Cloud storage (where data are stored in a cloud infrastructure) offers the advantage of flexibly adapting the amount of used storage based on the growing or shrinking storage demands of the data owners. In a cloud storage system, a distributed database management system (DDBMS) can be used to manage the data in a network of servers. The decisive features are replication (for recovery and scalability purposes) and load balancing (data distribution according to the capacities of servers). On the other hand, flexible query answering [1] offers mechanisms to intelligently answer user queries going beyond conventional exact query answering. If a database system is not able to find an exactly matching answer, the query is said to be a failing query. Conventional database systems usually return an empty answer to a failing query. In most cases, this is an undesirable situation for the user, because he has to revise his query and send the revised query to the database system in order to get some information from the database. In contrast, flexible query answering systems internally revise failing user queries themselves and by evaluating the revised query return answers to the user that are more informative for the user than just an empty answer. One way to obtain such informative answers is to use an ontology to return answers that are related to the original search term according to some notion of similarity. For example, in an electronic health record, when searching for the term cough, the terms bronchitis and asthma might be similar to cough and might be returned as related answers. Unfortunately, finding related answers at runtime by consulting the ontology for each query is highly inefficient.

2. Ontology-Based Fragmentation

In previous work [3], a clustering procedure was applied to partition the original tables into fragments based on a *relaxation attribute* chosen for anti-instantiation. Finding these fragments is achieved by grouping (that is, *clustering*) the values of the respective table column (corresponding to the relaxation attribute) and then splitting the table into fragments according to the clusters found.
We assume that each of the clusterings (and hence the corresponding fragmentation) is *complete*: every value in the column is assigned to one cluster and hence every tuple is assigned to one fragment. We also assume that each clustering and each fragmentation are also *non-redundant*: every value is assigned to exactly one cluster and every tuple belongs to exactly one fragment (for one of the relaxation attributes); in other words, the fragments inside one fragmentation do not overlap. More formally, we apply the clustering approach described in [3] (or any other method to semantically split the attribute domain into subsets) on the relaxation attribute, so that each cluster inside one cluster-

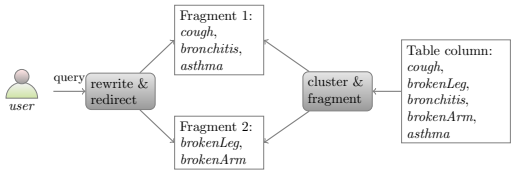


Figure 1. Ontology-based fragmentation

ing is represented by a *head* term (also called prototype) and each term in a cluster has a similarity *sim* to the cluster head above a certain threshold α. We then obtain a clustering-based fragmentation for the original table F into fragments.

3. Ontology-driven Query Answering

To enable ontology-driven query answering, when a user sends a query to the database, the term (that is, constant) that can be anti-instantiated has to be extracted, the matching cluster has to be identified and then the user query has to be rewritten to return answers covering the entire cluster.

3.1. Metadata and Test Dataset

In order to manage the fragmentation, several metadata tables are maintained:

- A **root** table stores an ID for each cluster (column *clusterid*) as well as the cluster head (column *head*) and the name of the server that hosts the cluster (column *serverid*).
- A **lookup** table stores for each cluster ID (column *clusterid*) the tuple IDs (column *tupleid*) of those tuples that constitute the clustered fragment.
- A **similarities** table stores for each head term (column *head*) and each other term (column *term*) that occurs in the active domain of the corresponding relaxation attribute a similarity value between 0 and 1 (column *sim*).

Our prototype implementation – the OntQA-Replica system – runs on a distributed SAP HANA installation with 3 database server nodes provided by the Future SOC Lab of Hasso Plattner Institute. All runtime measurements are taken as the median of several (at least 5) runs per experiment.
The example data set consists of a table (called *ill*) that resembles a medical health record and is based on the set of Medical Subject Headings (MeSH [2]). The table contains as columns an artificial, sequential *tupleid*, a random *patientid*, and a *disease* chosen from the MeSH data set as well as the *concept* identifier of the MeSH entry. We varied the table sizes during our test runs. The smallest table consists of 56,341 rows (one row for each MeSH term), a medium-sized table of 1,802,912 rows and the largest of 14,423,296 rows (obtained by duplicating the original data set 5 times and 8 times, respectively). A clustering is executed on the MeSH data based on the concept identifier (which orders the MeSH terms in a tree); in other words, entries from the same subconcept belong to the same cluster. One fragmentation (the clustered fragmentation) was obtained from this clustering and consists of 117 fragments which are each stored in a smaller table called *ill-i* where i is the cluster ID. To allow for a comparison and a test of the recovery strategy, another fragmentation of the table was done using round robin resulting in a table called *ill-rr*; this distributes the data among the database servers in chunks of equal size without considering their semantic relationship; these fragments have an extra column called *clusterid*.

3.2. Identifying matching Clusters

Flexible Query Answering intends to return those terms belonging to the same cluster as the query term as informative answers. Before being able to return the related terms, we hence have to identify the matching cluster: that is, the ID of the cluster the head of which has the *highest* similarity to the query term. We do this by consulting the similarities table and the root table. The relaxation term t is extracted from the query and then the top-1 entry of the similarities table is obtained when ordering the similarities in descending order:

```
SELECT TOP 1 root.clusterid
 FROM root, similarities
 WHERE similarities.term='t'
 AND similarities.head = root.head
 ORDER BY similarities.sim DESC
```

The query was tested on similarities tables of sizes 56341 entries, 14423296 entries and 72116480 entries. The runtime measurements show a decent performance of at most 125 ms impact even the largest table size.

3.3. Query Rewriting Strategies

After having obtained the ID of the matching cluster, the original query has to be rewritten in order to consider all the related terms as valid answers. We tested and compared three query rewriting procedures:

- lookup table: the first rewriting approach uses the lookup table to retrieve the tuple IDs of the corresponding rows and executes a JOIN on table *ill*.
- extra clusterid column: the next approach relies on the round robin table and retrieves all relevant tuples based on a selection predicate on the clusterid column.
- clustered fragmentation: the last rewriting approach replaces the occurrences of the *ill* table by the corresponding *ill-i* table for clusterid i.

Assume the user sends a query

```
SELECT mesh,concept,patientid,tupleid
 FROM ill WHERE mesh ='cough'.
```

and 101 is the ID of the cluster containing cough. In the first strategy (lookup table) the rewritten query is

```
SELECT mesh,concept,patientid,tupleid
 FROM ill JOIN lookup
 ON (lookup.tupleid = ill.tupleid
 AND lookup.clusterid=101).
```

In the second strategy (extra clusterid column) the rewritten query is

```
SELECT mesh,concept,patientid,tupleid
 FROM ill-rr WHERE clusterid=101
```

In the third strategy (clustered fragmentation), the rewritten query is

```
SELECT mesh,concept,patientid,tupleid
 FROM ill-101
```

In the small *ill* table with 56341 entries, 90 related answers are obtained, in the medium-sized *ill* table with 1802912 entries, 2880 related answers are obtained and in the large *ill* table with 14423296 entries, 23040 related answers are obtained. The runtime measurements in particular show that the lookup table approach does not scale with increasing data set size.

4. Query Answering with Derived Fragments

While the evaluation of a selection query on a single table shows a similar performance for all rewriting strategies, the evaluations of queries on two tables using a distributed JOIN show a performance impact for the first two strategies when the secondary table is large. We tested a JOIN on the patient ID with a secondary table called *info* having a column *address*. The original query is

```
SELECT a.mesh,a.concept,a.patientid,
 a.tupleid,b.address
 FROM ill AS a,info AS b
 WHERE mesh='cough'
 AND b.patientid= a.patientid
```

We devised two test runs: test run one uses a small secondary table (each patient ID occurs only once) and test run two uses a large secondary table (each patient ID occurs 256 times). For the first rewriting strategy (lookup table) the secondary table is a non-fragmented table. For the second strategy, the secondary table is distributed in round robin fashion, too. For the last rewriting strategy, the secondary table is fragmented into a derived fragmentation: whenever a patient ID occurs in some fragment in the *ill-i* table, then the corresponding tuples in the secondary table are stored in a fragment *info-i* on the same server as the primary fragment.

In the first strategy (lookup table) the rewritten query is

```
SELECT a.mesh,a.concept,a.patientid,
 a.tupleid,b.address,lookup.clusterid
 FROM ill AS a,info AS b,lookup
 WHERE lookup.tupleid=a.tupleid
 AND lookup.clusterid=101
 AND b.patientid= a.patientid.
```

In the second strategy (extra clusterid column) the rewritten query is

```
SELECT a.mesh,a.concept,a.patientid,
 a.tupleid,b.address
 FROM ill AS a,info AS b
 WHERE a.clusterid=101
 AND b.patientid=a.patientid.
```

In the third strategy (clustered fragmentation), the rewritten query is

```
SELECT a.mesh,a.concept,a.patientid,
 a.tupleid,b.address
 FROM ill-101 AS a
 JOIN info-101 AS b
 ON (a.patientid=b.patientid).
```

A small secondary table does not make much of a difference when executing the join operation (one matching tuple in the secondary table for each tuple in the primary table). However, for the larger secondary table (256 matching tuples in the secondary table for each tuple in the primary table), the impact of the lookup table access is huge in the case of the largest *ill* table with 14423296 entries.

5. Insertions

We tested the update behavior for all three rewriting strategies by inserting 117 new rows (one for each cluster). Any insertion requires identifying the matching cluster i again (see Section 3.2). Each insertion query looks like this for mesh term m, concept c, patientid 1 and tupleid 1:

```
INSERT INTO ill
 VALUES ('m','c',1,1).
```

In the first rewriting strategy, the lookup table has to be updated, too, so that two insertion queries are executed:

```
INSERT INTO ill
 VALUES ('m','c',1,1).
INSERT INTO lookup
 VALUES (i,1).
```

For the second rewriting strategy, the extra clusterid column contains the identified cluster i:

```
INSERT INTO ill-rr
 VALUES ('m','c',1,1,i).
```

For the third rewriting strategy, the matching clustered fragment is updated:

```
INSERT INTO ill-i
 VALUES ('m','c',1,1).
```

Here the lookup table approach has a huge runtime impact due to the maintenance of the lookup table entries.

6. Deletions

After the insertions we made a similar test by deleting the newly added tuples by issuing the query
`DELETE FROM ill WHERE mesh='m'.`
In the first rewriting strategy, the corresponding row in the lookup table has to be deleted, too, so that now first the corresponding tuple id of the to-be-deleted row has to be obtained and then two deletion queries are executed:

```
DELETE FROM lookup
 WHERE lookup.tupleid
 IN (SELECT ill.tupleid FROM ill
 WHERE mesh='m').
DELETE FROM ill WHERE mesh='m'
```

For the second rewriting strategy, no modification is necessary apart from replacing the table name and no clusterid is needed:
`DELETE FROM ill-rr WHERE mesh='m'`
For the third rewriting strategy, the matching clustered fragment i is accessed which has to be identified first (as in Section 3.2):
`DELETE FROM ill-`i `WHERE mesh='m'`
Interestingly, even the round robin approach with extra clusterid does not perform well on the largest data set.

7. Recovery

Lastly, we tested how long it takes to recover the clustered fragmentation by either using the lookup table or the extra column ID. The recovery procedure was executed first on the original table and the lookup table by running for each cluster i:

```
INSERT INTO c_i SELECT * FROM ill
 JOIN lookup
 ON (lookup.tupleid=ill.tupleid)
 WHERE lookup.clusterid=i
```

for each cluster i. Then, the recovery procedure was executed on the round robin fragmented table with the extra clusterid column for each cluster i:

```
INSERT INTO c_i
 SELECT * FROM ill-rr
 WHERE clusterid=i
```

While for the smallest and the largest table the two approaches perform nearly identically, for the medium-sized table the extra cluster id approach offers some benefit.

8. Overlaps and multiple relaxation attributes

So far, for the clustering approach only a *single* relaxation attribute has been considered. In order to support flexible query answering on multiple columns, one table can be fragmented multiple times (by clustering different columns); that is, we can choose more than one relaxation attribute. In this case, several fragmentations will be obtained. More formally, if α relaxation attributes are chosen and clustered, then we obtain α fragmentations F_l ($l = 1 \ldots \alpha$) of the same table; each fragmentation contains fragments $f_{l,s}$ where index s depends on the number of clusters found: if n_l clusters are found, then $F_l = \{f_{l,1}, \ldots, f_{l,n_l}\}$. Due to completeness, every tuple is contained in exactly one of the fragments of each of the α fragmentations: for any tuple j, if α relaxation attributes are chosen and clustered, then in any fragmentation F_l ($l = 1 \ldots \alpha$) there is a fragment $f_{l,s}$ such that tuple $j \in f_{l,s}$. Another fragmentation (the "range-based" fragmentation) is based on ranges of the patient ID and consists of 6 fragments for the small table, 19 for the medium-sized table and 145 for the large table; these fragments have an extra column called rangeid.
For the replication procedure, first the overlapping fragments (the "conflicts") are identified by using

```
SELECT DISTINCT clusterid, rangeid
 FROM c_i  JOIN c_i
 ON (r_j.tupleid=r_j.tupleid)
```

for each clustered fragment c_i and each range-based fragment r_j. Afterwards based on the conflicts a Bin Packing Problem with Conflicts (BPPC) problem is generated to enforce that overlapping fragments are placed on a different server. Based on the obtained solution, the fragments are moved to different servers by using

```
ALTER TABLE c_i
 MOVE TO 'severname' PHYSICAL.
```

9. Conclusion and Future Work

Due to the small size of the partitioned tables, the runtime performance is best for the clustered partitioning approach and the overhead of metadata management is negligible. It outperforms the lookup table approach that stores for each cluster the corresponding tuple IDs does not scale well as the data set size grows. In addition, the ontology-driven partitioning enables fine-grained load balancing and data locality: less servers have to be accessed when answering queries or updating tables. The idea of data locality can even be carried further by considering cluster affinity: if two clusters are accessed together frequently, their corresponding partitions can be placed on the same server. So far we did not address the dynamic adaptation of the clustering: whenever values are inserted or deleted, the clustering procedure on the entire data set might lead to different clusters. A particular problem that must be handled is the deletion of the head of a cluster: a new cluster head must be chosen before the current head can be deleted; in the simplest case, the term that is most similar to the previous head is chosen as the new head.
Similarly, deletions and insertions lead to shrinking or growing partitions. Hence in some situation it might

be useful to merge two smaller partitions that are semantically close to each other; or to repartition a larger partition into subpartitions based on a clustering of values of the relaxation attribute in the partition.
Another major research question that remains is how to parallelize the clustering step on multiple servers.

References

[1] K. Inoue and L. Wiese. Generalizing conjunctive queries for informative answers. In *Flexible Query Answering Systems*, pages 1–12. Springer, 2011.

[2] U.S. National Library of Medicine. Medical subject headings. http://www.nlm.nih.gov/mesh/.

[3] L. Wiese. Clustering-based fragmentation and data replication for flexible query answering in distributed databases. *Journal of Cloud Computing*, 3(1):1–15, 2014.

Natural Language Processing for In-Memory Databases: an Application to Biomedical Question Answering

Mariana Neves
Hasso-Plattner-Institute at the University of Potsdam, Germany
mariana.neves@hpi.de

Abstract

We currently face a deluge of textual documents which demands real-time processing for various natural language processing (NLP) applications. In-memory database (IMDB) technology has the potential of allowing traditional methods to be scaled for large document collections and speeding up the processing. In this report, I give an overview of my current efforts in utilizing IMDB for natural language processing tasks, and in particular for biomedical question answering.

1. Introduction

The current data deluge demands fast and real-time processing of large datasets to support various applications, also for textual data, such as scientific publications, Web pages or messages in the social media. Natural language processing (NLP) [7] is the field of automatically processing textual documents and includes a variety of tasks:

- tokenization: delimitation of words;
- part-of-speech tagging: assignment of syntactic categories to words;
- chunking: delimitation of phrases;
- and syntactic parsing: construction of syntactic tree for a sentence.

Further, NLP also involves semantic-related tasks:

- named-entity recognition: delimitation of predefined entity types, e.g., person and organization names;
- relation extraction: identification of pre-defined relations from text;
- and semantic role labeling: determining predefined semantic arguments.

Processing and semantically annotating large textual collection is a time-consuming and tiresome task which requires integration of various tools [11]. In-memory database (IMDB) technology comes as an alternative given its ability to quickly process large document collections in real time [13]. Indeed, it has already been efficiently used for a variety of NLP applications [9, 12, 6] but it still lacks more advanced NLP functionalities which could allow its use for a broader range of tasks as well as providing more precise analysis.

I propose the implementation of new methods for some of the NLP tasks introduced above, namely chunking, semantic role labeling and relation extraction. Further, I suggest utilizing these methods for applications in biomedicine, a field that currently demands both real-time processing of large textual collections as well as deliver of precise results. This report will describe the architecture and the preliminary results on the use of IMDB in the development of a question answering system for biomedicine.

2. Question answering

Question answering (QA) [7] is the task of posing questions, instead of only keywords, to a system and getting exact answers in return, instead of relevant documents. For instance, for the question "What disease is mirtazapine predominantly used for?" derived from the BioASQ dataset[1], the user expects to receive one or more disease names in return, e.g., "major depression", instead of documents which match the question keywords. Besides providing exact answers, QA systems can also return tailored summaries for the questions, thus, giving more details and context for the answer. For instance, the following passage is an adequate short summary for the above question: "Mirtazapine is predominantly used in the treatment of major depression.".

Question answering systems for biomedicine [1] are usually restricted to documents specific for the

[1] http://bioasq.org/

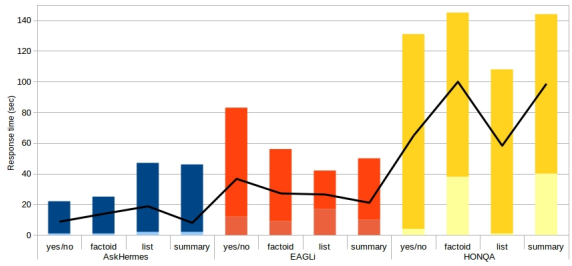


Figure 1. Time response from three question answering systems for biomedicine, e.g., AskHermes (blue), EAGLi (red) and HONQA (yellow), for 40 randomly selected questions from the BioASQ dataset.

domain, such as scientific publications from PubMed[2] and reliable Web sites. As far as I know, there are currently three available systems for biomedical QA, namely: AskHermes [2], EAGLi [5] and HONQA [4]. I have recently carried out an evaluation on both the quality of the answers and the response times of these systems and results shown that only five questions out of 40 could be appropriately answered and time responses varied from a couple of seconds to more than two minutes (cf. Figure 2.1).

In the following section, I will describe the architecture of of question answering system which is being built on the top of a SAP HANA instance of 1 Tb of memory provided by the FutureSOC[3]. at the Hasso-Plattner Institute.

2.1. Preliminary methods and results

Question answering systems are usually composed of three components [1]: question processing, passage processing and answer processing. My previous work [9] has relied on external resources for document retrieval, BioPortal[4] for query expansion and the IMDB technology for passage retrieval. Due to the impossibility of indexing the complete PubMed collection, I had to rely on a limited number of documents which I previously retrieved from PubMed using keywords-based queries.

I carried out an evaluation based on the more than 800 questions derived from the BioASQ challenge[5], an EU-funded project which aims to foster improvements on biomedical QA. The BioASQ dataset includes manually annotated information for relevant documents, passages and answers. Evaluation is given in terms of precision, recall, f-measure and MAP and is performed by on Oracle system available for registered users. My preliminary approach participated in the 2014 edition of the BioASQ challenge, and it is currently taking part in the 2015 edition of the challenge. I could recently obtain top results for snippet retrieval and the highest recall over all participating systems [9, 10].

Provided the inadequacy of querying the PubMed database on real-time applications, our current approach relies on a local copy of PubMed which is currently being integrated into a SAP HANA instance of 1 Tb of memory of the HPI Future SOC lab. Figure 2.1 shows the architecture of our QA system which includes integration of biomedical documents as well as various biomedical databases and ontologies.

For the moment, I have indexed more than 7,2 millions of the more than 24 millions citations included in PubMed. I currently consider only titles and abstracts of the documents and these entries are indexed in SAP HANA using the full text indexing feature, which automatically splits the text into sentences and the later into tokens. The indexed documents currently occupy more than 300 Gb of memory in HANA and comprises of more than 1,7 billion tokens and 68 millions sentences.

I have performed preliminary experiments with this incomplete copy of PubMed, which covers less than 80% of the documents referred in the snippets curated in the BioASQ datasets. In comparison to the results of my previous approach, I have obtained lower values both for precision and recall when querying for passages based only on keywords, This outcome suggests that a different approach is necessary when dealing with this huge collection. Improvements of the results requires implementation of more advanced NLP features for precisely retrieving passages which not only cite the words mentioned in the questions but also cites these words in similar contexts as those from the questions. Additionally, recall of the system can be improved through integration of domain knowledge into the methods, as the biomedical field is known for the high complexity and variability of the nomenclature of the entities.

2.2. Future work

Future work will include development of NLP features still not included in SAP HANA, as well as adaptation of existing NLP methods for the biomedical domain. The implementation will rely on supervised learning approach, e.g., support vector machines, and on available corpora from various domains, as summarized below:

[2] http://www.ncbi.nlm.nih.gov/pubmed
[3] http://hpi.de/en/research/future-soc-lab.html
[4] http://bioportal.bioontology.org/
[5] http://bioasq.org/

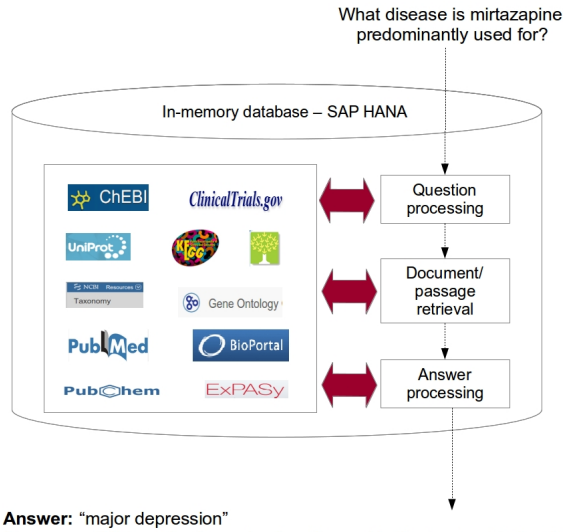


Figure 2. Architecture of a question answering which integrates various biomedical resources on a in-memory database.

- part-of-speech tagging and chunking: training and evaluation based on the GENIA corpus [8];
- semantic role labeling: training and evaluation based on the BioProp corpus [3].

Further, future work will also include development of the answer processing component which is still missing in our current system, improvements to the existing ones, use of the advanced NLP techniques described above and integration into the HANA database, as summarized below:

- question processing based on IMDB technology and implementation of methods for identifying the question type, e.g., yes/no, factoid or summary, and the expected answer, e.g., a gene or a disease;
- passage retrieval using the complete PubMed collection and previously identified named entities;
- answer processing based on IMDB technology, including exact answers and summaries.

References

[1] S. J. Athenikos and H. Han. Biomedical question answering: A survey. *Computer Methods and Programs in Biomedicine*, 99(1):1 – 24, 2010.

[2] Y. Cao, F. Liu, P. Simpson, L. D. Antieau, A. S. Bennett, J. J. Cimino, J. W. Ely, and H. Yu. Askhermes: An online question answering system for complex clinical questions. *Journal of Biomedical Informatics*, 44(2):277–288, 2011.

[3] W.-C. Chou, R. T.-H. Tsai, Y.-S. Su, W. Ku, T.-Y. Sung, and W.-L. Hsu. A semi-automatic method for annotating a biomedical proposition bank. In *LAC '06 Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*, 2006.

[4] S. Cruchet, A. Gaudinat, T. Rindflesch, and C. Boyer. What about trust in the question answering world? In *Proceedings of the AMIA Annual Symposium*, pages 1–5, San Francisco, USA, 2009.

[5] J. Gobeill, E. Patsche, D. Theodoro, A.-L. Veuthey, C. Lovis, and P. Ruch. Question answering for biology and medicine. In *Information Technology and Applications in Biomedicine, 2009. ITAB 2009. 9th International Conference on*, pages 1–5, 2009.

[6] K. Herbst, C. Fhnrich, M. Neves, and M.-P. Schapranow. Applying in-memory technology for automatic template filling in the clinical domain. In *CLEF Working Notes*, 2014.

[7] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall International, 2 revised edition, 2013.

[8] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182, 2003.

[9] M. Neves. HPI in-memory-based database system in task 2b of bioasq. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, pages 1337–1347, 2014.

[10] M. Neves. In-memory database for passage retrieval in biomedical question answering. *Journal Of Biomedical Semantics*, (submitted), 2015.

[11] M. Neves, A. Damaschun, N. Mah, F. Lekschas, S. Seltmann, H. Stachelscheid, J.-F. Fontaine, A. Kurtz, and U. Leser. Preliminary evaluation of the cellfinder literature curation pipeline for gene expression in kidney cells and anatomical parts. *Database*, 2013, 2013.

[12] M. Neves, K. Herbst, M. Uflacker, and H. Plattner. Preliminary evaluation of passage retrieval in biomedical multilingual question answering. In *Proceedings of the Fourth Workshop on Building and Evaluation Resources for Biomedical Text Mining (BioTxtM 2014) at Language Resources and Evaluation (LREC) 2014*, 2014.

[13] H. Plattner. *A Course in In-Memory Data Management: The Inner Mechanics of In-Memory Databases*. Springer, 1st edition, 2013.

Provision of Analyze Genomes Services in a Federated In-Memory Database System for Life Sciences

Matthieu-P. Schapranow, Cindy Fähnrich
Hasso Plattner Institute
Enterprise Platform and Integration Concepts
August–Bebel–Str. 88
14482 Potsdam, Germany
{schapranow|cindy.faehnrich}@hpi.de

Abstract

Next-generation sequencing generates large amounts of detailed diagnostic data within hours. We consider it as sensitive data, which must not be exchanged with public cloud infrastructures. Our Analyze Genomes cloud platform enables researchers and clinicians to design individual analysis pipelines and choose from a wide range of analysis tools without the need to have software and bioinformatics experts on site. In the given contribution, we developed a federated in-memory system connecting central Future SOC lab resources and external research facility in a hybrid cloud approach eliminating the need for transfer of sensitive data. Thus, we were able to provide analysis services as a service provider whilst sensitive data resides on local sites for processing.

1. Project Idea

In the scope of the Analyze Genomes project, we have built an analysis platform providing tools for setting up and executing analysis pipelines, assessing their results, and combining these with scientific data from distributed data sources [10]. We provide a modeling environment to create customized pipelines and choose from a range of ready-to-use third-party tools, which are executed in our own, distributed processing framework incorporating cloud computing and In-Memory Database (IMDB) technology to accelerate processing of genome data [6].

Today, data processing in cloud environments requires the transfer of data from local sites to our shared computing resources prior to its execution. However, processing of Next-Generation Sequencing (NGS) data easily involves 750 GB and more per patient sample [9]. Thus, transferring of raw NGS data to cloud resources requires a significant amount of time prior to its execution. Furthermore, we consider NGS data as sensitive information, which must remain on the premises of the research facility for reasons of data protection.

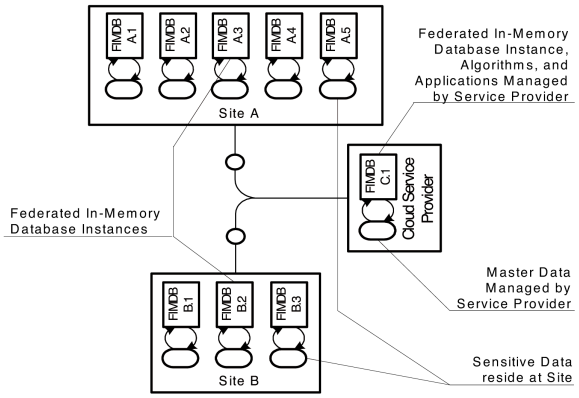


Figure 1: Computing resources and data reside on local sites whilst the service provider manages algorithms and apps remotely in our system.

In the given application scenario, we developed a Federated In-Memory Database (FIMDB) system combining local computing resources of research facilities and Analyze Genomes services as managed services in a unique hybrid cloud approach as depicted in Figure 1. NGS data are created at decentralized research sites or sequencing centers and must not be transferred to a central site due to legal restrictions for the use of the acquired patient-specific data. However, for assessment of treatment alternatives, the comparison of the concrete patient case with a spectrum of similar patient cases, e.g. similar patient history or identical diagnosis stored at individual partner sites, is required. Individual research sites and the services provider site can consist of multiple computing nodes forming together the FIMDB system.

Based on the concrete requirements of this use

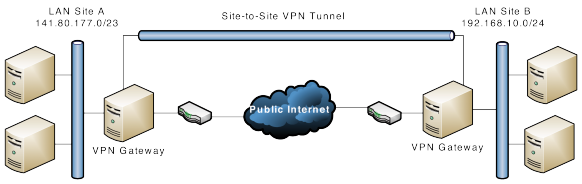


Figure 2: Site-to-site VPN connection interconnecting the two local area networks of site A and site B.

case, we attempt to build a unique cloud setup integrating decentralized computing resources to form a federated in-memory database system. The computing resources and data reside on local sites, whilst the service provider manages algorithms and apps remotely in our system.
In cooperation with a research facility in Berlin, we have started to set up our FIMDB, by connecting their cloud infrastructure with the Future SOC's resources. By that, our partner shall be able to use Analyze Genomes services while their data resides on their local computing resources. In the following, we document our current progress in setting up such a system and outline further steps that need to be undertaken.

2. Realization

In the following, we describe all steps that have been accomplished so far. To this belongs the setup of a Virtual Private Network (VPN) connection between the Future SOC and the research facility's network, the configuration of the remote directories, and the installation of the databases at the facility's local computing nodes.

2.1 VPN Connection

The network team of the research facility's local IT department needs to install and configure a VPN client. For our application scenario, we used and configured OpenVPN version 2.3.5 to establishing a secured bidirectional site-to-site VPN tunnel. The VPN tunnel connects local area networks at site A and site B via the public Internet as depicted in Figure 2. In the typical VPN setup, multiple VPN clients connect to a corporate network via a single VPN server, i.e. the corporate network is extended and clients consume corporate services in the same way they would access them being physically connected to the corporate network. In contrast, the site-to-site setup used by us connects multiple Local Area Networks (LANs) with each other, i.e. multiple LANs are connected forming a dedicated virtual network across all network topologies. By that, we are able to create any kind of point-to-point connections. In the given application scenario, the local research facility configured a single system as gateway system for the VPN connection while the updated network routes were pushed to individual computing nodes. Thus, the configuration efforts were minimized while a single point of maintenance was established.

2.2 Configuring the Remote Services Directory

The managed service provider grants the local research facility access to required algorithms that provided by the Analyze Genomes FIMDB system. In the application scenario, our services are either file- or database-based. We expose the file-based services, such as the alignment algorithms Burrows-Wheeler Aligner (BWA) or Bowtie, as runtime binaries in a remote service directory using a Network File System (NFS) [4, 3]. Thus, our service consumer needs to create a local mount point and add the configuration for automatically mounting the remote service directory to all of his local computing nodes. In the given scenario, the local sites integrated the configuration in their central configuration scripts, meaning that deployment to all involved nodes was a single step. Database-based services, e.g. stored procedures or analytical queries, are deployed via the shared database landscape system that has been set up prior. By that, our database-based services become automatically available once the local database instances are connected to Analyze Genomes' database landscape. Thus, no dedicated configuration steps were required.

2.3 Database Install

In order to use the services provided by Analyze Genomes on the local site of the research facility, we need to set up local IMDB instances that are then connected to the shared database landscape of Analyze Genomes forming the FIMDB system. Therefore, we need to install and configure an individual database instance on each of the research facility's local computing node. We incorporate SAP HANA version 1.00.82.394270 as our in-memory database system in landscape mode to form a distributed database [2]. The required database software is provided via a dedicated remote services directory. Thus, after mounting the services directory, we need to perform the installation of the local database instances. For minimizing the efforts of installing all database instances in a larger cluster, e.g. more than 25 computing nodes, we incorporate the parameter-based installation. This means that we predefined all parameters for the installation and provided them as

command parameters, which we then executed in parallel across all nodes at the same time using the Linux tool Parallel Distributed SHell (PDSH) version 2.29 [5]. As a result, the required binaries were copied to the local database nodes, the local instances started, and registered online with the SAP HANA master server, which is located within the Future SOC network, without any configuration downtime of the overall system. In the concrete use case, we incorporated the SAP HANA Database Lifecycle Manager with the `addhosts` command [7]:

```
./hdblcm
  --action=add_hosts
  --addhosts=node-01,...,node-25
  --root_user=lmroot
  --listen_interface=global
```

3. Next Steps

We faced various network configuration challenges in our attempt to connect both systems of the Future SOC Lab and the research facility causing delays in the overall system deployment. Therefore, we were not able to finish with the overall setup and testing by the end of the current Future SOC Lab period. In the following, we outline the steps we will further undertake to set up the overall system if Future SOC Lab resources are granted for the upcoming period, which are mainly subscribing to and configuring of selected services.

3.1 Subscribe to Managed Service

In the given application scenario, we provide the managed service for processing and analysis of genome data as a web application, which is accessible via any Internet browser. We host and manage the web applications at our site as the service provider whilst users of the research site using the URL of the application can access them. Currently, we support either local user accounts or the integration of existing authentication providers, e.g. using OAuth 2.0, for authentication [1].

Customer The application administrator of the research facility subscribes to the managed services for the entire facility or research department. The service provider grant access to administer the application and settings. The application administrator is responsible to maintain user groups and access rights for users of the research site within the application. The application administrator maintains the mapping of application users to corresponding database users and roles and the service provider maintain users and roles in the database.

Service Provider The service provider defines a dedicated database schema per research facility. A database schema is a container for a set of database tables, functions, and stored procedures. The service provider keeps each database schema isolated, i.e. tenant-specific data is separated to ensure data privacy [8]. The database administrator maintains specific user roles per tenant and grant them access to their tenant-specific database schemes. Each database schema is partitioned across a tenant-specific resource set, i.e. a local subset of the overall computing nodes, which is used for storing and processing the data. The database administrator can update the list of computing nodes online without interfering running operations, i.e. data is repartitioned without any database downtime. Furthermore, the database landscape administrator can assign additional resources to a resource set, e.g. to ensure scalability by adding resources of the service provider.

3.2 Configure Selected Service

The user of the research site accesses the managed service using the URL of the web application. The web application is accessed via the VPN connection, i.e. all data is exchanged via the secured tunnel. The end user is able to maintain her/his personal profile and configure application settings. In the application scenario, each end user was able to define her/his local home directory, which contains all genome data they were working on.

4. Conclusion

In this project report, we shared our current status and experience gained in setting up a hybrid cloud computing environment enabling a) the use of cloud service even if legal requirements do not allow exchange of sensitive data with traditional cloud apps and b) the processing of huge data sets locally when their exchange would significantly delay the processing of data even with latest network bandwidths. We described all steps that have been undertaken so far to interconnect local computing resources of research facilities in a star schema using secured VPN connections via the Internet. We also outlined what other steps are required before we can test our computing environment. As a result, we are be able to provide research apps in a Software-as-a-Service (SaaS) provision approach without moving high throughput NGS data to remote computing resources. Instead, we are be able to provide managed algorithms that are executed on the local research site computing resources where data resides. The results of

the data processing are stored in local database systems, which are then configured to form a single distributed database instance granting access only to personal results. We are convinced that sharing knowledge is the foundation to support research cooperation and to discover new insights cooperatively.

References

[1] D. Hardt. RFC6749: The OAuth 2.0 Authorization Framework. `http://tools.ietf.org/html/rfc6749/`, 2012. last accessed: Mar 27, 2015.

[2] F. Färber, S. K. Cha, J. Primsch, C. Bornhövd, S. Sigg, and W. Lehner. SAP HANA database: data management for modern business applications. *ACM Sigmod Record*, 40(4):45–51, 2012.

[3] S. S. Langmead B. Fast Gapped Read Alignment with Bowtie 2. *Nature Methods*, 9(357–359), 2012.

[4] H. Li and R. Durbin. Fast and Accurate Short Read Alignment with Burrows-Wheeler Transformation. *Bioinformatics*, 25:1754–1760, 2009.

[5] M.A. Grondona. Parallel Distributed Shell (PDSH). `https://code.google.com/p/pdsh/wiki/UsingPDSH`, 2012. last accessed: Mar 27, 2015.

[6] H. Plattner and M.-P. Schapranow, editors. *High-Performance In-Memory Genome Data Analysis: How In-Memory Database Technology Accelerates Personalized Medicine*. Springer-Verlag, 2014.

[7] SAP SE. Add Hosts Using the Command-Line Interface. `http://help.sap.com/saphelp_hanaplatform/helpdata/en/0d/9fe701e2214e98ad4f8721f6558c34/content.htm`, 2012. last accessed: Mar 27, 2015.

[8] J. Schaffner. *Multi Tenancy for Cloud-Based In-Memory Column Databases*. Springer, 2013.

[9] M.-P. Schapranow, F. Häger, C. Fähnrich, E. Ziegler, and H. Plattner. In-Memory Computing Enabling Real-time Genome Data Analysis. *International Journal on Advances in Life Sciences*, 6(1-2), 2014.

[10] M.-P. Schapranow, F. Häger, and H. Plattner. High-Performance In-Memory Genome Project: A Platform for Integrated Real-Time Genome Data Analysis. In *Proceedings of the 2nd Int'l Conf on Global Health Chall*, pages 5–10. IARIA, Nov 2013.

Inspection and Evaluation of Modern Hardware Architectures

Frank Feinbube, Felix Eberhardt, Wieland Hagen, Max Plauth, Lena Herscheid
and Andreas Polze
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam
Germany
{frank.feinbube, felix.eberhardt}@hpi.de

Abstract

Modern hardware architectures introduce complex memory hierarchies and a variety of processing unit designs. To allow a beneficial distribution of data and processing, successful algorithms and competitive software systems need to be enabled to deduce the composition and the capabilities of the underlying hardware. Examples of such modern hardware architectures are NUMA-based computer systems, GPU-based accelerators, Intel's Xeon Phi, and hybrid architectures that have multiple such characteristics.

The purpose of this study is to evaluate means to inspect hardware architectures and develop tools to monitor the execution of algorithms in order to find potential performance opportunities. Based on these findings we plan to optimize important algorithms to better utilize the resources at hand.

1 Introduction

In the course of our studies in FutureSOC Lab of the Hasso Plattner Institute, we looked into the capabilities of modern hardware architecture to improve the performance of algorithms with well-known application bottlenecks. Due to the novelty in their on-chip networks, interconnection characteristics and memory hierarchy modern NUMA systems were of special interest for our studies. In order to evaluate their capabilities for improved scalability, execution performance and programmability we studied the optimization of well-known algorithms from the Berkeley Dwarfs (1) (2) collection. This report summarizes our work and findings.

2 Survey on NUMA tools

NUMA tools provide means to achieve one of the following goals: acquire topology information or identify application bottlenecks. Acquiring the topology information is crucial for an efficient data partitioning and task mapping onto the underlying hardware. To further optimize the resulting performance is then essential to be able to identify and study the application bottlenecks.

In our survey we evaluated and categorized the following NUMA tools:

- ACPI
- Linux sysfs
- Numactl –hardware
- Intel Performance Counter Monitor
- Hwloc Istopo
- Linux Perf
- Numatop
- MemAxes
- VTune
- Mlc -e

Furthermore we looked into their capabilities to supported performance engineers in discovering and studying the topology, as well as, gaining a better understanding of the performance-critical execution characteristics of their algorithms on the target hardware architecture. The full survey is available in (3).

3 Related Work

After we looked into the body of NUMA tools (3) (4) (5) (6), we widened our view by studying the current trends in NUMA Hardware (7) (8) (9) and their impact on Operating Systems developments (10) und support (11). Furthermore we looked into popular parallel programming models and reviewed their capabilities to support NUMA-oriented parallelizations (12) (13) (14) (15), as well as, state-of-the-art for thread and data placement (16) and memory consistency models (17).

After this elaborate survey of the research field, we started to work on a variety of case studies to apply the state-of-the-art, identify research gaps and propose new optimization schemes for important application bottlenecks (1) (2).

4 Case Study #1: Error Detection Code Graph Search

The EDC Graph Search algorithm was provided by the Research group for Fault-tolerant Computing by Prof. Michael Gössel. It is the foundation of their current research in error detection codes, since it allows the derivation of such codes from error graphs modelling arbitrary errors. A detailed description of the algorithm can be found in (18). The graph of all possible error codes is iteratively refined. In each iteration the vertex with the highest rank is removed.

EDC Graph Search is an NP-complete problem representing Berkeley Dwarf number 2: Graph Traversal. It is a very compute-intensive algorithm with an exponentially growing solution space. Based on the sequential version used in the paper, we investigated its potential for parallelization and performance optimization to speed up the computation and to allow the computation of larger problem sizes.

Graph algorithms are a very interesting case for NUMA environments: Their inherent dependencies lead to very bad performance results in fully distributed execution environments, while their vast resource requirements only allow the computation of very small problems in the small-sized memories of UMA environments. For NUMA systems on the other hand, the strong notion of locality that can usually be found in the graphs' dependencies seems to promise significant performance potential if they can be mapped to the interconnect memories accordingly.

As a first step, we used the *Intel VTune profiler* to identify where most of the runtime is spent. We identified the bottleneck of the EDC algorithm to be a maximum search over the graph, a typical reduction operation. We used OpenMP to parallelize the *for loop* at the heart of the function: each thread operates on a subset of the vertices, first determining a local maximum and eventually merging its result with the other threads.

Figure 1, Figure 2 and Figure 3 show the observed execution times for one to 240 threads on our hierarchical NUMA test system. For the 15 cores of a single processor the achieved speedup is nearly linear, i.e. very close to the optimum. This demonstrates, that we indeed found the bottleneck and that the parallel part – obtaining thread local maxima – is significantly larger than the accumulation of the results at the end of our parallel region. The straight forward optimization that we applied is well-suited for UMA situations, where the graph completely resides in the local memory of the assigned processor.

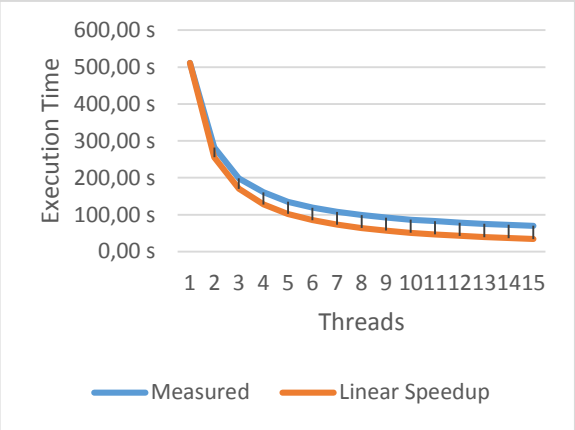


Figure 1: Single-Processor execution times. On a single 15-core processor, we achieve close-to optimal speed up.

When we also utilize the other three processors on the same blade (Figure 2) we see the first NUMA effects bending our speedup curve. This degradation in relative performance is typical for parallel algorithms that are highly-optimized for UMA systems. Without taking the NUMA characteristics into account and applying further optimizations accordingly, it is still possible to get faster with more resources, but the speedup becomes increasingly insignificant.

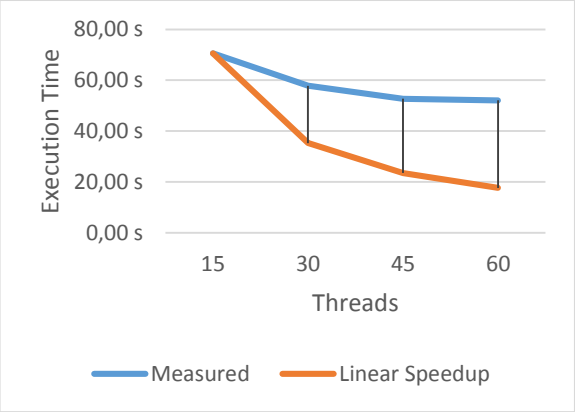


Figure 2: Multi-Processor execution times. On a single 4-processor blade, we see speedup degradations with each additional processor due to the NUMA overhead.

When we start using the processors on the other three blades as well (Figure 3), the NUMA overhead becomes so strong, that the additional resources increase the overall execution time. The bandwidth and latency limitations are so significant, that the access times to the graph data cannot be mitigated by the increased processing performance of the additional processors.

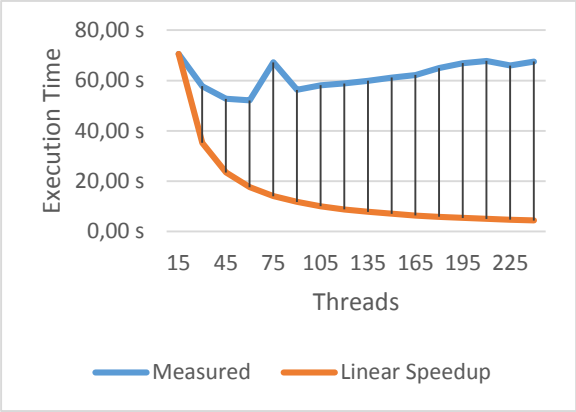


Figure 3: Multi-Blade execution times. The NUMA overhead between blades is so huge, that the application of more than one blade literally decreases the execution performance.

This case study demonstrates, that in order to benefit from the additional resources provided by modern business-class NUMA systems, algorithms have to be refined and novel optimizations techniques have to be applied accordingly.

Further details on this case study are provided in (3).

5 Case Study #2: Matrix-Matrix Multiplication

The multiplication of matrices is a computational pattern which plays a pivotal role in many big data analyses, since it is fundamental for statistics algorithms, finance, scientific computing, signal processing, imaging, and many more. It is the most prominent representative of Berkeley Dwarf number 1: Dense Linear Algebra. An implementation of matrix multiplication is part of any high performance computing library, such as Intel Math Kernel Library (MKL). We studied the parallel performance of matrix multiplication using several NUMA optimizations.

A naïve implementation of a Matrix-Matrix Multiplication comes with a complexity of $O(n^3)$. It can be parallelized in a straight-forward fashion by distributing the input data to all computation nodes and pinning threads accordingly so that they only work on locally available memory. This is very efficient for NUMA systems as well, because it minimizes the amount of remote memory accesses. Using a naïve Matrix-Matrix-Multiplication parallelized by rows, we achieved a speed up of 45 on an 8-node NUMA system with 128 cores. (See Figure 4) Furthermore, the impact of distributing threads and memory accordingly varies from speed up factors of 1.5 to 2.0. (See Figure 5)

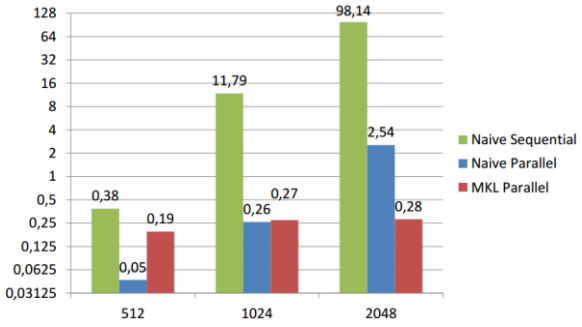


Figure 4: Execution time of Matrix-Matrix-Multiplications on an 8-node NUMA system with 128 cores for matrix sizes of 512, 1024 and 2048.

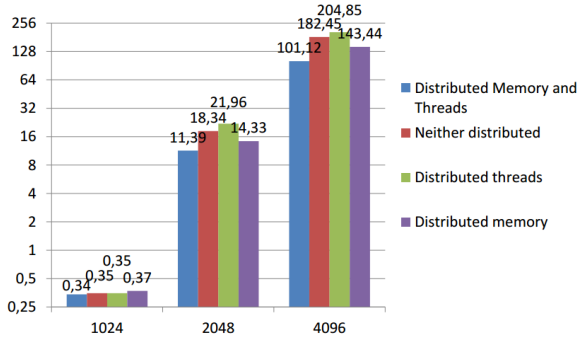


Figure 5: Execution time of thread and memory placements on an 8-node NUMA system with 128 cores for matrix sizes of 1024, 2048 and 4096.

A more sophisticated matrix multiplication algorithm is the *Strassens algorithm*. Instead of $O(n^3)$, as with the naïve approach, its asymptotic complexity is only about $O(n^{2.8})$. The key idea is the recursive definition of temporary submatrices. The result can be expressed in a certain way using these submatrices, thus saving one expensive multiplication operation.

After implementing a parallel version of Strassens algorithm, we applied further optimizations, such as *tiling*. This is the approach of dividing the data into tiles, whose size depends on the cache size. Within a tile, a partial result can be computed while decreasing the likelihood of cache misses. Since the multiplication always affects one row of the left hand side matrix, and one column of the right hand side matrix, one of the rows will not be cache aligned. Therefore, a further optimization step is to transpose one matrix before multiplying. As shown in Figure 6, the amount of cache misses can be reduced and the runtime shortened significantly.

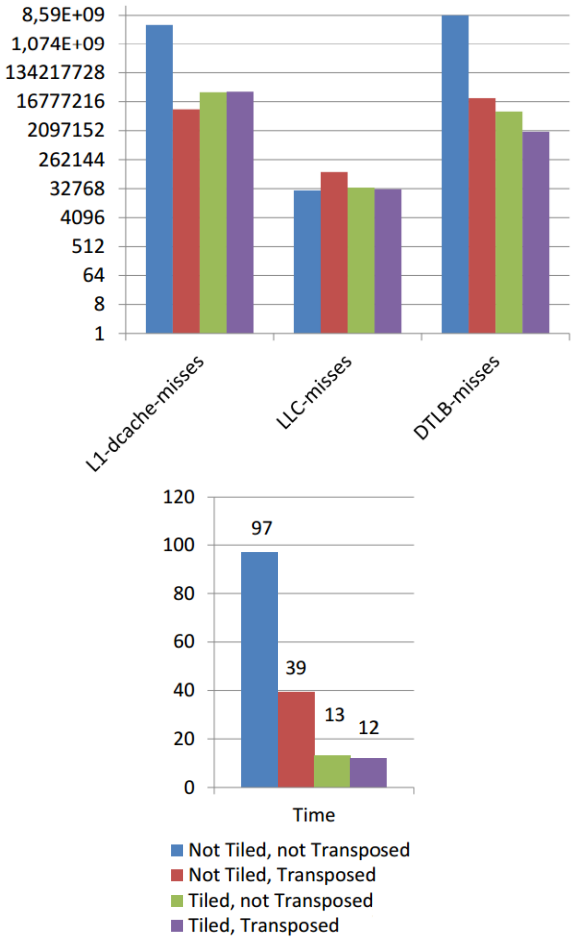


Figure 6: Cache misses and execution time for tiling and transformation with a matrix size of 2048.

Next, we leveraged SSE instructions to gain further performance improvements. Provided, the data is aligned in 128 bit pieces, these instructions can operate efficiently on multiple floating point numbers at once. The allocation of 16 bit aligned memory for SSE purposes was implemented as depicted below. We observed that using SSE instructions pays off, especially for small matrices. A comparison of runtimes for small matrices (Figure 7) and large matrices (Figure 8) is depicted below.

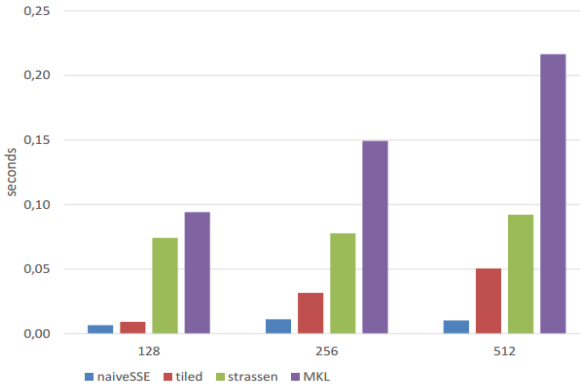


Figure 7: Execution time of naive, SSE-based, Strassen, and MKL matrix multiplications for small matrices.

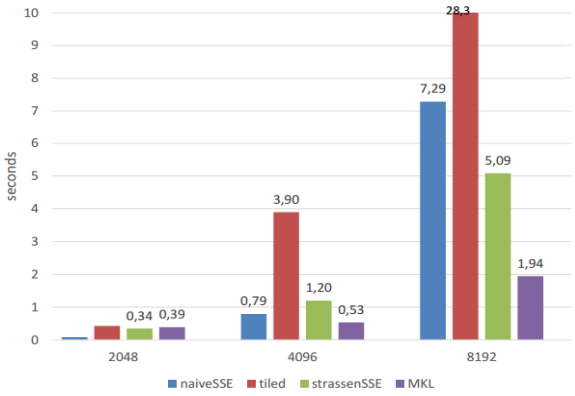


Figure 8: Execution time of naive, SSE-based, Strassen, and MKL matrix multiplications for larger matrices.

Even for basic fundamental algorithms such as matrix multiplication, realizing efficient implementations for NUMA systems are challenging. At a high level, the thread and data placement needs to be considered carefully, but this does not imply that low level optimizations like using SSE instructions are no longer necessary to draw on the full potential of the hardware.

Further details on this case study are provided in (19) and (3).

6 Case Study #3: Reader/Writer-Locks

One of the fundamental challenges to ensure computational correctness in parallel systems is the prevention of race conditions by synchronization mechanisms. Wrongly applied mutual exclusion mechanisms can lead to deadlocks, live locks, starvation and fairness problems. These mechanisms, as well as, their lock-free counterparts also have very severe performance implications due to the sequentialization, protocol overhead and the interaction with coherency protocols. The performance costs for cache-coherence and the management for the data structures needed for synchronization implementations are significantly higher in NUMA systems, a cost all algorithms that cannot avoid the use of shared resources or critical sections will have to pay.

This underlines the fact that with hierarchical NUMA-systems, not only the placement of memory and threads is a crucial, but it is likewise important to consider cache and resource issues related to synchronization primitives. A solid NUMA-aware implementation of mechanisms would be very advantageous for a vast amount of algorithms and applications.

In this case study, we looked into Cohort Locks which are the state-of-the-art in NUMA-aware Reader-Writer-Locks. They are specifically tailored to reduce the overhead of locking mechanisms for NUMA systems. Cohort Locks differ from basic Reader-Writer-Locks by two introducing the notion of cohorts and by support thread obliviousness. Cohorts are a set of

threads interested in accessing the same resource. Cohort detection allows a thread to determine whether or not there are additional thread waiting to acquire a desired lock. Thread obliviousness makes it possible that a lock can be acquired by one thread and released by another. The concept of Cohort Locking is very interesting because it allows to compose NUMA-aware locking primitives from NUMA-oblivious ones. The cohort concept allows to leverage locality and reduce the overhead for cache coherence protocols. The impact on caches is demonstrated in Figure 9.

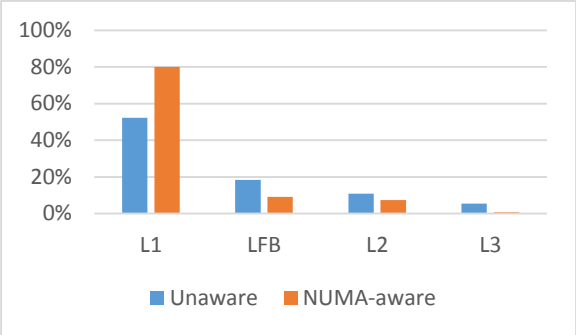


Figure 9: Percentage of cache hits for Reader/Writer Lock benchmarks.

In contrast to the NUMA-unaware pthreads-based implementation, most of the cache hits for the NUMA-aware implementation occur in the L1 cache, which results in reduced execution times.

Further details on this case study are provided in (20) and (3).

7 Case Study #4: Speeded Up Robust Features (SURF)

SURF is a descriptor-based object detection algorithm. Similar to other object detection algorithms SURF is realized by a number of distinct computation steps, the compute heaviest being a convolution.

Convolution algorithms are a popular representative of Berkeley Dwarf number 5: Structured Grid. These kinds of computational patterns are often found in computer vision, signal processing and simulations of all kinds.

Data is organized in a regular multidimensional grid usually realized as an array data type. Accesses to the grid are regular, statically determinable and therefore predictable. Computations are realized as a sequence of logically concurrent grid updates where the new value for a cell is determined using values from its neighborhood only. It is usually implemented as a sequential sweep through the computation domain, either using an in-place update algorithm or two grids: a read-only 'old' version of the grid and a write-only 'new' version.

The characteristics of this algorithm allow for very efficient parallelizations on both, UMA and Distributed systems. It is usually highly vectorizable and provides a vast amount of fully independent operations. If implemented accordingly, both spatial localities along cache lines, as well as, temporal localities allowing for cache reuse can be exploited. Furthermore the partitioning of the grid into sub-grids allows a distributed computation, requiring communication only for the synchronization of boundary cells.

While existing parallelization strategies already provide exceptional performance results, the inherent affinity of the algorithm to produce localities makes it the perfect fit for hierarchical NUMA systems. (See Figure 10) Since cells behind the boarder need to be accessed, communication overhead is introduced on Distributed systems. These cells are accessed in a read-only fashion, having a decreasing access frequency behind the border.

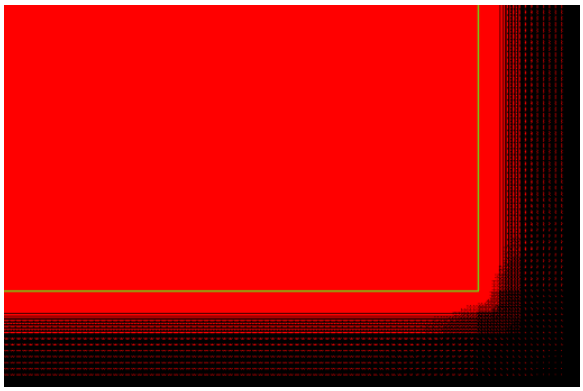


Figure 10: Detailed view of the memory access pattern for SURF at the border of the 2D grid.

On NUMA systems the efficient computation of grids way larger and far more complex than what a UMA system could support can be afforded, without the price of big latencies and small bandwidth usually found in Distributed systems. The interesting research objective is the reliable determination of the "sweet spot" of copying data versus accessing it remotely.

Further details on this case study are provided in (21) and (3).

8 Conclusion and Future Work

We see a vast amount of innovative applications seizing the opportunities of real-time analytics on Big Data which is made possible by the creation of novel business-class hierarchical NUMA systems. We feel that these systems promise significant performance improvements for the identified crucial application bottlenecks. What really sets them apart from powerful cluster systems, though, is the fact that they also provide a close-to-ideal environment for composite algorithms. In contrast to the specific, pure algorithms usually found in benchmarks, real applications are a complex choreography of interwoven algorithms, each having distinct requirements for efficient computation and communication. By supporting the scalability of Distributed systems while by providing a UMA-like environment of coherent memory with reasonable

latencies and bandwidth characteristics, hierarchical NUMA systems, allow to realize all types of communication and computation patterns efficiently.

The hard question is, though, how to express an application in a way that it can be mapped and executed on a NUMA system in the best possible way. To date, best practices and optimization techniques focus either on parallel shared memory systems (UMA, e.g. with OpenMP) or distributed message-passing systems (e.g. with MPI); pure NUMA optimizations have been mostly neglected because the performance penalties were moderate and there is no intuitive programming metaphor allowing for portable performance.

The emerging class of hierarchical cache-coherent NUMA systems requires:

- Novel portable optimization techniques and best practices
- NUMA-aware tools, libraries, programming models, patterns, distribution schemes, …
- The adequate consideration of topology and hardware characteristics.

In our future research we are focusing on contributing to these efforts, in order to pave the way for complex and challenging applications of the future.

9 Acknowledgments

This work was only possible due to the generous offer to conduct our studies on the modern hardware within the HPI FutureSOC Lab. Hereby we want to show our gratitude to the FutureSOC Lab steering committee which accepted our project proposal and the FutureSOC Lab team for the daily cooperation. Special thanks go to Bernhard Rabe for friendly and timely support, advice and help.

References

1. **Patterson, David, et al.** *The Landscape of Parallel Computing Research: A View from Berkeley.* Berkeley : University of California, 2006. UCB/EECS-2006-183.

2. —. Dwarf Mine. *The Landscape of Parallel Computing Research: A View From Berkeley.* [Online] 2015. http://view.eecs.berkeley.edu/wiki/Dwarfs.

3. **Eberhardt, Felix, et al.** *Technical Report NUMA4HANA #2.* Potsdam : Hasso Plattner Institute, 2015.

4. *Topology Detection.* **Knebel, Sven.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

5. *Scientific approaches: NUMA Profilers/analyzing runtime behavior.* **Swart, Malte.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

6. *Performance Counter.* **Tausche, Karsten.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

7. *Multiprocessor architectures.* **Heidler, Kirstin.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

8. *Cache Coherence in NUMA Systems* . **Frohnhofen, Johannes.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

9. *Interconnection Technologies.* **Zarisheva, Elina.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

10. *Linux NUMA evolution.* **Teschke, Fredrik and Pirl, Lukas.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

11. *NUMA Kernel APIs.* **Korsch, Dimitri.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

12. *NUMA in High-Level Languages.* **Siegler, Patrick.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

13. *NUMA and Open-CL.* **Sachse, Jan Philipp.** Potsdam : NUMA Seminar, 2015. Hasso Plattner Institute.

14. *NUMA with OpenMP.* **Springer, Matthias.** Potsdam : NUMA Seminar, 2015. Hasso Plattner Institute.

15. *NUMA with OpenMPI.* **Fiedler, Carolin.** Potsdam : NUMA Seminar, 2015. Hasso Plattner Institute.

16. *Scientific approaches to Thread and Data Placement.* **Eckert, Fabian.** Potsdam : NUMA Seminar, 2014. Hasso Plattner Institute.

17. *C++ 11 Memory Consistency Model.* **Gerstenberg, Sebastian.** Potsdam : NUMA Seminar, 2015. Hasso Plattner Institute.

18. *Error Detection Codes for Arbitrary Errors Modeled by Error Graphs.* **Nieß, Günther, Kern, Thomas and Gössel, Michael.** Porto : GI/ITG/GMA Technical Committee "Dependability and Fault Tolerance" (VERFE) and DFG Priority Program SPP 1500 "Dependable Embedded Systems", 2015. 11th Workshop on Dependability and Fault Tolerance (VERFE'15).

19. *NUMA-aware Matrix-Matrix-Multiplication.* **Reimann, Max and Otto, Philipp.** Potsdam : NUMA Seminar, 2015. Hasso Plattner Institute.

20. *NUMA-aware Reader-Writer Locks.* **Herold, Tom and Lamina, Marco.** Potsdam : NUMA Seminar, 2015. Hasso Plattner Institute.

21. *NUMA-aware SURF.* **Sterz, Christoph and Schmidt, Patrick.** Potsdam : NUMA Seminar, 2015. Hasso Plattner Institute.

Project Report on "Large-Scale Graph-Databases based on Graph Transformations and Multi-Core Architectures"[1]

Hasso Plattner Institute

Matthias Barkowsky
Matthias.Barkowsky@student.hpi.uni-potsdam.de

Henriette Dinger
Henriette.Dinger@student.hpi.uni-potsdam.de

Lukas Faber
Lukas.Faber@student.hpi.uni-potsdam.de

Felix Montenegro
Felix.Montenegro-Retana@student.hpi.uni-potsdam.de

Abstract

Updating large scale graph data interactively accessed by multiple users applying graph transformations requires a high throughput. This can only be achieved by concurrently executing multiple queries. In order to obtain consistency of the graph data, different synchronisation strategies for graph transformations were prototypical implemented and evaluated using the Future Soc Lab. This project report presents the project's basics and results of the test runs.

1. Introduction

Large scale graph databases keep gaining influence on the web, for example in social networks. Accordingly, the amount of requested updates on the graph data becomes hard to handle with requests potentially incoming faster than they can be processed. To obtain the possibility of interactively querying [5] and of a multi-user mode the data the throughput of queries needs to be increased. This can be achieved using parallelism on multicore architectures. Then, the problem of ensuring consistency in the data needs to be addressed. Therefore different synchronisation strategies of synchronising access to graph data were developed and implemented in the project. In our project updates are performed as graph transformations [4], which consist of a left side graph, which is matched onto the graph data and replaced by a right side graph. There are transformation engines like Henshin [1] or Story Diagram Interpreter [2] capable of performing such graph transformations.
Their work consists of the search for matches and replacing them. We distinguish engines in one-ary and two-ary engines. Two-ary engines like Henshin are able to execute a query in two steps: First, the matches are found and in a second step they are replaced. In contrast one-ary engines like Story Diagram Interpreter lack this option only offering an execution in one command.

2. Implementation and Synchronisation

The framework offers adapters for different transformation engines which are operating on graph data generated by the Eclipse Modelling Framework [3] [3]. To assure a consistent state during parallel executions, accesses to the databases must be synchronised. In this project, multiple synchronisation strategies are implemented and evaluated.

2.1. Global Lock Synchronisation Strategy

The Global Lock Synchronisation Strategy allows an execution after acquiring a lock for the entire graph database, so no two queries will be executed simultaneously. This strategy is used as a reference to evaluate the other strategies.

[1] http://www.eclipse.org/henshin/

[2] http://www.hpi.uni-potsdam.de/giese/public/mdelab/mdelab-projects/story-diagram-tools/

[3] http://www.eclipse.org/modeling/emf/

2.2. Improved Global Lock Synchronisation Strategy

The Improved Global Lock Synchronisation Strategy additionally uses read-write-locks. Thus, any number of reading queries can be executed in parallel. Since reading queries do not have side effects, the concurrent execution of an arbitrary number of reading queries cannot harm the consistency of the database. Since data can be stored in different graphs the strategy can further be improved by providing one lock for each graph, allowing parallel executions on different graphs [2].

2.3. Version Synchronisation Strategy

A Version Synchronisation Strategy lets changes be written on a copy of the graph data, a so called version. Thus all queries can be executed simultaneously. As changes of unfinished queries are performed in versions, the original graph data is always in a consistent state. Therefore queries that only need reading access can work on the original graph. When a writing query is finished the version and the original graph have to be merged. The result is a new original graph on which new reading queries operate. Merging versions is the most difficult part of this strategy since it has to maintain the database in a consistent state and at the same time ensure that all changes were saved [2].

3. Work on the Future SOC Lab

3.1. Used Future SOC Lab Resources

All experiments were executed on a Fujitsu RX600S5 - 2 architecture with 4 Xeon X7550 processors, each having eight cores with 2GHz, providing 1024 GB of RAM and running Java7 and Ubuntu 14.04.

3.2. Results

In our experiments, we executed two different benchmarks consisting of 1.000 queries each on a graph with 500.000 nodes. We used six types of queries:

- Reading queries (R): A reading query queries for one node and all adjacent nodes.
- Complex Reading queries (CR): A complex reading query reads approximately 10% of all nodes in the graph.
- Modifying queries (U): A writing query modifies an attribute of one node based on the attribute values of its adjacent nodes.
- Complex modifying queries (CU): A complex query modifies an attribute of approximately 10% of all nodes in the graph.
- Creating queries (C): A creating query copies one node, thereby also creating several edges.
- Deleting queries (D): A deleting query deletes one node, thereby also deleting several edges.

From these queries we built two benchmarks. One of these benchmarks named ”Reading” only includes reading queries, the other named ”Mixed” included both reading and writing queries. The distribution of queries can be seen in table 1.

benchmark	R	CR	U	CU	C	D
Reading	95%	5%	0%	0%	0%	0%
Mixed	45%	5%	34%	1%	10%	5%

Table 1. Percentage of query types in each benchmark

The tested strategies include a simple Global Lock Synchronisation Strategy (GL), an improved variation, the Improved Global Lock Synchronisation Strategy (IGL) and a Version Synchronisation Strategy (V).
The Global Lock Synchronisation Strategy only allows the execution of a single query at a time, regardless of whether this query is a reading or a writing query. This means that using the Global Lock Synchronisation Strategy, all queries are executed sequentially.
The Improved Global Lock Synchronisation Strategy uses a Read/Write-Lock to enable concurrent execution of reading queries. It also allows writing queries that are executed with a two-ary engine to search for matches while other reading queries are executed, but blocks concurrent write operations.
The Version Synchronisation Strategy creates a copy of the graph that writing queries can work on without interfering with reading queries which means that reading and writing queries can be executed simultaneously, but also only allows one writing query at a time to avoid conflicting versions. Because writing queries modify only the copy a parallel execution of reading queries is possible.
Figure 1 shows the throughput achieved by the different strategies. It can be seen that the Improved Global Lock Synchronisation Strategy and the Version Synchronisation Strategy have a very high throughput when there are only reading queries since they are allowed to work at the same time. The throughput significantly drops when there are writing queries.
The Improved Global Lock Synchronisation Strategy suspends reading queries while a writing query is being executed. The Version Synchronisation Strategy creates one copy for writing queries, which is periodically merged. However, writing queries can only sequentially modify that copy. Additionally, the overhead of copying the whole graph on for each ten writing queries further reduces the throughput of the Version Synchronisation Strategy compared to the Im-

proved Global Lock Synchronisation Strategy. Unlike the Improved Global Lock Synchronisation Strategy, writing queries modify that copy so that reading queries can concurrently operate on the original graph. The Global Lock Synchronisation Strategy has a very low throughput overall, because only one query is allowed to work at the same time. This also explains that the throughput for this strategy is nearly the same for the reading and mixed benchmark.

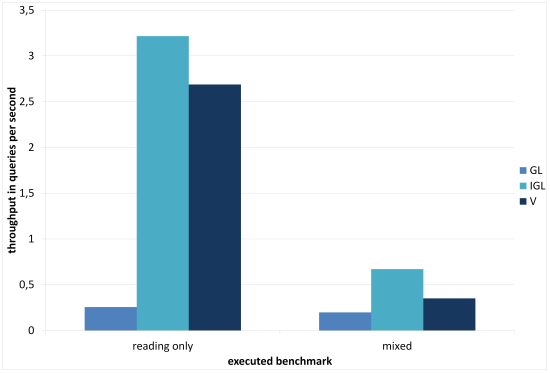


Figure 1. Average number of benchmark queries executed per second using different synchronisation strategies

Figure 2 shows how much time the queries needed depending on the strategy. As indicated, the Improved Global Lock Sychronisation Strategy and the Version Synchronisation Strategy need almost the same time to execute the reading query benchmark. Executing the writing benchmark the Version Synchronisation Strategy takes significantly longer. Similarly, the Global Lock Sychronisation Strategy shows a comparably low performance in both benchmarks. Striking, however, is that the curves of the improved global lock and the Version Synchronisation Strategy both have a sharp bend, which occurs when the reading queries, which are executed faster, are finished. As the Improved Global lock Synchronisation Strategy allows simultaneous execution only for reading queries, they will be preferred. Writing queries are then delayed until all reading queries are finished. This could become a problem when multiple users access the database, since it cannot be assured that a writing query will start at any time.

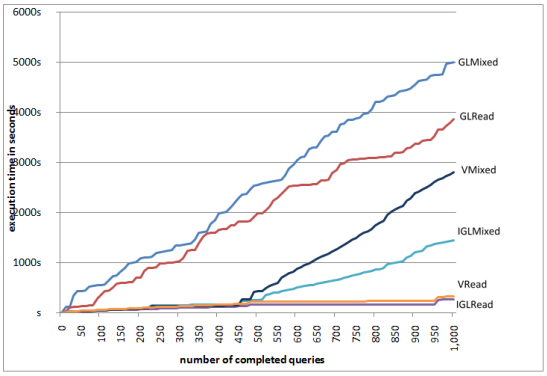


Figure 2. Execution progress of two benchmarks using different synchronisation strategies

3.3. Further Steps

Further steps may include improving the strategies to allow higher parallelism when executing writing queries and allowing the user to parametrise the strategies.

Another strategy, the Node Lock Synchronisation Strategy, has been prototypical implemented and will be advanced to support the complex queries of the test framework. The Node Lock Synchronisation Strategy ensures that no two parallel executed transformations access the same node of the graph data by providing a lock for each node. In a variation of this strategy a conflict graph is built in which each node represents a query. If two queries want to access the same node, an edge appears between their nodes. Thus, it is visible which queries cannot be executed simultaneously. A scheduler decides in which order conflicting queries will be executed. The user has the possibility to affect the execution order by a parametrisation of the scheduler, influencing for example the maximum or average waiting time of the queries or prioritise them. The danger of deadlocks that this strategy brings when using one-ary engines will be handled by documenting changes. Thus, they can be undone when a deadlock occurs or saved in a version that will be merged after being finished. [2].

To increase the performance of the Version Synchronisation Strategy, a more efficient algorithm for creating the versions can be developed. In the strategy used in our experiments, versions include the whole graph. Thus, many node copies are created but never used. One approach to minimise the number of unused copies is a copy-on-write strategy that only creates copies of nodes at the moment they are changed.

Also, this variation of the Version Synchronisation Strategy only allows a single writing query at a time. The use of an algorithm for merging multiple versions would allow a concurrent execution of multiple writ-

ing queries. However this approach needs to address how to handle conflicting versions for one or multiple nodes.
Merging nodes can be avoided by a combination with the Node Lock Synchronisation Strategy, only allowing a node to be copied to at most one version. This will allow a maximum number of simultaneously writing queries on a copy of the graph data while not blocking any number of reading queries.
Further evaluation will include simultaneous querying of different graphs. Additionally, more data sets and benchmarks will be used to evaluate the applicability of the strategies in different domains and to further identify their advantages and drawbacks.

4. Conclusion

By looking at our test results, it becomes apparent that a highly parallel execution is desirable and greatly increases the throughput of the developed database. To maximize the amount of concurrently executed queries while retaining a consistent state of the stored data, an efficient synchronisation strategy has to be used. We presented different possible solutions on how to maximise concurrent modifications of graph data. We implemented and evaluated two global lock strategies and a version synchronisation strategy. Improving strategies to simultaneously execute both reading and writing queries will allow further parallelism, thereby allowing higher throughput of queries.

References

[1] Large-scale graph-databases based on graph transformations and multi-core architectures. `http://hpi.de/fileadmin/user_upload/hpi/dokumente/studiendokumente/bachelor/bachelorprojekte/2014_15/BA_Projekt_G1_FG_Giese_Framework_Graphendatenbanken.pdf`.

[2] Barkowsky, Matthias, Dinger, Henriette, Faber, Lukas, Montenegro, Felix. Anforderungsdokument. Hasso Plattner Institute, Universität Potsdam, 2014. Bachelorprojekt.

[3] Barkowsky, Matthias, Dinger, Henriette, Faber, Lukas, Montenegro, Felix. Designdokument. Hasso Plattner Institute, Universität Potsdam, 2014. Bachelorprojekt.

[4] Habel, Annegret, Pennemann, Karl-Heinz . Correctness of high-level transformation systems relative to nested conditions. *Mathematical Structures in Computer Science*, 19, 2009. Cambridge University Press.

[5] Miller, Robert B. Response time in man-computer conversational transactions. In *Proc. AFIPS Fall Joint Computer Conference*, pages 267–277, 1968.

Modelling wide area networks using SAP HANA in-memory database

Tadeusz Czachórski
Institute of Informatics
Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
tadek@iitis.pl

Monika Nycz
Institute of Informatics
Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
monika.nycz@polsl.pl

Tomasz Nycz
Institute of Informatics
Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
tomasz.nycz@polsl.pl

Abstract

This report presents our preliminary results in the area of numerical modelling of dynamics of flows in wide area computer networks with TCP/IP protocol with the use of SAP HANA in-memory database. The aim of the project is to explore the possibility of transforming the widely-known modelling method, fluid-flow approximation, into database language, thus overcoming the need of development of dedicated solutions and transmission of the results from the application to the database. We have implemented the modelling logic as the SQL procedures and performed mathematical calculations for an exemplary topology. The experiments show that the database engine may be used to perform all model computations, however further studies how to optimize them are still necessary.

1. Introduction

The analysis of transient phenomena, that occur in computer networks, is performed by simulation or analytical models. Simulation models are usually more detailed, therefore, in the case of the large network, e.g. Internet topologies, are time-consuming and require considerable computing power. Analytical models are based on mathematical equations defining the changes, occurring in the network. The use of mathematical models maintains sufficient accuracy of results and makes the analysis less time-consuming. Transient queue analysis is needed to model time-dependent flows and the dynamics of changes of router queues in modern computer networks. It helps to predict packet loss probability and queueing delays, which are the major factors of the Internet quality of service. The most popular methods of modelling TCP flows are: Markov chains, diffusion approximation and fluid-flow approximation. Each approach has its advantages and drawbacks, but it is the fluid-flow approximation [4, 2], that is widely used for modelling transient states in wide area networks, including the Internet.

The fluid approximation uses first-order ordinary linear differential equations to determine the dynamics of the average length of node queues, eq. (1), and the dynamics of TCP congestion windows, eq. (2), in a modelled network.

The TCP flows traverse in the network from sources to the destinations, through predefined sets of routers. Each router receives K classes and each class consists of M identical flows. Packets within flows are either send forward, when the station is empty, or queued otherwise.

The changes of a queue length at a router v, $dq_v(t)/dt$, eq. (1), are defined as the intensity of the input stream, reduced by the intensity of output flow C_v, i.e. the number of packets sent further in a time unit. The amount of input stream, i.e. the sum of all flows, traversing a particular node, is controlled by the the factor $(1 - p_v(t))$, which implies the rejection of the packet in case of congestion.

$$\frac{dq_v(t)}{dt} = \sum_{i=1}^{K} \frac{W_i(t)}{R_i(\boldsymbol{q}(t))} \cdot M_i \cdot (1 - p_v(t)) + - \mathbf{1}(q_v(t) > 0) \cdot C_v \quad (1)$$

The single flow is characterized by actual window size and RTT time - the time, after which the sender is informed whether the recipient successfully received the packet or not. The window size grows by one at each RTT time $R_i(t)$ in the absence of packet loss and decreases by half after every packet loss, occurring on the path. The amount of loss for the entire TCP connection is modelled as throughput intensity multiplied

by total drop probability on the flow path.

$$\frac{dW_i(t)}{dt} = \frac{1}{R_i(\boldsymbol{q}(t))} - \frac{W_i(t)}{2} \cdot \frac{W_i(t-\tau)}{R_i(\boldsymbol{q}(t-\tau))} \cdot \cdot \left(1 - \prod_{j \in V}(1 - P_{ij})\right) , \quad (2)$$

The RTT time, eq. 3, in the above formulas determines the time needed for the information on congestion and packet loss to propagate through the network, back to the sender of a flow i. It consists of total queue delays at all nodes defined along the connection and the total link propagation delay.

$$R_i(\boldsymbol{q}(t)) = \sum_{j=1}^{K} \frac{q_j(t)}{C_j} + \sum_{j=1}^{K-1} Lp_j \, . \quad (3)$$

The routers have mechanisms preventing overloading their buffers, such as RED, which proactively drops packets when queue increases, with probability $p_v(t)$:

$$p_v(x_v) = \begin{cases} 0, & 0 \leqslant x_v < t_{min_v} \\ \frac{x_v - t_{min_v}}{t_{max_v} - t_{min_v}} p_{max_v}, & t_{min_v} \leqslant x_v \leqslant t_{max_v} \\ 1, & t_{max_v} < x_v \end{cases} , \quad (4)$$

where t_{min}, t_{max} are the thresholds, p_{max} is the probability for t_{max} threshold and $x(t)$ is the weighted average queue length, which is the sum of current queue $q_v(t)$ taken with a weight parameter α and previous average queue with $(1 - \alpha)$.

The fluid-flow differential equations are solved numerically.

2. Project idea

The analysis of the behaviour of computer networks, based on mathematical and numerical approach, implies generating, processing and storing large amounts of output data. The project aims to explore the possibility of transferring the modelling application logic into database engine, thus overcoming the need of development of dedicated solutions and transmission of the results from the application to the database. If we consider a thousand- and million-node topologies, the calculations generate a large amount of data. The standard solution is to create a dedicated software structure for storing and analysing the obtained data or to use the database as a storage. To eliminate the necessity of customizing the resulting structures and waiting times for sending the results to storage, we transferred the application logic into database layer. However, such solution requires a specialized hardware infrastructure. Thus, we focused on SAP HANA as an in-memory database solution.

3. Methodology & Findings

The research consisted of three phases: preprocessing, implementation and modelling, visualization. In the first phase we selected exemplary real network topology, [3] with 134023 nodes and generated 50000 router pairs for flows. Next, we generated the link propagation delays (tens to hundreds of milliseconds), based on which the flows paths were determined using Dijkstra algorithm. Last step concerned the preparation of the initial configuration data as the CSV files, which were imported into SAP HANA database.

The application modelling logic assumed the update of all network parameters at each step and stored them as historical data. The values of particular step were used in subsequent modelling step. The main project phase involved the translating of the numerical fluid-flow approximation logic into the database language. In this case we created tables, loaded them with initial data and test several algorithms based on combination of updates, upserts and inserts. SAP HANA provides two types of data storage: row store and column store. Both were used to determine the most effective way to perform modelling inside the database engine. The logic was processed within the SQL procedure using loop or as a SQL script. Despite the fact that the modelling logic calculated aggregates on both routers and flows, the preliminary results indicated the update-insert on row store as the best solution. However, the project period turned out to be too short to check all necessary optimization possibilities, which could change the best option for the benefit of insert-only approach on column store.

The final phase involved the visualization of calculated results. Therefore, we extracted the values, such as the queue length, for particular steps from SAP HANA database as CSV files. Then we imported the basic topology into Gephi tool (The Open Graph Viz Platform, [1]) and were updating it with files obtained from database. The process ended with visualization, which presented the changes of queues in the whole network.

4. Numerical results

The loads of network nodes (percentage) are defined in our visualization by the colours: 0% (green), 50% (yellow), 100% (red). At the beginning (t = 1 sec), their queues are mainly empty, Fig. 1. With the increase of time the network congestion grows, Fig. 2, reaching a critical point (t = 26 sec), Fig. 2. With further time growth, the increasing congestion activates the mechanisms, that reduce the window sizes of flows, which countermeasure the overflows in nodes and balance the network loads, Fig. 4.

We may also apply the filtering on the particular routers or flows and observe the situation from a detailed point of view. In our example, Figs. 5 and 6

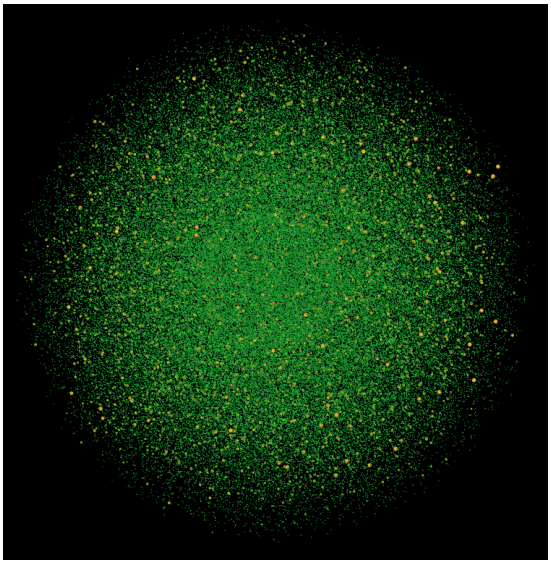


Figure 1. The loads of nodes in the modelled network for time = 1 sec.

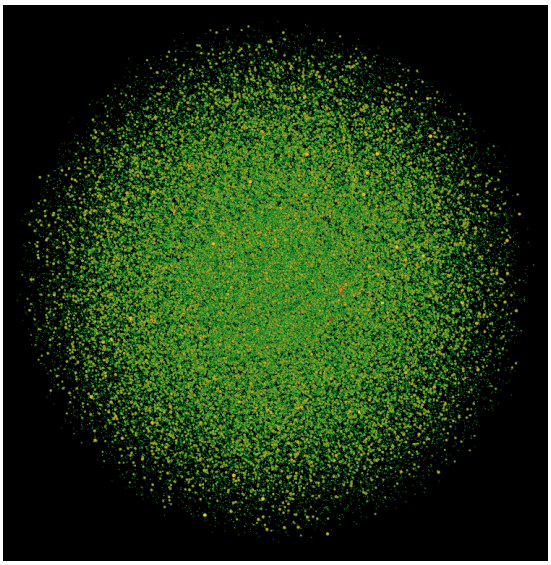


Figure 2. The loads of nodes in the modelled network for time = 13 sec.

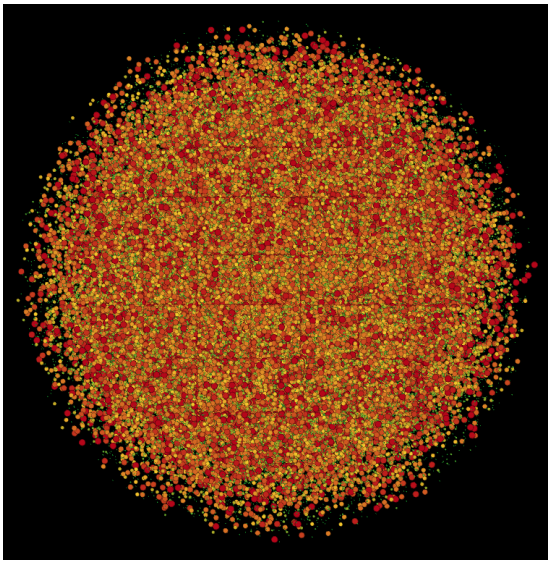


Figure 3. The loads of nodes in the modelled network for time = 26 sec.

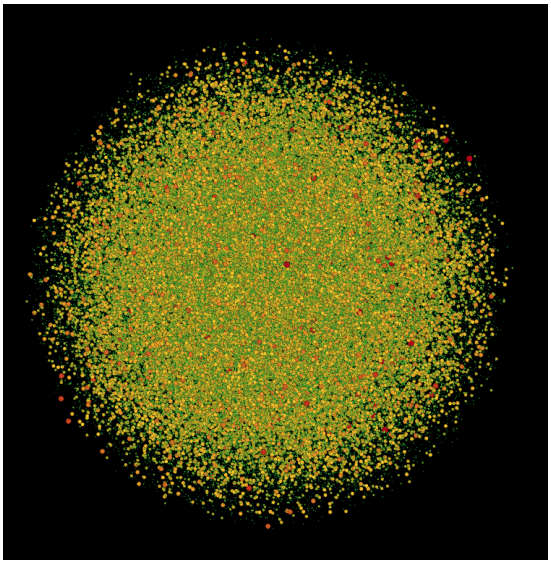


Figure 4. The loads of nodes in the modelled network for time = 50 sec.

show the behaviour of the longest flow (42 nodes) in the network. As it turned out, it was not congested during critical point in time = 26 sec.

5. Future SOC Lab resources

During the project we used the HPI Future SOC Lab HP DL980 G7 server having i. a. 4 x Xeon (Nehalem EX) X7560 and 1 TB RAM. The calculations were performed by running the SQL procedures and scripts within SAP HANA Studio on SAP HANA 1.00.091 version.

6. Conclusions & Next steps

The studies, intending to analyse the final behaviour of the fluid-flow approximation model for large networks have indicated that it is possible to solve first-order differential equations with the use of database engine. Our proposed and examined method concentrated on pure SQL code as an implementation language. We used the code to solve the model for real Internet topology and the characteristics of obtained results, concerning the network congestion changes, were presented with the use of visualization tool.

The time needed for the load of the initial configura-

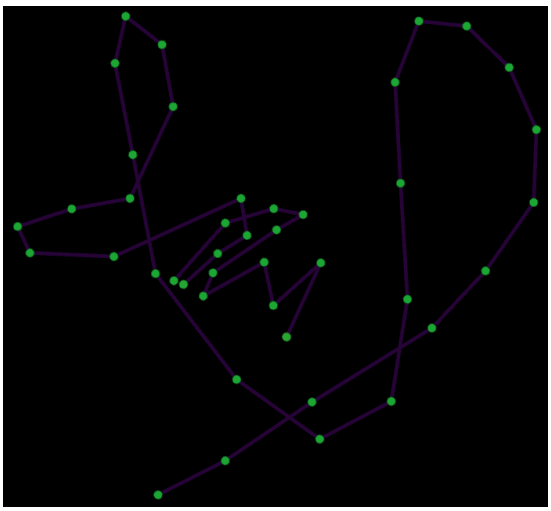


Figure 5. The loads of nodes of the longest flow in the modelled network for time = 1 sec.

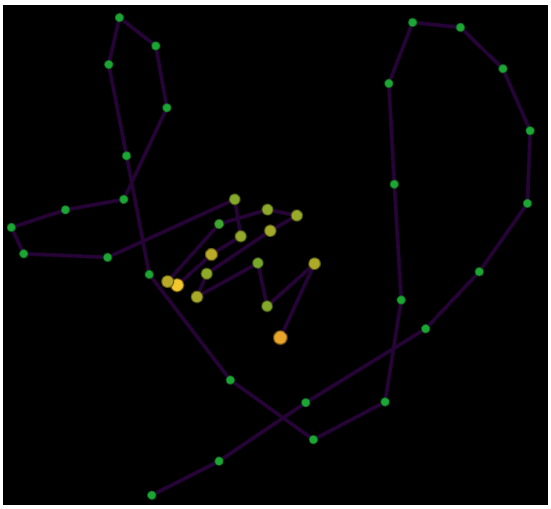


Figure 6. The loads of nodes of the longest flow in the modelled network for time = 26 sec.

tion data is surprisingly small as well as the time for exctrating the results in approprate format acceptable by Gephi (few to several seconds). Thus, once the data are generated, we have wide spectrum of tools and algorithms to mine the knowledge about the network loads. Generally, the main advantages of the our solution are its scalability, portability and universality. However, further studies on this subject are still necessary.

The next step of the research will focus on further optimization of the developed SQL algorithms, based on a set of time performance tests, both on column and row store, with the use of inserts, updates and information models in case of aggregates. We would like to model other topologies also based on CAIDA measurements. Additionally, we plan to compare the performance of our approach with competitive OLTP engines, graph databases solutions and a multiprocessor native implementation.

References

[1] Gephi: The open graph viz platform. `http://gephi.github.io/`.

[2] Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu. Fluid models and solutions for large-scale ip networks. *ACM/SigMetrics*, 2003.

[3] B. Lyon. The Opte Project. `http://www.opte.org/`.

[4] V. Misra, W.-B. Gong, and D. Towsley. A fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. In *Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM 2000)*, pages 151–160, 2000.

Using Process Mining to Identify Fraud in the Purchase-to-Pay Process

- Progress Report -

Galina Baader
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
galina.baader@in.tum.de

Veronika Besner
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
veronika.besner@in.tum.de

Sonja Hecht
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
sonja.hecht@in.tum.de

Michael Schermann
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3 85748 Garching, Germany
Michael.schermann@in.tum.de

Helmut Krcmar
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
krcmar@in.tum.de

Abstract

The aim of our project is to investigate the use of process mining to identify fraud in the purchase-to-pay business process. Therefore, we used the process mining tool Celonis[1] to reconstruct the as-is process from the underlying log data of our ERP system. After loading the respective ERP tables into HANA, we had to transform the relevant data into Celonis readable event log tables. We then displayed the as-is process including 74 deviations from the standard purchase-to-pay processes and analyzed them regarding fraud. Our preliminary results identified indicators for two types of fraud: non-purchase fraud and double issue of an invoice fraud. In the next project period we plan to refine our analysis and use it to identify fraud in real-time within a scientific experiment. For further information, please refer to our project extension application.

1 Introduction

Corporate fraud is a massive problem in most companies, consuming an estimated 5% of the annual revenues of a typical organization [1]. One of the most common and important fraud schemes is asset misappropriation. Since a large number of such cases occur, manual auditing methods are mostly unable to cope with this number. Therefore computer-assisted audit tools and techniques are needed. With these tools and techniques auditors can retrieve and analyze huge amounts of data from enterprise resource planning (ERP) and linked systems [2 ; 3].

Most auditing tools use data mining based fraud detection, but rarely utilize temporal information[2]. One reason might be that running such algorithms on a productive system might have a huge impact on the performance of the system[3]. Process mining can be used to utilize this temporal information. Furthermore, the use of the in-memory database SAP HANA can provide reasonable response times for the analysis of temporal data.

2 Project Goal

The goal of the project is to determine the suitability of process mining to detect fraudulent behavior in the purchase-to-pay business process. We therefore used Celonis process mining, as the tool is able to visualize business processes based on event logs from the underlying system (here SAP ERP). Deviations from the standard processes are visualized and can be investigated regarding fraudulent behavior.

[1] http://www.celonis.de/

The project should result in a proof of concept to show that fraudulent behavior can be efficiently found when using process mining. Furthermore, it should be shown that the use of SAP HANA could overcome performance issues, which limits the use of process mining up to now.

3 Project Design

The basic design of the project and its underlying dataset are presented in the following section.

3.1 Underlying Dataset

The data used in the project was created in a previous project, the White Collar Hacking Contest (WCHC). In the WCHC, seven teams competed against each other by first hiding fraud in the purchase-to-pay business process in an ERP system and after switching datasets trying to detect the frauds of another team. Each team was advised by a professional fraud consultant or auditor, so the datasets include different fraud cases that happen in real companies.

3.2 Project Architecture

Figure 1 shows, how the different tools we used in the project are interconnected. At the client side, the SAP HANA Studio is installed and allows interaction and manipulation of data stored in SAP HANA. We created the needed event log tables for Celonis directly in HANA and gave Celonis access to the corresponding database scheme. Celonis was installed on a web server provided from the HPI.

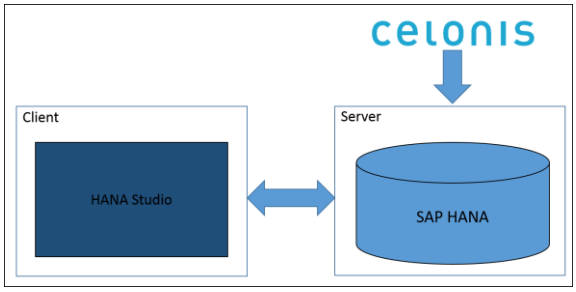


Figure 1: Project Architecture

4 Project Procedure

In the following, the current status of the process mining project in cooperation with Celonis and the HPI will be described.

The project consists of six main steps:

1. Load data into HANA
2. Create activity, case and process tables
3. Create data cubes and data models in Celonis
4. Create analysis
5. Detect process deviations, investigate analysis, filter and explore processes
6. Refine analysis to detect further fraud cases

In the following, these steps will be described in detail.

4.1 Load Data into HANA

The underlying dataset for the examination was taken from the White Collar Hacking Contest. We loaded the respective tables from the ERP system into HANA using SAP BusinessObjects Data Services Designer.

Table 1 provides an overview of the most important tables that we used for the analysis.

Table Name	Table Description
EBAN	Purchase Requisition
EKKO	Purchasing Document Header
EKPO	Purchasing Document Item
EKBE	History per Purchasing Document
BSEG	Accounting Document Segment
BKPF	Accounting Document Header
LFA1	Vendor Master
CDHDR	Change Document Header
CDPOS	Change Document Items

Table 1: Important Purchase-to-Pay Tables [6]

4.2 Create Activity, Case and Process Tables

Once the data tables are imported into HANA, certain event log tables have to be created manually to display process models with Celonis. Therefore, we developed a script (using SQLScript as it is supported by HANA) that creates the respective activity, case and process table.

Activities represent every single step in every order (like purchase requisition, purchase order, invoice received etc.) and cases represent the complete process path of every order position. The third table, the process table, is a utility table and maps activities to integer activity IDs for Celonis' improvements. The activity table is created based on the previously imported tables and afterwards the case and process tables are derived from this activity table.

In the following section, this table creation will be explained.

```
CREATE TABLE CELONIS_P2P_ACTIVITIES(
    ActivityCaseID VARCHAR(18)
    ,Activity VARCHAR(40)
    ,EventTime TIMESTAMP
    ,Sorting INTEGER
    ,EventUser VARCHAR(12));
```

Figure 2: Creation of the Activity Table

For the activity table, at first five different columns are needed. The creation of these columns can be seen in figure 2. The first one is the "ActivityCaseID" column, which is a unique number for each case. A case represents one single position of an order and all the steps belonging to this order. The "Activity" column is a textual representation of the current action, for example "Purchase Requisition", "Purchase Order", or "Invoice Receipt". The "EventTime" column provides a timestamp consisting of the date and the time when the action was performed. "Sorting" provides a classification of the normal order, in which the activities should be performed; "Purchase Requisition" should, for example, in a normal case happen before "Purchase Order", which again should happen before "Invoice Receipt". The "EventUser" is the SAP system user, who created the respective activity, for example, booked the purchase requisition or order in the system.

Three more columns are generated by a Celonis procedure. One is the "Lifecycle" column, which is derived from the "Sorting" and "Timestamp" columns and represents the position in a case's lifecycle. Another one is the "Case_Num_ID", which gives each case (defined by "ActivityCaseID") an integer number that is easier to process. Finally, a "PrimaryKey" is added for each position of the activity table.

To fill the activity table with values, for every activity different SAP tables have to be joined to meet certain criteria. To create the purchase requisition tables, for example, the tables EBAN and EKPO

```
INSERT INTO CELONIS_P2P_ACTIVITIES(
SELECT
    (EKPO.MANDT||EKPO.EBELN||EKPO.EBELP)
    AS ActivityCaseID,
    'Purchase Requisition' AS Activity,
    EBAN.BADAT AS EventTime,
    10 AS Sorting,
    EBAN.ERNAM AS EventUser
    FROM EBAN
    JOIN EKPO ON EBAN.MANDT = EKPO.MANDT
    AND EBAN.EBELN = EKPO.EBELN
    AND EBAN.EBELP = EKPO.EBELP);
```

Figure 3: Join of Tables EBAN and EKPO

have to be joined, as it can be seen in figure 3.

Further, we join EKPO and EKKO to get the purchase orders, EKPO and EKBE to get the invoice receipts, or EKPO with CDPOS and CDHDR to get deleted positions or changed items. A deep understanding of the relevant tables is necessary. For example deletion of a position can be seen in table CDPOS (Change Document Items). The column "table name" should equal 'EKPO' to refer to the purchasing document and it must be deleted (so the fieldname equals 'LOEKZ' and the new value equals 'L').

Once we included all needed activities in the activity table, we created the case and the process tables. The creation of the process table is quite straightforward as it only maps an "Activity_ID" to each activity in the activity table.

Additionally to these four columns, one can choose to add further columns. It could, for example, be interesting to know the vendor for each case, as well as the ordered material and its price and amount.

4.3 Create Data Cubes and Data Models in Celonis

Once we have the activity, case and process tables we can create data cubes and models in Celonis. To do so we first give Celonis access to the respective database scheme in HANA, where the tables are located and choose this scheme as our data store. Then we define the tables Celonis should use to create our data model, in our case the activity, case and process tables created in step 2.

As a next step we can configure this data model and define, which tables and columns Celonis should use for each attribute. Finally we have to create at least one foreign key to connect the activity and case table.

4.4 Create Analysis

As soon as the data cube and data model is set up and configured properly, we can create our analysis. To do so we have to create a new document based on our data model, choose a suitable format and Celonis automatically creates a process model based on our dataset. In figure 4 an exemplary process model can be seen. It represents the first part of the standard purchase-to-pay process going from purchase requisition to purchase order, to goods receipt and to invoice receipt without consideration of specific deviations like deleted positions or changed order of activities. As one can see most cases in the example do not have a purchase requisition, but start directly with the purchase order and not all cases go through all of the activities.

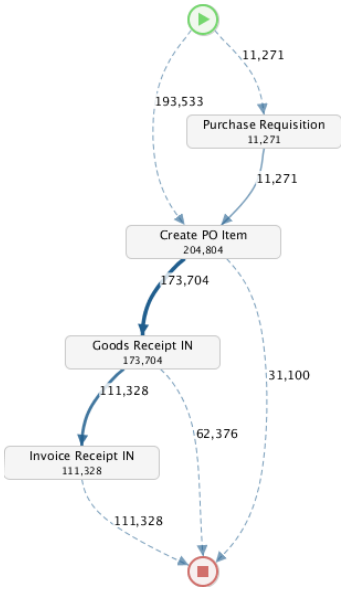


Figure 4: Exemplary Process Model

4.5 Investigate Analysis, Filter and Explore Processes

Once the process model is created there are many possibilities to explore it and to display process deviations. The model depicted in figure 4, e.g., is a strong simplification of all the processes that are derived from the used event log. Depending on the filter one choses, the tool displays a specific number of variants – which equal our different process paths. With the currently used data set we have 74 variants of the process, each showing a different path that is used by one or more processes. Many processes, for example, use a path that is similar to the one shown in figure 4, but where the invoice receipt was created before the good receipt creation. Since the order of these two activities can be swapped without consequences, this is therefore no indicator for fraudulent behaviour. A different path shows, e.g., all the deleted orders and another one all the orders where the price of the good was changed after the order was created. These deviations may, for example, be more interesting to check for occurring fraud cases. If needed, different filters can also be used, e.g., hide a certain activity or only show processes that start or end at a certain activity.

Besides these possibilities to visualize the different process paths, Celonis provides a lot more features to investigate the analysis. It contains, among others, a case explorer, where the path of a specific case can be analysed, the possibility to define different charts and tables and diverse useful statistics. Generally the visualisation possibilities are extensive.

4.6 Refine Analysis to Detect Further Fraud Cases

We analysed all process deviations in our dataset and found indicators for fraud. For example we have identified hints that show a double issue of an invoice or the non-purchase fraud.

In double issue of an invoice the fraudster receives two or more invoices for one product to get a double payment from the vendor. We have seen 724 fraud cases where two invoices have been received for one product as it can be seen in figure 5. Analysing this data in detail reveals that in some cases the sum of the multiple invoices exceeds the amount of the prize for the ordered product, which is an indicator for fraud.

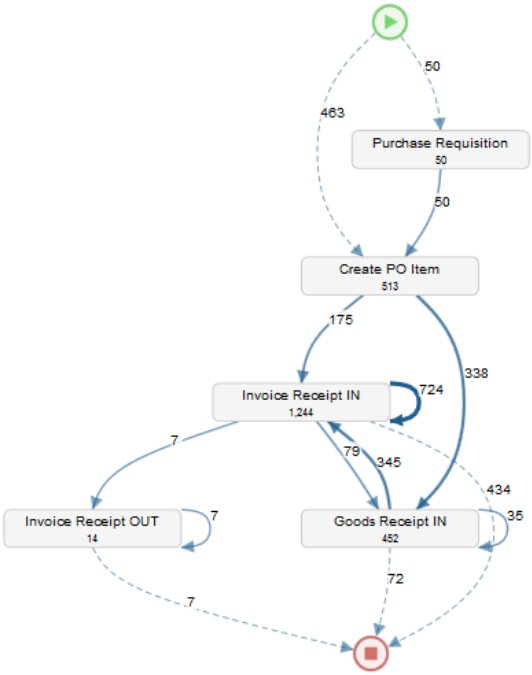


Figure 5: Double Received Invoices for one Product

As a further example we were able to identify 41 cases, where we have not received a good, but the invoice was issued as displayed in figure 6. In the so-called non-purchase fraud, a certain good is not delivered but the invoice paid. This is a further strong indicator for fraud.

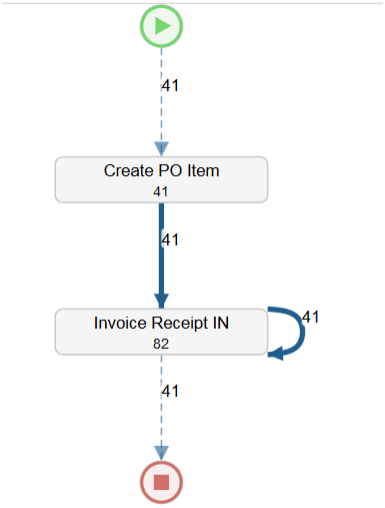


Figure 6: Invoice receipt without goods receipt

However, there are fraud cases included, that cannot be spotted through process deviations. One example is the kickback scheme. A fraudster and a complice vendor agree on an overpayment and share the over-paid amount. This fraud can be analysed comparing different process instances and looking for deviations. Celonis further offers the possibility to create key figures using SQL. A key figure “over-average payment” can also be added to spot the respective fraud. We plan to further improve our fraud detection strategy in the next project phase of the HPI.

5 Conclusion and Outlook

Up to now, we identified all necessary tables of the purchase-to-pay business process and loaded the tables into HANA. We then created the necessary Celonis tables to be able to display the business process with all deviations. Our preliminary fraud analysis showed fraud cases like double issue of an invoice and the non-purchase fraud. We further analysed cases that cannot be identified through process deviations, but as data irregularities in certain columns or by comparing different process instances.

In the next project phase, we plan to improve our fraud detection strategy by including further KPIs and to examine fraud in real-time. Instead of loading data into HANA, our aim is to use an ERP system running on HANA. Our identified fraud detection strategy should display fraud while it happens. We will test our strategy in a scientific experiment, where fraud should be perpetrated and examined in real-time. For more information please refer to our project extension application.

Sources

[1] ACFE, „Report to the Nations on Occupational Fraud and Abuse,“ Association of Certified Fraud Examiners, Austin, Texas, USA, 2012.

[2] A. Bönner, M. Riedl und S. Wenig, Digitale SAP®-Massendatenanalyse: Risiken erkennen - Prozesse optimieren, Berlin (Germany): Erich Schmidt, 2011.

[3] D. Coderre, Computer Aided Fraud Prevention and Detection: A Step by Step Guide, John Wiley & Sons, 2009.

[4] C. Phua, V. Lee, K. Smith und R. Gayler, A Comprehensive Survey of Data Mining-based Fraud Detection Research., 2010.

[5] Y. Yannikos, F. Franke, C. Winter und M. Schneider, „3LSPG: Forensic tool evaluation by three layer stochastic process-based generation of data,“ in *Computational Forensics Vol. 6540*, Berlin/Heidelberg, Springer Verlag, 2011, pp. 200-211.

[6] "SAP Datasheet," [Online]. Available: http://www.sapdatasheet.org. [Accessed 18 03 2015].

Comparison of Image Classification Models on Varying Dataset Sizes

Timur Pratama Wiradarma
Hasso - Plattner - Institut
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
pratama.wiradarma@student.hpi.uni-potsdam.de

Christian Hentschel
Hasso - Plattner - Institut
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
christian.hentschel@hpi.de

Harald Sack
Hasso - Plattner - Institut
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
harald.sack@hpi.de

Abstract

This paper aims to compare two competing approaches for image classification, namely Bag-of-Visual-Words (BoVW) and Convolutional Neural Network (CNN). Recent works have shown that CNNs have surpassed hand-crafted feature extraction techniques in image classification problems. The success of CNNs can be mainly attributed to two factors: recent advances in GPU supported computation and the availability of large training datasets for selected applications scenarios. In classification scenarios where only limited training data is available BoVW-based approaches may still perform better than CNNs. This assumption makes BoVW a viable option for a smaller training datasets. In this paper, we verify this hypothesis by training both approaches on a varying number of training images. We conduct our experiments using the publicly available 2012 ImageNet Large Scale Visual Recognition Challenge dataset and limit the training data to smaller randomly selected fractions. Since training a CNN is only reasonably possible using GPU hardware support, we largely benefit from the Future SOC infrastructure.

1 Introduction

Content-based image classification is an important means to address the challenge of search and retrieval in large image datasets such as community and stock photo galleries. Typically, manual tagging is impossible due to large amount of visual information. Research efforts have therefore been focusing on (semi-)automatic approaches for generating content-descriptive tags for visual information. Bag-of-Visual-Words (BoVW) [12] has been representing the state-of-the-art in image classification for couple of years, until it was outperformed by Convolutional Neural Networks (CNN) [1]. Their success should mainly be attributed to the fact that differing from BoVW approaches, which use hand-crafted visual feature descriptors, CNNs *learn* visual feature descriptors as part of the training process. Thus, the derived features can represent the data much better which usually leads to more accurate models. While the general idea of using CNNs in image classification is not new [8] only recently the availability of large scale computing power as well as large training datasets – assembled using crowd sourcing – made CNNs successful. Generally, a CNN model is trained on a GPU [1, 4, 11], which offers a high degree of parallelization and enables the model to learn from thousands of hand-labeled training images. While one could argue that the necessity for powerful computing hardware will less likely be a constraining factor in future due to technological progress, providing large amounts of manually annotated training data cannot be easily substituted. For many classification scenarios it is simply impossible to provide a sufficiently large amount of labeled images as it is a tedious and time consuming task. Therefore, in scenarios with limited amount of training data available, hand-crafted feature extractors may provide better results since CNN will likely fail to extract good features.

As the main contribution of this paper, we intend to help researchers and practitioners to decide, which approach best suits certain use cases. Therefore, we compared the two aforementioned approaches (BoVW and CNN) with respect to the classification accuracy obtained for varying amount of training data. We evaluated different training set sizes in order to identify the threshold where CNNs outperform BoVW approaches.

This paper is structured as follows: In the following section we briefly present relevant related work. In Section 3 we describe both approaches and present our experimental setting. Section 4 discusses the obtained results and Section 5 provides an outlook on future work.

2 Related Work

Since the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [1] where Krizhevsky et al. have presented a convolutional neural network-based approach which significantly outperformed all other solutions [1], CNNs have gained a lot of attention in image classification. Their success was partly attributed to the fact that the ILSVRC authors were able to provide massive amounts of training data (1.2 million images were manually assigned to more than 1000 categories) assembled in a huge crowdsourcing effort [10]. This gives rise to the question of how to handle classification tasks in which these huge amounts of training data are not available and where it would be too costly and time-consuming to provide these datasets. Furthermore, the authors in [1] state that training the proposed CNN model took between 5 to 6 days on two GTX 580 3GB GPUs, which even today represents hardware resources not available to every researcher. Many research efforts have therefore been focusing on alleviating both – the necessity for large scale training sets as well as highly optimized hardware resources.
The most promising results in this direction were given by approaches that reuse the penultimate layer of a deep CNN trained on a large dataset such as ImageNet can serve as a powerful image descriptor even when applied for different datasets. The authors in [4] show that these results could even be improved by fine tuning the pre-trained network on the novel dataset before using it for feature extraction. A work by Zeiler et al. [14] shows that a CNN model pre-trained on ImageNet and fine-tuned to the Caltech-101 dataset [5] outperforms Bag-of-Visual-Words implementation of Bo et al. [3]. However, as the ImageNet and Caltech-101 datasets tend to be highly overlapping in terms of object/scene categories (essentially many of the image classes in Caltech-101 are likewise found in ImageNet) and both collect photos of real-world objects the two datasets necessarily show very similar visual characteristics.
In this project we evaluate the impact of varying training set sizes on the classification performance of BoVW and CNN approaches. Our hypothesis is that BoVW performs better than CNNs for smaller training sets since a CNN model needs first to learn the respective feature maps whereas BoVW uses pre-engineered features.

[1] ILSVRC 2012 – http://www.image-net.org/challenges/LSVRC/2012/

3 Classification Models

In this section, we will describe briefly the Bag-of-Visual-Words and the Convolutional Neural Network implementations used throughout our experiments. We also present the hardware infrastructure we build upon to train the classification models. Finally, we describe the train and test datasets we used for training and evaluating the respective classification models.

3.1 Bag of Visual Words

Bag of Visual Words (BoVW) is an adaptation of Bag of Words model that has been successfully applied for text classification and retrieval and hence was extended to visual features for image classification. As a visual analogue for a *word* the BoVW model employs vector quantized visual features extracted at local image regions [12]. Typically Scale-Invariant Feature Transform (SIFT) [9] descriptors are computed at a dense grid of sampling points. These local features are aggregated into a global image descriptor that encodes the frequency distribution of individual visual words within an image. In our experiment, we use Vector of Locally Aggregated Descriptors (VLAD) [2, 6] encodings as provided by the VLFeat implementation [13] and train linear SVM (Support Vector Machine) models for each category (one-vs.-rest). This implementation was extended in order to make use of the SMP architecture provided by the HPI Future SOC Lab[2].

3.2 Convolutional Neural Networks

A Convolutional Neural Network is the variation of a multilayer neural network, which learns local filter kernels that are each convolved with the image to produce feature maps. Each map is subsampled (usually with mean or max pooling). These local filters exploit the strong local spatial correlations present in an image. Connected to an arbitrary number of convolutional layers may be any number of fully connected layers. These are identical to the layers in a standard multilayer neural network (for a more detailed description of the architecture, see [11, 1]).
In our experiments we used the CNN architecture that has been successfully applied in the 2012 ILSVRC[1]. A replication is provided by the Caffe framework [7] (with small modifications to the order of pooling and normalization steps – pooling is applied before normalization). The CNN architecture consists of 5 convolutional-layers, 3 fully-connected layers, with the last layer representing the output classes for the 1.000 ImageNet categories.
Following [1], we resize the input images to 256×256 pixels. We also subtract the mean – obtained by aver-

[2] HPI Future SOC Lab – http://hpi.de/en/research/future-soc-lab.html

Ratio	no. images
5 %	622
10 %	1,242
20 %	2,485
40 %	4,969
60 %	7,455
80 %	5,082
100 %	6,352

Table 1. Training sets generated by randomly sub-selecting images from the 2012 ILSVRC training set according to the denoted ratio.

aging the pixel values from all training images – from each input image. Unlike proposed by the authors in [1], we don't perform any further data augmentation, since it was reported to contribute only slightly to the results.[3]

The HPI Future SOC Lab provides two Nvidia Tesla K20X GPUs that allowed us to train different CNNs multiple times.

3.3 Dataset

We run our experiments using the 2012 ILSVRC training data, since it provides one of the largest manually annotated data collection available. We split the dataset into subsets with increasing number of images by taking random percentages of the entire ILSVRC 2012 training set. In total we generated 7 datasets (cf. Table 1).

In order to avoid having to train the entire dataset multiple times, we selected 10 categories to train models for. These categories have been chosen by taking the 5 easiest and 5 most difficult classes to train considering the mean error from the top 5 predictions from all submissions to the 2012 ILSVRC[4] (cf. Table 2).

We trained models for each category for every training data subset using both approaches – BoVW+VLAD as well as CNN. The duration of the training process varies for each approach as well as for each subset. While it took about 10 minutes to train the CNN model using the smallest dataset (5%) on the Tesla K20X GPU and about 15 minutes to train VLAD models on a 48-core machine, training the entire (100%) dataset took about 160 minutes for the CNN model as well as 190 minutes for the VLAD models.

In order to validate the classification accuracy of the individual models and compare the different approaches we built a validation dataset using all images

[3]We use the following training parameters: momentum 0.9, weight decay 5.10^{-4}, initial learning rate 10^2 (which is decreased by a factor of 10, for every 20 epochs) and we train a total 90 epochs for each dataset.

[4]See `http://image-net.org/challenges/LSVRC/2012/ilsvrc2012.pdf` for more information.

Easiest	mean error
geyser	0.001
odometer	0.011
canoe	0.013
yellow lady's slipper	0.015
website	0.015
Most difficult	**mean error**
Ladle	0.877
Hatchet	0.857
Spatula	0.833
Muzzle	0.832
Hook and Claw	0.805

Table 2. Easiest and hardest categories to classify based on evaluation of the mean error of the top 5 predictions from all submissions to the 2012 ILSVRC.

from the respective classes from the ILSVRC 2012 validation set (in total 500 images – 50 per class – were selected).

4 Results

Figure 1 shows the average precision (AP) scores obtained for different training set sizes using both approaches. When considering the 5 "easy" classes ('geyser', 'odometer', 'canoe', 'yellow lady's slipper', 'website') BoVW and CNN reach similar accuracy using 60% of the entire training dataset. However, when using datasets much smaller in size BoVW clearly outperforms the CNN-based approach achieving close to perfect ($AP \approx 1.0$) results for datasets comprising only 5% of the original training data.

With respect to the most difficult categories when considering the 2012 ILSVRC results, BoVW also performs better than the CNN approach for most categories. Surprisingly, however, both approaches show a significant drop in accuracy when increasing the training data to more than 80% of the original dataset. At this point of our experiments, we can only explain this by certain image characteristics not fully reflected in the testing data.

As discussed in Section 3 the CNN approach learns filter kernels using the training data while the BoVW approach makes assumptions about the characteristics of the training data reflected in the extracted local features (i.e. SIFT). By adding more unlabeled images to the training set, the CNN approach will most likely be able to learn better features and thus will show an increased classification performance. In order to validate this assumption, we added images from an additional 90 classes from the ImageNet dataset and trained a CNN model for a total of 100 classes using the same training subset ratios as before. Figure 2 shows the mean average precision scores (obtained from predict-

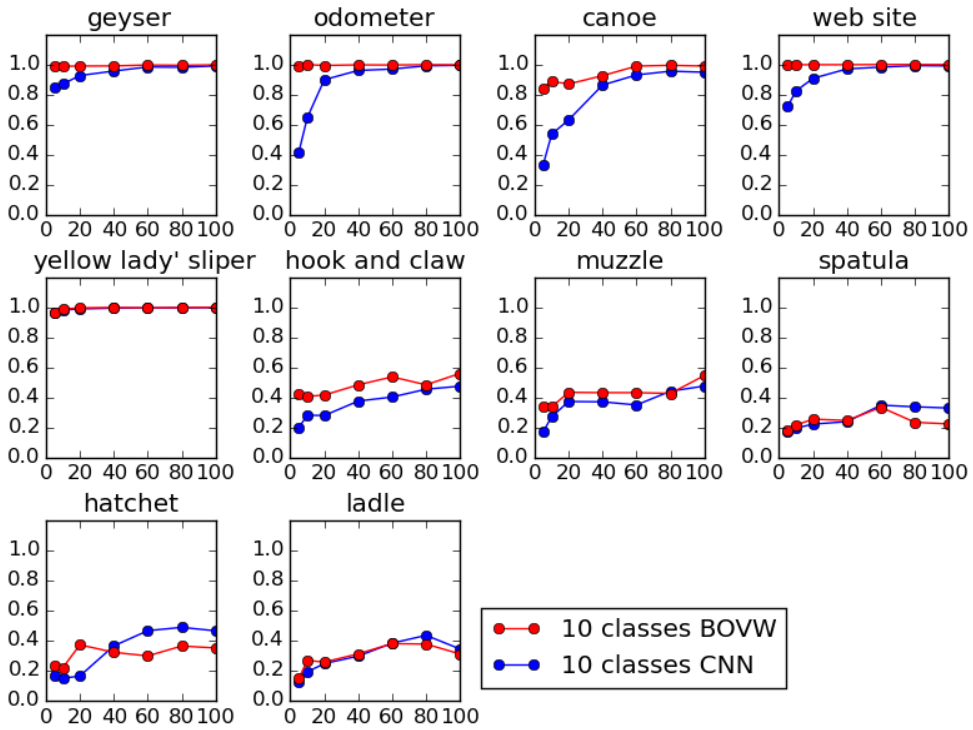


Figure 1. Average Precision Scores obtained for different training set sizes comparing BoVW and CNN approach

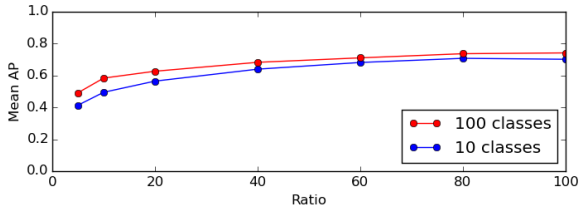


Figure 2. Mean average precision scores obtained from training using more data.

ing only the initial 10 classes on the same testset as before). As can be seen the CNN model indeed performs better, when increasing the number of training images.

5 Future Work

In this work we presented a comparison between two competing approaches for image classification, namely Bag of Visual Words and Convolutional Neural Networks. We analyzed both approaches for the influence of varying training set sizes and have shown that BoVW outperforms CNN when only small amounts of training data is available. Recent research efforts have focused on reusing CNN models that have been trained on a large image corpus (i.e. the entire ImageNet dataset) and use the obtained models for feature extraction (i.e. by taking the penultimate fully connected layer as a feature descriptor and training a Support Vector Machine using these features). In a next step we therefore intend to compare the results presented here to those achieved using a pre-trained model. We will further investigate how pre-trained models perform when used as feature extractors on a dataset that differs visually from the one used to train the CNN (e.g. using collections of art images or video keyframes).

References

[1] G. E. Alex Krizhevsky, Ilya Sutskever. Imagenet classification with deep convolutional neural network. *NIPS*, 2012.

[2] R. Arandjelović and A. Zisserman. All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[3] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 660–667, Washington, DC, USA, 2013. IEEE Computer Society.

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *CoRR*, abs/1405.3531, 2014.

[5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an

incremental Bayesian approach tested on 101 object categories. In *CVPR 2004 Workshop on Generative-Model Based Vision*, 2004.

[6] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311, jun 2010.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, 2004.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[12] C.-F. Tsai. Visualizing and understanding convolutional networks. *ISRN Artificial Intelligence*, 2012.

[13] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[14] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

Sentiment Analysis for the needs of benchmarking the Energy Sector – Project Report –

Witold Abramowicz Wioletta Sokolowska Tymoteusz Hossa Jakub Opalka
Karol Fabisz Mateusz Kubaczyk Milena Cmil
Department of Information Systems
Faculty of Informatics and Electronic Economy
Poznan University of Economics
Al. Niepodleglosci 10, 61-875 Poznan, Poland
firstname.lastname @kie.ue.poznan.pl

Abstract

This report gives an insight into activities performed in the field of sentiment analysis using analytical and computational power of SAP HANA, so to expand the possibilities of the previously created working prototype of an in-memory Business Intelligence solution for the support of decision making in the field of energy sector. The report provides information on the project main objectives, used HPI Future SOC Lab resources, findings as well as next steps envisioned.

1. Introduction

The energy system operators are dealing with an influx of diverse information from multiple sources that have a great influence on their decisions. On daily basis they need to analyze vast amount of data on energy supply and demand at different time span, energy prices, technical data and much more. The currently observed deployment of Smart Grid infrastructures implies the emergence of previously unknown problems associated with processing and analyzing large and diverse data sets in a real time. Therefore, the energy utilities (and their business analysts) are becoming to tackle with the *Big Data*[1]challenge[2]. Moreover, the shift from traditional to more customer-oriented energy market can be noticed. Customers are becoming crucial market players as they now have an access to tools and information enabling them for reducing the energy consuption and taking control of their own energy supply needs [4]. Finally, as they are more socially interconnected, they are willing to comment, discuss and rate the energy suppliers and their offers provided on different Internet portals (professional fora or popular social media websites).

Taking into account all the aforementioned issues, the energy utilities will need to respond to the changing market needs and learn how to efficiently tackle with not only structured data (i.e. historical observations of energy generation and consumption) but especially with vast amount of data coming from different types of unstructured sources (like tweets, comments, spatial data etc.). Only, by handling both they will be able to gain an invaluable information helping to i.e. predict the customers fall-out rate anddynamically adjust the offer to market expectations. The adoption of data coming from social media, satellites or sensors into analytics creates new reguirements towards defining and creating an adequate analytical tool.

The project is a continuation of four projects: Quasi Real-Time Individual Customer Based Forecasting of Energy Load Demand Using In Memory Computing; Forecasting of Energy Load Demand and Energy Production from Renewable Sources using In-Memory Computing and Prototype of an In-Memory Business Intelligence Solution for the Support of Forecasting of Energy Load Demand and Smart Data Analysis for the Support of Rational Decision Making in the Energy Sector ran previously under HPI Future SOC Lab. These projects focused on the analysis of structured data (coming from the energy consumption and energy generation from renewable sources) and to some extent the unstructured data for the needs of business analysts working in the energy sector. Within the two last, taking advantage of the previously achieved outcomes, the dashboard like solution was developed and enriched to equip business analysts with up-to-date forecasts and sentiment analysis results. However, in terms of automatically extracting and performing the analysis of the unstructured data sources using SAP HANA, there were only very few issues explored in detail. The customer opinions on energy unitilites were extracted from one forum directly to the SAP

[1]The BIG DATA is described with the following terms: Volume, Velocity, Variety and Veracity, being also known as 4V model [1], [2], [3], [5].

[2]The Big Data issue within the energy sector will not be discussed in detail in this report.

HANA and basic sentiment analysis was conducted with the usage of the default SAP HANA Text analysis tool. In current project the emphasis was put on extending the scope of the sentiment analysis for the needs of rational decision-making by focusing on analyzing more than 10 000 comments, developing a new method while using a domain specific dictionary and thus expanding the possibilities of previously created working prototype of our Business Intelligence solution.
The document presents the attemps undertaken within this project and it is organized as follows. First, the project aims are shortly presented. Then, the used Future SOC Lab Resources are pointed and few technical details are given. Next, the obtained results are briefly summarized. The document concludes with final remarks.

2. Project idea

As already mentioned, the project reported in this document is a part of a cycle of undertakings aiming at building an analytical solution using SAP HANA to support business analysts in the energy sector. The main research activities focused on continuing our work on the analysis of the unstructured data coming from selected Internet sources in order to identify relevant information (e.g. a sentiment on the energy provider and relevant aspects of provider's offer). To realize this goal, we decided to extend our data set at least ten times in comparison to the previous project, where we focused on analysing the sentiment of around 900 opinions. In addition, data had to be extracted from several different sources. In parallel, we aimed at veryfing the already developed method using the newly acquired data and improving the accuracy of the so far proposed solution. Moreover, the idea was not only to improve the method but also to create a domain specific dictonary that would be added to the default ones used within the SAP HANA Text Analysis.

3. Future SOC Lab resources used

During the project, we accessed a standard physical machine with SAP HANA instance (12) with 24 cores and 64 GB RAM. Moreover, the work on the strictly unstructured data required to use SAP HANA Text Analysis.
We used Python programming language to create a code that would scrap all reviews from identified sources [3]. Our Python code visited each of subwebpages and gathered only the opinions on energy providers. 10 012 opinions were automatically gathered, finally 9583 of them were uploaded to SAP

[3] In the previous project, Python was used for performing the ETL process and running the method that would assess the sentiment for a particular energy provider.

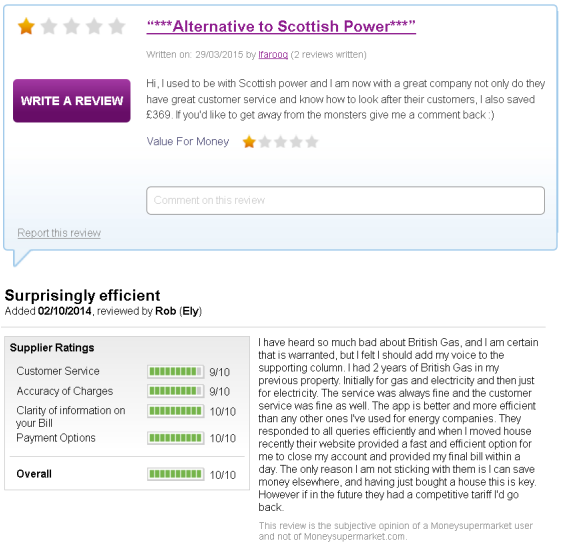


Figure 1. Opinion of energy provider's client - example. Source `http://www.moneysupermarket.com`, `http://www.reviewcentre.com/Electricity-Suppliers`

HANA. The average length of each opinion was greater than 160 characters. Figure 1 presents an example of opinion available on the aforementioned websites. these comments were then stored together in one coherent corpus (please see 4.1). The whole corpus analysis was conducted in AntConc software [4] - a corpus analysis toolkit for concordancing and text analysis. The newly developed method enables for loading data into SAP HANA, conducting text analysis with the use of FullText Index on VOICEOFCUSTOMER configuration. Then a sentiment of each comment is calculated (to learn more about the results please go to 4.2). The method was implemented in SAP HANA using SQLScript. In this way the overall computing time shortened to 12 sec.
Moreover, the customized SAP HANA Dictionary was created. The final dictionary is an XML file (coherent with a SAP HANA custom dictionary syntax) The XML file was implemented in SAP HANA as a **hdbtextdict** file and added to the repository. Next step included creating **hdbtextconfig file** where a new dictionary was added to the list of default dictionaries. As a result, it was possible to run sentiment analysis using the new set of dictionaries (to learn more about the results please go to 4.1) .

4. Findings

The results of our project consist of two main elements. One of our project's goals was to create domain

[4] AntConc software website: `http://www.laurenceanthony.net/software.html`

specific dictionary in SAP HANA consisting of terms connected to the energy industry's issues. The second goal was to implement a method which will be able to judge the sentiment (positive, neutral or negative) of a single customer's opinion. The hyphotesis was that the method created under this project will perform better than the method prepared within the last edition (with reference to the Gold Standard).

4.1. Custom dictionary

To achieve the first goal we conducted a corpus analysis in AntConc software. The corpus consist of 15798 word types and 946047 word tokens and includes 9583 individual comments on 17 British energy suppliers. There was an attempt to find all possible variants of energy suppliers' names. Although the reviews refer only to the 17 utilities, in the dictionary there are also different companies mentioned that occurred in the corpus, mainly the less significant energy providers (for instance previous suppliers or utilities in the comparative phrases). Table 1 presents an example of a companys name and its variants. The standard form British Gas occurred in the corpus 905 times. However, a lot of people prefer using different, usually shorter variants like abbreviations (for example Brit Gas) or acronyms (BG).

British Gas	
Name's variants	A number of instances in the corpus
BG	339
British Gas	905
Brit Gas	4
BritGas	1
Scottish Gas	26

Table 1. British Gas - standard form and variant names.

In the corpus analysis the emphasis was also put on searching the most significant words and phrases. The analysis of the bigrams, trigrams and four-grams which occurred mostly was conducted. In this way it was possible to identify issues that are more important than others. Subsequently, a list of words characteristic for energy invoices[5] was defined on the basis of the analysis of three examples of particular energy invoices and the Reuters Financial Glossary.
The final dictionary is an XML file (coherent with SAP HANA custom dictionary syntax). It comprises of 14 categories and 110 words and phrases connected with the energy suppliers and all electricity relevant issues.

[5] http://www.momentumenergy.com.au/system/files/images/Sample_Invoice/Invoice%20Explainer.pdf

4.2. Sentiment analysis method

To perform evaluation of the implemented sentiment analysis' method we used the so-called gold standard. The precision, recall and F-measure parameters were calculated with the reference to the human annotated opinions.
First of all, we tested our new method on the small data set (656 comments) from our previously run Future SOC Lab project. Moreover, we compared the obtained results with the latest ones. Table 2 presents the gold standard values for the method of sentiment analysis we implemented in the previous project.

	Precision	Recall	F-measure
Positive	85%	71%	77%
Neutral	51%	59%	55%
Negative	57%	73%	64%

Table 2. Gold standard values of precision, recall, F-measure parameters for positive, neutral and negative comments (the previous SA method on the small data set).

In the Table 3, we can see that new method is performing almost as good as the previous one. The experiment showed that the identification of the positive sentiment among comments is even better. It must be emphasized that the processed data set contained relatively balanced customer reviews.

	Precision	Recall	F-measure
Positive	79%	81%	80%
Neutral	8%	44%	13%
Negative	85%	45%	59%

Table 3. Gold standard values of precision, recall, F-measure parameters for positive, neutral and negative comments (the new SA method on the small data set).

The second step was to apply the new SA method on a big data set (9583 comments). Table 4 presents the results of the experiment. The identification of the positive and negative sentiment among comments is very good, as it is indicated by high values of the precision and recall measures. However, it is fair to say that our new method is underperforming in field of identifying neutral sentiment in comments. It should be also

pointed that the data set processed this time contained many customer reviews which are clear to judge.

	Precision	Recall	F-measure
Positive	91%	75%	82%
Neutral	26%	15%	19%
Negative	74%	95%	83%

Table 4. Gold standard values of precision, recall, F-measure parameters for positive, neutral and negative comments (the new method on the big data set).

At the end, we also tested our method performance after applying the custom dictionary to the set od dictionaries used within the Voice of Customer analysis in SAP HANA. The obtained results are presented in Table 5. Unfortunately, the experiment showed that precisely extracted words and phrases concerning electricity and gas topics do not give more relevant and satisfying results. Surprisingly, all gold standard values were worse than in the case wof the default set of dictionaries available in SAP HANA Text Analysis tool.

	Precision	Recall	F-measure
Positive	85%	68%	75%
Neutral	4%	2%	3%
Negative	64%	85%	73%

Table 5. Gold standard values of precision, recall, F-measure parameters for positive, neutral and negative comments (the new method on big data set, using domain specific dictionary).

5. Conclusion

The conducted research using the Future SOC Lab resources allowed us to deepen our knowledge about SAP HANA, especially its text analysis capabilities. Therefore, we were able to expand the possibilities of previously created working prototype of Business Intelligence solution by enriching our method for analyzing the sentiment.
The main conclusions from our experiments are as follows:

- Sentiment analysis was conducted on a bigger unstructured data source (9583 comments) and its results are satisfying.
- The default set of dictionaries in SAP HANA Text Analysis can be improved by enriching it with a domain specific dictionary. However, its development must be preceded by a thorough words' corpus analysis.
- Working with code and queries is too technical for any practical use, therefore a lot of work has to be done before the data may be visualized in an easy to read and interpret manner;

To conclude, the our research proved that business analyst may take an advantage from using our application. The later employs the analytical and computational power of SAP HANA, PAL, R, SAP HANA Text Analysis and Tableau Software, especially when the use of more advanced methods than the default ones is required. The main lesson that comes from activities perforrmed within this project is that the default capabilities of SAP HANA Text Analysis should be supported by methods improvement and expanded with domain specific dictionaries in order to achieve better results.

References

[1] K. Fabisz, A. Filipowska, and T. H. R. Hofman. Profiling of prosumers for the needs of energy demand estimation in microgrids. In *Proccedings of the 5th International Renewable Energy Congress*, 2014.

[2] A. Filipowska, K. Fabisz, T. Hossa, M. Mucha, and R. Hofman. Towards forecasting demand and production of electric energy in smart grids. In *Perspectives in Business Informatics Research, 12th International Conference BIR2013*, 2013.

[3] T. Hossa, A. Filipowska, and K. Fabisz. The comparison of medium-term energy demand forecasting methods for the needs of microgrid management. In *Proceedings of SmartGridComm, IEEE International Conference on Smart Grid Communications*, 2014.

[4] PwC. Utility of the future. a customer led shift in the electricity sector, April 2014.

[5] W. Sokolowska, J. Opalka, T. Hossa, and W. Abramowicz. The quality of weather information for forecasting of intermittent renewable generation. In J. Marx Gmez, M. Sonnenschein, U. Vogel, A. Winter, B. Rapp, and N. Giesen, editors, *INFORMATION AND COMMUNICATION TECHNOLOGY FOR ENERGY EFFICIENY, Proceedings of the 28th International Conference on Informatics for Environmental Protection (EnviroInfo 2014)*. Oldenburg: BIS-Verlag, Carl von Ossietzky University Oldenburg, Germany, 2014.

Model-based Quantitative Security Analysis of Mobile Offloading Systems under Timing Attacks

Tianhui Meng
Department of Mathematics and Computer Science
Freie Universität Berlin
Takustr. 9, 14195 Berlin, Germany
tianhui.meng@fu-berlin.de

Abstract

Mobile offloading systems have been proposed to migrate complex computations from mobile devices to powerful servers. While this may be beneficial from the performance and energy perspective, it certainly exhibits new challenges in terms of security due to increased data transmission over networks with potentially unknown threats. Among possible security issues are timing attacks which are not prevented by traditional cryptographic security. Metrics on which offloading decisions are based must include security aspects in addition to performance and energy-efficiency. This project aims at quantifying the security attributes of mobile offloading systems. The offloading system is modeled as a stochastic process. Experiments are conducted to obtain the parameters for the model.

1 Introduction

Cloud computing has become widely accepted as computing infrastructure of the next generation, as it offers advantages by allowing users to exploit platforms and software provided by cloud providers (e.g., Google, Amazon and IBM) from anywhere on demand at low price [5]. At the same time, mobile devices are progressively becoming an important constituent part of everyday life as very convenient communication and business tools with a wide variety of software covering all aspects of life. The concept of computation offloading has been proposed with the objective to migrate large computations and complex processing from mobile devices with energy limitations to resourceful servers in the cloud. This avoids a long application execution time on mobile devices which results in large power consumption.

Along with the benefits of high performance, the offloading system witnesses potential security threats including compromised data due to the increased number of parties, devices and applications involved, that leads to an increase in the number of points of access. Security threats have become an obstacle in the rapid expansion of the mobile cloud computing paradigm. Significant efforts have been devoted in research organisations and academia to build secure mobile cloud computing environments and infrastructures [2]. However, work on modelling and quantifying the security attributes of mobile offloading system is rare.

We propose a state transition model of a general mobile offloading system under the specific threat of timing attacks. In a timing attack the attacker deduces information about a secret key from runtime measurements of successive requests [4]. This process can be interrupted by frequently changing the key. From the security quantification point of view, since the sojourn time distribution function in different system states may not always be exponential, the underlying stochastic model needs to be formulated as a Semi-Markov Process (SMP). Computing the combined system security and cost trade-off metric, we investigate the cost for a given security requirement. Our results will give security metrics on which offloading decisions are based. The remainder of this report is structured as follows. In Section 2, we develop a Semi-Markov model for a general offloading system under the threat of timing attack. The steady-state probabilities leading to the computation of steady-state security measures is addressed in Section 3. Section 4 shows the implementation details and results. And finally, the report is concluded and future work are presented in Section 5.

2 Security Analysis based on SMP Model

The state transition model represents the system behaviour for a specific attack and given system configuration that depends on the actual security requirements. Semi-Markov Processes (SMPs) are generalizations of Markov chains where the sojourn times in the states need not be exponentially distributed [3].

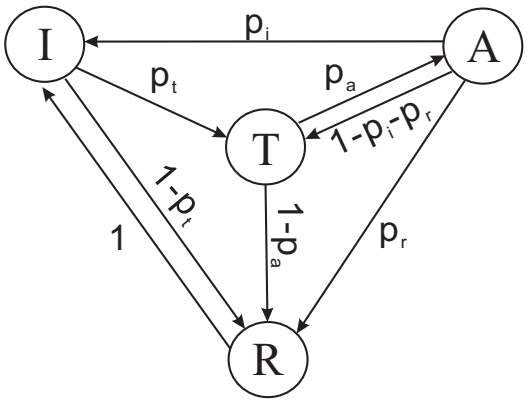


Figure 1. State transition diagram for a generic offloading system

2.1 Behaviour of attacker and system

In the offloading systems we consider, a server master key is used for the encryption and decryption operations of user data. In order to improve security, the server should regularly or irregularly change the master key. The system has to process all user-files with both the new and the old master key. In this process, the system does not accept any other user commands. In timing attacks to our offloading system, an attacker will continue to send requests to the server and the obtained service will be properly performed by the server. In addition the attacker records each response time for a certain service and tries to find clues to the master secret of the server by comparing time differences from several request queues. If the attacker successfully breaks the secret information from the timing results, he can read and even modify other users information without authorisation.

2.2 The Model

Fig. 1 depicts the state transition model we propose for describing the dynamic behaviour of a generic offloading system. This system is under the specific threat of timing attacks conducted by random attackers. We describe the events that trigger transitions among states in terms of probabilities and cumulative distribution functions.
The states and parameters of the SMP model are summarized here:

- I Initial state of the offloading system after star up
- T Timing attack happening state
- A Attack state after the attacker get the secret of the system
- R Rekeying state
- p_t probability that an attacker begin to conduct a timing attack to the system
- p_a probability of attack system confidentiality after a successful timing attack
- p_i probability that the system return initial state by manual intervention
- p_r probability that the attack is terminated due to rekeying operation

After initialisation, the system is in the good state I. The sojourn time in state I is the life time of the system before an attacker starts a timing attack or the system renews its key. We assume there is only one attacker in the system at one time. If an attack happens, the system is brought to state T, in which the timing attack takes place and the attacker decyphers the encryption key by making time observations. So while the system is in state T, the attacker is not yet able to access confidential information. We assume that it takes a certain time to perform the timing attack after which the attacker will know the encryption key and the system moves to the compromised state A. Changing the encryption key can prevent or interrupt a timing attack. During rekeying the system is in state R. The challenge is to find an optimal value for the rekey interval. The rekeying should certainly happen before or soon after the system enters the compromised state. Rekeying will bring the system back to the initial state I. If the attacker succeeds to determine the encryption key through time measurements confidential data will be disclosed which is assumed to incur a high cost. This can only happen if the system is in the compromised state A and we call the incident of entering the compromised state a security failure. One possibility is that one attacker stops himself and another attacker comes for a new timing attack. So the system is brought from compromised state A to another timing attack state T. The attack can also be stopped by manual intervention, i.e. triggering the rekey operation. This can happen either in the attack state T or in the compromised state A, both transitioning the system to the rekey state R from which it will return to the initial state.

2.3 Measures on SMP

The measures are defined in this work as system cost and confidentiality that are functions of the state probabilities of the SMP model.
The system loses sensitive information in the compromised state, and cost is also incurred when the system deploys a rekeying process regularly. The steady-state probabilities π_i may be interpreted as the proportion of time that the SMP spends in the state i. In our model, the rekeying cost and the data disclosed cost are both interpreted as the proportion of system life time, that is, the steady-state probability of the SMP. We define two weights c and its complement $1 - c$ for the two kinds of cost. We use normalization weights for simplicity. The system cost is defined as:

$$\mathit{Cost} = c\pi_A + (1 - c)\pi_R \ . \qquad (1)$$

where $\pi_i, i \in \{A, R\}$ denotes the steady-state probability that the SMP is in state i. $0 \leq c \leq 1$ is the weighting parameter used to share relative importance between the loss of sensitive information and the effort needed to rekey regularly. Similarly, if a timing attack to the offloading system is successful, the attacker obtains the master key and can browse unauthorised files thereafter. The entered states denote the loss of confidentiality. Therefore, the steady-state confidentiality measure can then be computed as

$$\textit{Confid} = 1 - \pi_A \ . \tag{2}$$

In order to investigate how system security will interact with the cost, we also define a trade-off metric. An objective function formed from the division of the security attribute confidentiality and system cost is created to demonstrate the relationship between the cost the system has to pay and the corresponding security system gain. The trade-off metric shows the how much security per cost you can obtain.

$$\textit{Trade} = \frac{\textit{Confid}}{\textit{Cost}} \ . \tag{3}$$

3 Semi-Markov Process analysis

In this section, we derive and evaluate the security attributes using methods for quantitative assessment of dependability.

3.1 DTMC steady-state probability

It was explained earlier in order to carry out the security quantification analysis, we need to analyse the SMP model of the system that was described by its state transition diagram. The steady-state probabilities $\{\pi_i, i \in X_s\}$ of the SMP states are computed in terms of the embedded DTMC steady-state probabilities v_i and the mean sojourn times h_i[6]:

$$\pi_i = \frac{v_i h_i}{\sum_j v_j h_j} \quad i, j \in X_s. \tag{4}$$

Assuming the existence of the steady-state in the underlying DTMC, it can be computed as

$$\overrightarrow{v} = \overrightarrow{v} \cdot \mathbf{P} \quad i \in X_s. \tag{5}$$

where $\overrightarrow{v} = [v_I, v_T, v_A, v_R]$ and $\mathbf{P}$ is the DTMC transition probability matrix which can be written as:

$$\mathbf{P} = \begin{array}{c} \\ I \\ T \\ A \\ R \end{array} \begin{array}{c} \begin{array}{cccc} I & T & A & R \end{array} \\ \begin{pmatrix} 0 & p_t & 0 & 1-p_t \\ 0 & 0 & p_a & 1-p_a \\ p_i & 1-p_i-p_r & 0 & p_r \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{6}$$

In addition, we have the total probability relationship:

$$\sum_i v_i = 1 \quad i \in X_s. \tag{7}$$

The transition probability matrix $\mathbf{P}$ describes the DTMC state transition probabilities between the DTMC states as shown in Fig. 1. The first step towards evaluating security attributes is to find the steady-state probability vector $\overrightarrow{v}$ of the DTMC states by solving Eqs. 5 and 7. We can get solutions:

$$\begin{aligned} v_I &= \frac{p_i p_a + 1 - p_a + p_a p_r}{\phi}, \\ v_T &= \frac{p_t}{\phi},\ v_A = \frac{p_t p_a}{\phi},\ v_R = v_I - \frac{p_i p_t p_a}{\phi} \end{aligned} \tag{8}$$

For the sake of brevity, we assume: $\phi = 2 + 2p_i p_a + p_t + p_t p_a - 2p_a + p_a p_r - p_i p_t p_a$.
In the next subsection, the DTMC steady-state probabilities are used to compute the SMP steady-state probabilities.

3.2 Semi-Markov model analysis

The mean sojourn time h_i in a particular state $i \in X_s$ is the other quantity that is needed to compute the SMP steady-state probabilities. We put the h_i again here:

- h_I the mean time the system spends before an attacker conducts a timing attack or rekey itself
- h_T the mean time before the attacker break the master secret of the server by timing attack
- h_A the mean time the system is losing information
- h_R the mean time for rekeying process

In our modle, some parameters can be obtained from experiments. The measurements we are in process of taking are based on a offloading server under timing attacks. We have built a timing attack demonstrator and measure the mean time for a successful attack which will be used as h_I. Some parameters, e.g. probability that an attacker begin to conduct a timing attack and attack system confidentiality after a successful timing attack will be assumed as an attacker. Other parameters used in our system can be tune by the system administrator, like the rekey probability p_r and the mean sojourn time in initial state h_I.
Here, we can compute the steady-state probabilities $\{\pi_i, i \in X_s\}$ of the SMP states by using Eqs. 4 and 8. Again, for the sake of brevity, we assume:
$\Phi = (p_i p_a + 1 - p_a + p_a p_r)h_I + p_t h_T + p_t p_a h_A + (p_i p_a + 1 - p_a + p_a p_r - p_i p_t p_a)h_R$. The solutions are presented as

$$\pi_I = \frac{p_i p_a + 1 - p_a + p_a p_r}{\Phi} h_I \tag{9}$$

$$\pi_T = \frac{p_t}{\Phi} h_T \tag{10}$$

$$\pi_A = \frac{p_t p_a}{\Phi} h_A \tag{11}$$

$$\pi_R = \frac{h_R}{h_I}\pi_I - \frac{p_i p_t p_a}{\Phi} h_R \tag{12}$$

Given the SMP model steady-state probabilities, various measures can be computed via Eqs. 1 to 3.

4 Implementation details and results

4.1 Timing attack implementation

We have implemented a file-upload server webpage based on Apache Tomcat 7.0.54. The server is deployed on a HP ProLiant DL980 G7 machine in Future SOC Lab. A timing attack demo-Client is also developed to upload a text file to server and record the response time. When a file is uploaded to the server, the content in it will be read and encrypted, then decrypted to show the original text. The encryption and decryption time are recorded and sent back to the client. The client visited the server through VPN and measured the time from uploading the file to receiving the response from server. First we record the time for one step of the attack, i.e. one service request and response time. For 1024 bit RSA(the modulus N is 1024 binary bit), p and q are both 512 bit. Using B.B. attack, an attacker can break the private key in 256 step. Because after recovering the half-most signicant bits of q, we can use Coppersmiths algorithm [1] to retrieve the complete factorization. Fig.2(a) and (b) show the completion time distribution of one step attack. We assume that each attack step is independent and identically distributed (i.i.d.). The cumulative distribution function of one complete attack finish time can be computed by iteratively convolution method. To simplify the computational process, we can do the convolution pairing. We obtain the time distribution for a complete timing attack as shown in Fig. 2 (c).

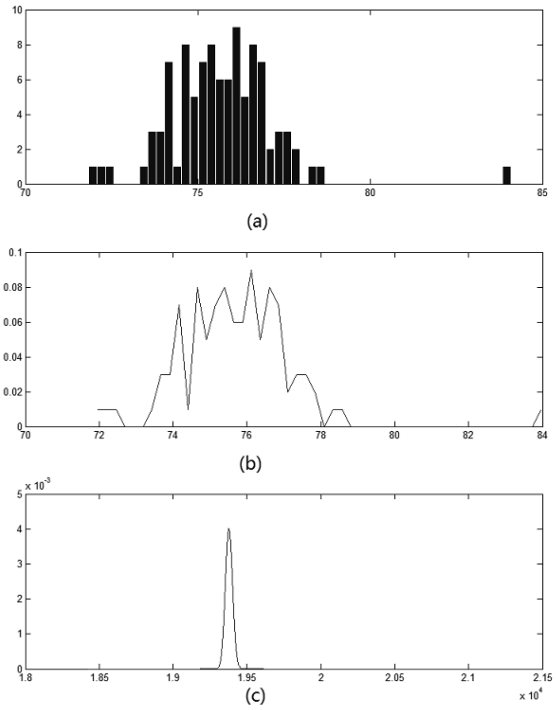


Figure 2. Time distribution of timing attack

4.2 Numerical Study

In this section we give numerical results as examples to show how one can evaluate security attributes of the SMP model defined in the previous sections using different measures.

First, we assume that the probability of a timing attack coming to the offloading system is equal to the one that the system will trigger its rekeying process, i.e., p_t = 0.5. The mean time the system spends before an attacker conducts a timing attack or it rekeys is h_I = 10 time units. Further, the probability that the attacker successfully cracks the system secret using a timing attack is p_a = 0.6 and the probability of an unsuccessful attack $1 - p_a$ = 0.4. The time taken by a successful timing attack is assumed to be h_T = 5 time unites. Besides, suppose that the probability that the system return initial state by manual intervention is p_i = 0.2 and probability of the attack is terminated due to rekeying operation is p_r = 0.5. Hence, the probability that the current attack stops and another timing attack affects the system is $1-p_i-p_r = 0.3$. We also assume the duration for a specific attack is supposed to be h_A = 3 time unites and rekeying process time is h_R = 1 time unite respectively.

Using the values given above as the model input parameters and Eqs. 9 - 12, we obtain the steady-state probabilities of the Semi-Markov process as:

$\pi_I = 0.6634$, $\pi_T = 0.2023$, $\pi_A = 0.0728$, $\pi_R = 0.0615$.

The steady-state probabilities π_i may be interpreted as the proportion of time that the SMP spends in the state i. For the assumed values of the input parameters, the proportion of time that the offloading system spends in the initial state I is approximately 66% of the whole system life time.

Fig. 3 shows the system cost measure as a function of

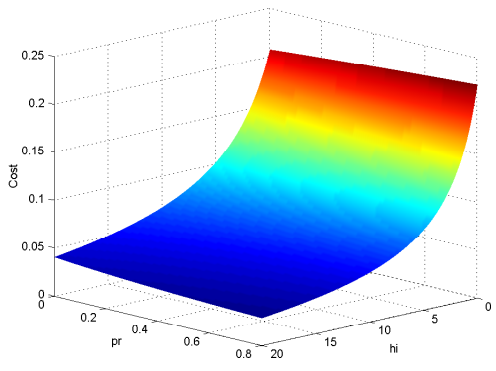


Figure 3. $Cost$ **as a function of** h_I **and** p_r

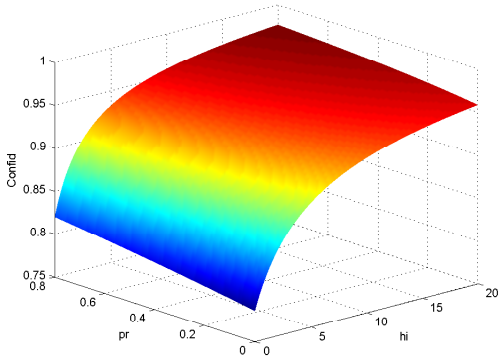


Figure 4. $Confid$ **as a function of** h_I **and** p_r

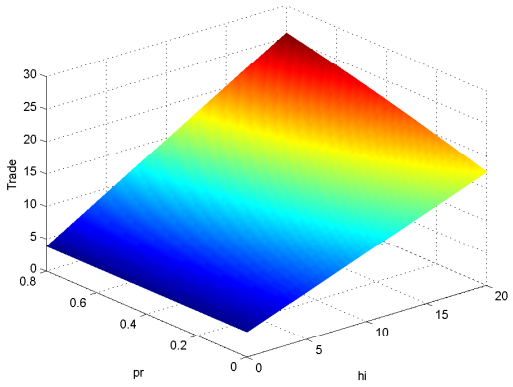


Figure 5. Trade-off metric as a function of h_I **and** p_r

h_I and p_r. Interestingly, when the mean time in initial state h_I is short, the system cost increase as the rekey probability p_r increase. However, we see a decrease in system cost as p_r increase, when h_I is very long. Also we can see, the system cost is more sensitive to h_I than to p_r. In Fig. 4, we conduct sensitivity analysis to the system confidentiality measure $Confid$. It increase dramatically with the sojourn time h_I in state I when the system is rekeying more frequently. However, it does not interact that strikingly with model input parameter p_r.

The trade-off metric as a function of h_I and p_r is depicted in Fig. 5. As expected, the trade-off metric monotonically increase as p_r and h_I increase. That is because the system more often rekeys and it spends more time in good state.

5 Conclusion and future work

In this report, we have presented an approach for quantitative assessment of security attributes for an offloading system under the specific threat of timing attacks. We have solved for steady-state probabilities of the Semi-Markov Process model as the foundation of security attributes analysis. Also, the model analysis is illustrated in a numerical example.

The objective of our future work is conducting more experiments to get the precise input parameters for our model. Furthermore, transient measure of our model will be conducted.

References

[1] D. Coppersmith. Small solutions to polynomial equations, and low exponent rsa vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997.

[2] A. N. Khan, M. M. Kiah, S. U. Khan, and S. A. Madani. Towards secure mobile cloud computing: A survey. *Future Generation Computer Systems*, 29(5):1278–1299, 2013.

[3] N. Limnios and G. Oprisan. *Semi-Markov processes and reliability*. Springer Science & Business Media, 2001.

[4] C. Rebeiro, D. Mukhopadhyay, and S. Bhattacharya. An introduction to timing attacks. In *Timing Channels in Cryptography*, pages 1–11. Springer, 2015.

[5] S. Subashini and V. Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11, 2011.

[6] K. S. Trivedi. *Probability & statistics with reliability, queuing and computer science applications*. John Wiley & Sons, 2008.

Towards Process Mining on Big Data: Optimizing Process Model Matching Approaches on High Performance Computing Infrastructure

Sharam Dadashnia, Tim Niesen, Peter Fettke, Peter Loos
Institute for Information Systems (IWi) at the
German Research Center for Artifical Intelligence (DFKI) Campus D3 2, 66123 Saarbrücken
{sharam.dadashnia | tim.niesen | peter.fettke | peter.loos}@iwi.dfki.de

Abstract

The project "Process Mining on Big Data" illustrates the potential of high performance computing infrastructures to face problems in the context of process mining, especially process model matching. Since a large number of input data exponentially affects the determination of correspondences between process models, processing huge model sets leads to an explosion in complexity and, thus, cannot be performed on standard machines. An iterative architectural prototyping research approach is used to calculate different parameter configurations for a newly developed matching algorithm in order to determine an optimal configuration. Optimal parameters were determined according to a defined quality criteria using different process model collections and used to further improve the matching approach.

1 Introduction

The project *Process Mining on Big Data* aims at investigating the potential of high performance computing (HPC) architectures for the field of process mining and related problem domains. Due to an increased availability of process log data, algorithms to extract ("discover") process models based on real usage scenarios gain in importance. This results in a growing number of inductively-created business process models that have to be managed in an efficient way. Against this background, one of the most prominent problems is the identification of similar or equivalent process models in a collection [1].

An important challenge in that regard is the determination of correspondences in the form of common substructures or shared elements between pairs of models [2]. Since a large number of input data exponentially affects correspondence determination, processing huge sets of process models leads to an explosion in complexity and, thus, cannot be performed on standard machines. In this project, we focused on a specific sub-problem related to the problem of correspondence determination, which is known as process model matching. In particular, the following research question has been investigated:

> *RQ: Can HPC provide adequate means to improve existing matching algorithms and to enhance state-of-the-art process mining?*

Against this background, the remainder of this report is structured as follows: Section 2 describes the research approach that our findings are based on. Next, section 3 shortly elaborates on the matching approach that was adapted to the HPC platform before section 4 reports on the optimization scenario. Section 5 then discusses the results as well as their implications. Finally, section 6 concludes the report and gives an outlook on possible follow-up projects.

2 Research Approach

The research described in the present report is based on the concept of architectural prototyping (AP) from software architecture development. An architectural prototype in that regard represents a means to learn and communicate different styles, features and patterns of a system under development and, hence, helps to explore and evaluate the best alternative in the software architecture development process [3].

The main objective of the approach described hereafter relates to an optimization problem regarding parameter configurations in the context of business process model matching. The problem is faced in an iterative manner: by using a repetitive cycle containing a feedback loop, parameters are incrementally refined towards a solution that performs best on-average as described in section 4. Figure 1 visualizes the employed four-step research approach with multiple iterations in phases two and three, which are executed on the IT basis infrastructure provided by the HPI Future SOC Lab consisting of a dedicated Ubuntu Linux-based server blade (24 cores, 64 GB of RAM). The approach is implemented as a first prototype in Java, running on a multi-threaded Java Virtual Machine (JVM) to demonstrate the applicability of the matching concept.

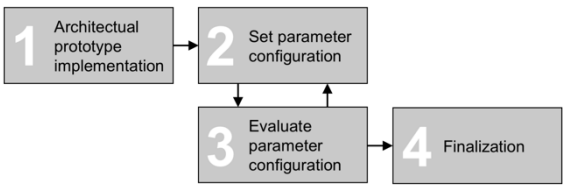


Figure 1: AP research approach

3 Vector Based Matching

In order to introduce the application setting for the following description of our HPC-based evaluation scenario, we briefly present the underlying matching algorithm. It was first published in [4] and describes a multi-phase approach to business process model

matching, which adopts the vector space model from Information Retrieval and combines it with additional techniques from Natural Language Processing.

As briefly mentioned before, process model matching aims at identifying correspondences between the nodes of models, for instance *identical* or *similar* model elements. Our approach focuses on semantic similarities within element description texts ("labels") and is based on the four-phase procedure sketched in figure 1. The phases are processed subsequently, i. e. a subordinate phase is only reached if no satisfying result could be generated in a preceding phase.

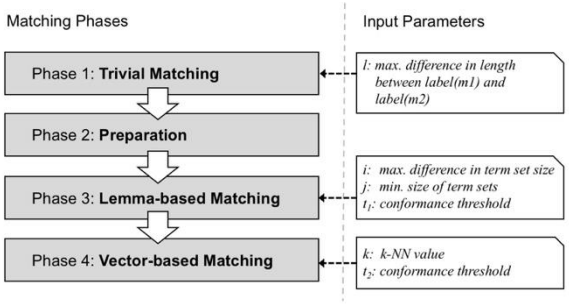


Figure 2: Overview of the multi-phase matching procedure

Phase 1: Trivial Matching. Following the assumption that identical labels and one label being a substring of another almost certainly indicate a correspondence, a *trivial* matching is performed in the first place. Parameter *l* denotes the allowed difference in the labels' length and can be adjusted to optimize matching performance.

Phase 2: Preparation. In order to execute phase 3 and 4, further preprocessing of label texts is necessary. For instance, lemmatization of words and stopword removal is performed in the preparation phase.

Phase 3: Lemma-based Matching. By considering each label as a set of words while abstracting from specific order and grammatical aspects (e. g. inflection), correspondences between "mostly identical" labels are identified. In particular, the intersection between the two sets must be greater than a threshold parameter t_1 and each label set must have a minimum size (parameter *j*). Furthermore, the set sizes must not vary by a value greater *i*.

Phase 4: Vector-based detail Matching. The final matching phase is based on a vector space approach, where process models as well as model labels are represented as individual vectors. By using k-NN clustering (parameter *k*), subcorpora are constructed, which are in turn used to accurately estimate the importance of label terms and, hence, to construct the vectors. To calculate the degree of similarity between two vectors, standard vector similarity measures like cosine similarity are computed and compared against a threshold (parameter t_2): similarity scores above the threshold indicate corresponding labels.

4 Evaluation

4.1 Evaluation Parameters

Referring to the matching procedure described in section 3, we have a total of six parameters that influence the actual matching performance. By using different sets of process model to be matched along with a set of corresponding reference matches, an optimal parameter configuration can be determined to optimize the average matching performance with respect to the available data. Therefore, possible combinations of parameter values have to be investigated, which depicts a complex combinational problem. Table 2 gives an overview of the parameters with the respective sets of possible values.

parameter	value set *m*	$\|m\|$
l	{1,2,3,4,5,6,7}	7
t_1	{0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8}	8
i	{1,2,3,4,5,6}	6
j	{5,6,7,8,9,10}	6
k	{1,2,3,4,5}	5
t_2	{0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8}	8

Table 1: Parameter values and combinations

Calculating each possible value combination of the six parameters results in a total of 7*8*6*6*5*8 = 80.640 walkthroughs for the whole matching procedure. As a single walkthrough took approximately 25 seconds on the employed workstation, the overall runtime can roughly be estimated to more than 23 days, which was not feasible in the scenario.

4.2 Evaluation Scenario

From the set of 80.640 possible parameter configurations, the evaluation scenario aims at finding the one configuration that performs best in the task of matching process models from a given collection. Performance is therefore measured in terms of the so-called *F-measure* value, which has been established in the field of process model matching [5].

In order to calculate the respective measure, a "reference matching" for a given process model collection must be available. To account for that issue, the described evaluation scenario draws on the setup of the second edition of the Process Model Matching Contest (PMMC) [4]. In the context of the PMMC, three different model collections ("datasets") have been provided along with a set of reference matchings between combinations of models ("goldstandard").

The results presented in section 5 are based on a detailed examination of dataset 1 and 2 from PMMC: the results produced by our vector-based matching algorithm are checked against the provided goldstandards while testing different parameter configurations. The configuration which yields the highest F-measure across all datasets is denoted the *optimal* or *best* configuration.

5 Results

Building on the evaluation scenario described in section 4, the following paragraphs present the results of the conducted analyses.

In terms of processing time and resource requirements, the usage of the HPC infrastructure provided by the HPI lead to a tremendous improvement. The overall processing time, which had been estimated to at least 23 days using a standard machine to test possible parameter configurations, could be reduced to less than one day. Interestingly, this improvement could already be realized by running the original implementation without further changes to the source code. The only alteration in comparison to the original run configuration relates to the settings of the respective Java virtual machine, which have been slightly adapted to better suit the characteristics of multi-core environments. While the achieved performance benefit is already high it still reveals the potential of a dedicated implementation that fully exhausts the possibilities of multi-threading optimizations.

Regarding the evaluation results, testing every parameter configuration out of the set of 80.640 on all datasets was key to determine interdependencies between the parameters and to find a final configuration that performs best on average across all datasets regarding F-measure. Generally speaking, some parameters tend to have a stronger influence on matching performance than others. For instance, the parameters t_1 and t_2 strongly influence matching performance especially for dataset 1 and 2, whereas the effect was not that large for dataset 3, while it was still significant.

Table 2 shows an exemplary excerpt from the calculation results focusing on a specific evaluation scenario where parameters t_1 and t_2 were varied while other parameters have been fixated. As can be seen from the data, F-measure values vary from a minimum of 0.21 (t_1 = 0.1, t_2 = 0.4-0.8) to a maximum of 0.41 (t_1 = 0.7-0.8, t_2 = 0.8).

Figure 3 visualizes the results in an area graph diagram, emphasizing the dependency between the two variables: while greater values for only one parameter do not yield a considerable gain in F-measure performance, a common increase of both parameters at the same time clearly leads to higher performance.

t_2\t_1	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**
0.1	0.22	0.25	0.29	0.29	0.30	0.30	0.30	0.30
0.2	0.22	0.25	0.30	0.31	0.32	0.32	0.32	0.32
0.3	0.22	0.24	0.30	0.31	0.32	0.32	0.32	0.32
0.4	0.21	0.24	0.30	0.30	0.32	0.32	0.32	0.32
0.5	0.21	0.24	0.31	0.31	0.33	0.33	0.33	0.33
0.6	0.21	0.24	0.32	0.33	0.35	0.35	0.35	0.35
0.7	0.21	0.24	0.33	0.35	0.38	0.38	0.38	0.38
0.8	0.21	0.24	0.34	0.37	0.40	0.40	**0.41**	**0.41**

Table 2: F-measures in dataset 1 dependent on parameters t_1 and t_2

6 Conclusion and Outlook

The project *Process Mining on Big Data* further extends fundamental work that has been conducted in a preceding project as a first step towards addressing new requirements regarding the application of high performance computing in the context of business process management [6]. It addresses a specific sub-problem from the field of process mining, namely business process model matching. Within the project, a novel,

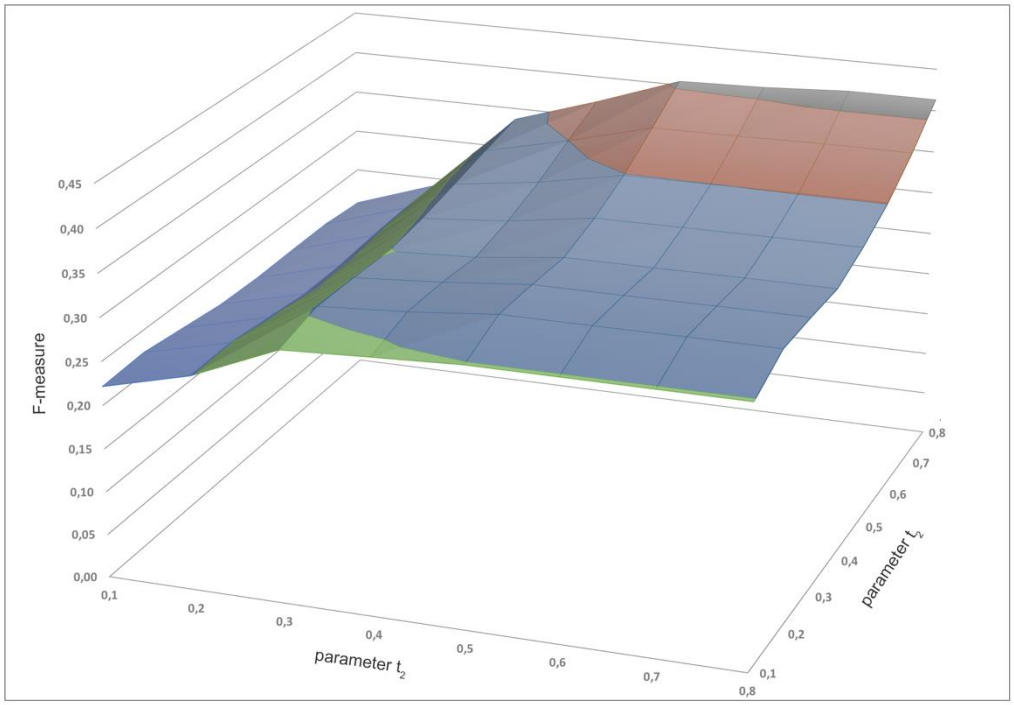


Figure 3: Visualization of parameter correlation in dataset 1

innovative matching algorithm has been evaluated in terms of different parameter configurations to determine an optimal setting. The usage of a high performance computing infrastructure enabled this to be done in a reasonable amount of time, thus, allowing adequate adjustments of implementation details. Yet, it also revealed the potential for further optimizations of the implementation regarding specific adaptions to multi-core environments. Finally, this enabled an iterative approximation of parameter configurations which is also in line with the employed research approach of architectural prototyping. As a next step, the evaluation results will be used to extend the existing matching procedure towards a dataset-adaptive learning approach.

Acknowledgement

The provided high performance IT infrastructure from the HPI allowed the investigation of concrete problem fields in information systems research. The authors thank the HPI Future SOC Lab for the chance of using these resources and appreciate a continuation of the project.

The basic concepts were developed in context of the project "Konzeptionelle, methodische und technische Grundlagen zur induktiven Erstellung von Referenzmodellen (Reference Model Mining)", which is funded by the Deutsche Forschungsgemeinschaft DFG (GZ LO 752/5-1).

7 References

[1] Thaler, T.; Dadashnia, S.; Sonntag, S.; Fettke, P.; Loos, P.: The IWi Process Model Corpus. In: Publications of the Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI), IWi-Heft, Vol. 199, 10/2015.

[2] Weidlich M., Mendling J.: Perceived consistency between process models. Information Systems 37 (2012) 2, S. 80–98.

[3] Bardram, J. E.; Christensen, H. B.; Hansen, K. M.: Architectural prototyping: An approach for grounding architectural design and learning. In: Software Architecture, 2004. WICSA 2004. Proceedings. Fourth Working IEEE/IFIP Conference on (pp. 15-24). IEEE.

[4] Antunes, G.; Bakhshandelh, M.; Borbinha, J.; Cardoso, J.; Dadashnia, S.; Chiara, F.; Gragoni, M.; Fettke, P.; Gal, A.; Ghidini, C.; Hake, P.; Khiat, A.; Klinkmüller, C.; Kuss, E.; Leopold, H.; Loos, P.; Meilicke, C.; Niesen, T.; Pesquita, C.; Peus, T.; Schoknecht, A.; Sheetrit, E.; Sonntag, A.; Stickenschmidt, H.; Thaler, T.; Weber, I.; Weidlich, M.: The Process Model Matching Contest 2015. In: Kolb J, Leopold H, Mendling J (eds) Proceedings of the 6th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA-15). pp 1–22.

[5] Cayoglu, U.; Dijkman, R.; Dumas, M.; Fettke, P.; Garca-Banuelos, L.; Hake, P.; Klinkmüller, C.; Leopold, H.; Ludwig, A.; Loos, P.; Mendling, J.; Oberweis, A.; Schoknecht, A.; Sheetrit, E.; Thaler, T.; Ullrich, M.; Weber, I.; Weidlich, M.: The Process Model Matching Contest 2013. In: 4th International Workshop on Process Model Collections: Management and Reuse, PMC-MR. Springer 2013.

[6] Thaler, T.; Dadashnia, S.; Fettke, P.; Loos, P.: Multi-Facet BPM: Identification, Analysis and Resolution of Resource-Intensive BPM Application. In: Proceedings of the Fall 2014 Future SOC Lab Day. HPI Future SOC Lab, October 29, Potsdam, Germany, Hasso-Plattner-Institut, 2014.

A survey of security-aware approaches for cloud-based storage and processing technologies

Max Plauth, Felix Eberhardt, Frank Feinbube and Andreas Polze
Operating Systems and Middleware Group
Hasso Plattner Institute for Software Systems Engineering
University of Potsdam
Potsdam, Germany
{max.plauth, felix.eberhardt, frank.feinbube, andreas.polze}@hpi.de

***Abstract*—In the Gartner hype cycle, cloud computing is a paradigm that has crossed the peak of inflated expectations but also has overcome the worst part of the trough of disillusionment. While the advantages of cloud computing are the best qualification for traversing the slope enlightenment, security concerns are still a major hindrance that prevent full adoption of cloud services across all conceivable user groups and use cases. With the goal of building a solid foundation for future research efforts, this paper provides a body of knowledge about a choice of upcoming research opportunities that focus on different strategies for improving the security level of cloud-based storage and processing technologies.**

I. Introduction

Providing low total cost of ownership, high degrees of scalability and ubiquitous access, cloud computing offers a compelling list of favorable features to both businesses and consumers. At the same time, these positive qualities also come with the less favorable drawback, that guaranteeing data confidentiality in cloud-based storage and processing services still remains an insufficiently tackled problem. As a consequence, many companies and public institutions are still refraining from moving storage or processing tasks into the domain of cloud computing. While this reluctance might be appropriate for few, highly sensitive use-cases, it poses the risk of an economic disadvantage in many other scenarios.

This paper provides an overview about the current state of the art in security-aware approaches for cloud-based storage and processing technologies. Since there are numerous ways to approach the topic, a large variety of potential starting points is presented. The goal is to provide a solid body of knowledge, which will be used as a foundation upon which novel security mechanisms can be identified and studied in the future. In the ensuing section, we present a selected list of preceding contributions to the field of security research in the context of cloud computing. Afterwards, a comprehensive review of the state of the art is provided to form a body of knowledge.

II. Preceding contributions

In the last couple of years, several aspects relevant to security-aware approaches for cloud-based storage and processing technologies have been researched at our research group. Among these aspects are technologies such as threshold cryptograph, trust-based access control, virtual machine introspection and searchable encryption. Since our ongoing research efforts build up on top of the insights gained in these preceding contributions, a brief overview is provided.

A. Threshold Cryptography

In a widely distributed environment, traditional authorization services represent a single-point of failure: If the service is unavailable, the encrypted data cannot be accessed by any party. In distributed setups, simple replication mechanisms can be considered a security threat, since attackers can gain full control as soon as a single node has been compromised. In order to eliminate this weakness, the general approach presented by Neuhaus et al. (2012) [1] employs the concept of Fragmentation-Redundancy-Scattering [2]: Confidential information is broken up into insignificant pieces which can be distributed over several network nodes.

The contribution of Neuhaus et al. (2012) [1] is the design of a distributed authorization service. A system architecture has been presented that enables fine-grained access control on data stored in a distributed system. In order to maintain privacy in the presence of compromised parties, a threshold encryption scheme has been applied in order to limit the power of a single authorization service instance.

B. Trust-Based Access Control

The Operating System and Middleware Group operates a web platform called *InstantLab* [3], [4]. The purpose of the platform is to provide operating system experiments for student exercises in the undergraduate curriculum. Virtualization technology is used to provide pre-packaged experiments, which can be conducted through a terminal session in the browser. Thus far, massive open online-courses (MOOCs) have not been well suited for hands-on experiments, since assignments have been non-interactive. The main goal of *InstantLab* is to provide more interactive assignments and enable iterative test-and-improve software development cycles as well as observational assignments.

Providing a platform that enables a large audience to perform live software experiments creates several challenges regarding the security of such a platform. Malicious users might abuse resources for other means than the intended software experiments. In order to detect misuse of the provided

resources, virtual machine introspection is applied. Furthermore, *InstantLab* [3], [4] demonstrates how automatic resource management is enabled by trust-based access control schemes. The purpose of trust-based access control is to restrict user access to resource intensive experiments. The approach implemented in *InstantLab* [3], [4] calculates a user's trust level based on his/her previous behavior.

C. Virtual Machine Introspection

In the age of cloud computing and virtualization, virtual machine introspection provides the means to inspect the state of virtual machines through a hypervisor without the risk of contaminating its state. Inspection capabilities are useful for a wide range of use case scenarios, ranging from forensics to more harmless cases such as making sure a tenant is not violating against the terms of use of the provider.

The work of Westphal et al. (2014) [5] contributes to the field of virtual machine introspection by providing a monitoring language called VMI-PL. Using this language, users can specify which information should be obtained from a virtual machine. Unlike competing approaches like libVMI [6] and VProbes [7], VMI-PL does not limit users to hardware level metrics, but it also provides operating system level information such as running processes and other operating system events. Furthermore, the language can also be used to monitor data streams such as network traffic or user interaction.

D. Searchable Encryption

For many use cases, efficient and secure data sharing mechanisms are crucial, especially in distributed scenarios where multiple parties have to access the same data repositories from arbitrary locations. In such scenarios, the scalability of cloud computing makes resources simple to provision and to extend. However, when it comes to storing sensitive data in cloud-hosted data repositories, data confidentiality is still a major issue that discourages the use of cloud resources in sensitive scenarios. While traditional encryption can be used to protect the privacy of data, it also limits the set of operations that can be performed efficiently on encrypted data, such as search. Encryption schemes which allow the execution of arbitrary operations on encrypted data are still utopian. However, searchable encryption schemes exist that enable keyword-based search without the disclosure of keywords.

Neuhaus et al. (2015) [8] studied the practical applicability of searchable encryption for data archives in the cloud. For their evaluation, an implementation of Goh's searchable encryption scheme [9] was embedded into the document-based database MongoDB. With the encryption scheme in place, benchmarks revealed that the overhead for insertions is negligible compared to an unencrypted mode of operation. Search queries on the other hand come with a considerable overhead, since Goh's scheme [9] mandates a linear dependency between the complexity of search operations and the number of documents. However, the processing time of encrypted queries should be in acceptable orders of magnitude for interactive use cases where the increased security is mandatory.

III. State of the Art

In the context of cloud computing, the field of work related to security-aware approaches for storage and processing technologies comprises a wide range of diverse directions. In the consequent part of this document, the state of the art is presented for a selection of differentiated topics. First, projects are highlighted which provide best practices for increasing security. Afterwards, new trends in virtualization strategies are outlined, followed by a brief introduction to novel hardware security mechanisms. Finally, the security aspects of providing coprocessor resources in virtual machines is illustrated.

A. New trends in virtualization strategies

Virtualization still remains as one of the main technological pillars of cloud computing. The main reason for this key role is that it enables high degrees of resource utilization and flexibility. Today, the most common approach for virtualization resorts to low-level hypervisors like *Xen* or *KVM* that employ hardware assisted virtualization in order to run regular guest operating systems in a para-virtualized or fully virtualized fashion. Recently however, new virtualization approaches have gained momentum. While containerization approaches move the scope of virtualization to higher levels of the application stack, unikernels are working at the same level of abstraction as regular operating systems but at the same time change the operating system drastically. A comparison of the different approaches is illustrated in Figure 1.

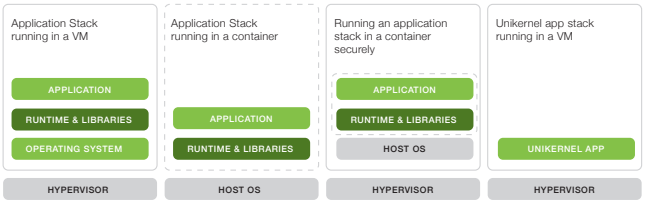


Fig. 1: The virtual machine stack as well as both containerization approaches come with a significant amount of overhead. Unikernels aim at reducing the footprint of virtualized applications. Source: [10]

1) Containers: In contrast to hypervisor-level virtualization approaches, where an entire operating system instance is virtualized, containers belong to the class of operating-system-level virtualization strategies that utilize multiple user-space instances in order to isolate tenants. The main goal of popular containerization implementations like Linux Containers and Docker is to reduce the memory footprint of hosted applications and to get rid of the overhead inherent to hypervisor-based virtualization. Recent studies demonstrate that the concept of containerization is able to outperform matured hypervisors in many use cases [11]–[13]. Regarding security aspects, most containerization approaches thus far rely on the operating system kernel to provide sufficient means of isolation between different containers. However, the LXD project aims at providing hardware-based security features to

containers in order to provide isolation levels on par with hypervisor-level based virtualization.

2) Unikernels: Unikernels are a new approach to hypervisor-level virtualization. The core concept of unikernels is based on the idea of deploying applications by merging application code and a minimal operating system kernel into a single immutable virtual machine image that is run on top of a standard hypervisor [14]–[16]. Since unikernels intentionally do not support the concept of process isolation, no time-consuming context switches have to be performed. The general idea behind unikernel systems is not entirely new, as it builds up on top of the concept of *library operating systems* such as exokernel [17] or Nemesis [18]. The main difference to library operating systems is that unikernels only run on hypervisors and do not support bare metal deployments, whereas library operating systems are targeting physical hardware. Due to the necessity to support physical hardware, library operating systems struggled with compatibility issues and proper resource isolation among applications. Unikernels are solving these problems by using a hypervisor in order to abstract from physical hardware and to provide strict resource isolation between applications [15]. Currently, the two most popular unikernel implementations are OSv [19] and Mirage OS [20].

According to Madhavapeddy et al. [15], unikernels are able to outperform regular operating systems in the following disciplines:

a) Boot time: Unikernel systems are single purpose systems, meaning that they run only one application. Unnecessary overhead is stripped of by only linking libraries into a unikernel image which are required by the application. As a result, very fast boot times can be achieved. In their latest project *Jitsu: Just-In-Time Summoning of Unikernels* [21], Madhavapeddy et al. managed to achieve boot times in the order of 350ms on ARM CPUs and 30ms on x86 CPUs, which enables the possibility of dynamically bringing up virtual machines in response to network traffic.

b) Image size: Since a unikernel system only contains the application and only the required functionality of the specialized operating system kernel, unikernel images are much smaller compared to traditional operating system images. The smaller binaries simplify management tasks like live-migration of running virtual machine instances.

c) Security: By eliminating functionality which is not needed for the execution of an application inside a unikernel image, the attack surface of the system is reduced massively. Furthermore, the specialized operating system kernel of a unikernel image is usually written in the same high-level language as the application. The resulting absence of technology borders facilitates additional opportunities for code checking like static type checking and automated code checking. However, even if an attacker should manage to inject malicious code into a unikernel instance, it can only cause limited harm since no other application runs within the same image.

B. Hardware-based security mechanisms

1) Trusted Execution Technology (TXT): The goal of Intel Trusted Execution Technology (TXT) [22] technology is that the user can verify if the operating system or its configuration was altered after the boot up. This requires a trusted platform module (TPM) which stores system indicators securely. The approach TXT is using is called dynamic root of trust measurement. For this methodology, the system can be brought into a clean state (SENTER instruction) after the firmware was loaded. In this approach as mentioned earlier only the operating system level software gets measured. These measurements can be compared with the original files / properties of the OS that have to be known beforehand.

2) Software Guard Extensions (SGX): Sensitive tasks have to face an abundance of potential threats on both the software and the hardware level. On the hardware level, sensitive information such as encryption keys can be extracted from the systems main memory using DMA attacks or cold boot attacks. On the software level, the worst case has to be assumed and even the operating system has to be considered as a potential threat. While the concept of processes implements a high level of isolation between different applications, the elevated privileges of an operating system allow it to tamper with any process. These capabilities always pose a security threat, not just in the obvious case where the operating system might not be fully trusted. Even with a trusted operating system, there is always a certain risk that malicious code running in a separate process might gain elevated privileges. As soon as that happens, a malicious application can tamper with any process running on the system.

As a countermeasure to these threats, the Intel Software Guard Extensions (SGX) [23] introduced secure enclaves, which allow the safe execution of sensitive tasks even in untrustworthy environments. Enclaves are protected memory areas, which are encrypted and entirely isolated (see Figure 2). Even privileged code is not able to access the contents of an enclave. One process can even use multiple enclaves, which allows a high degree of flexibility. SGX does not require a trusted platform module (TPM), as the entire feature is implemented on the CPU. This level of integration reduces the list of trusted vendors to the CPU manufacturer and thus minimizes the number of potential attack vectors.

C. Virtualization of coprocessors resources

Coprocessors such as *Graphics Processing Units* (GPUs), *Field-Programmable Gate Arrays* (FPGAs) or Intel's *Many Integrated Core* (MIC) devices have become essential components in the *High Performance Computing* (HPC) field. Regarding the domain of cloud computing however, the utilization of coprocessors is not that well established. While there has been little demand for HPC-like applications on cloud resources in the past, the demand for running scientific applications on cloud computing infrastructure has increased [25]. Furthermore, moving compute-intensive applications to the cloud is becoming increasingly feasible [26] when it comes to CPU-based tasks. While several providers already offer

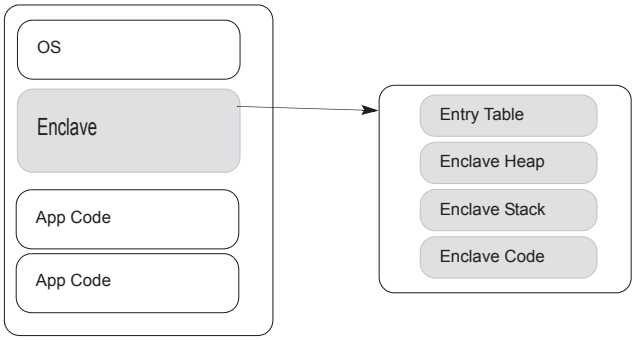


Fig. 2: Secure enclaves provide an encrypted address space that is protected even from operating system access. Source: [24]

cloud resources with integrated GPUs, their implementation is based on pass-through of native hardware. Assigning dedicated devices to each virtual machine results in high operational costs and decreased levels of flexibility. In such setups, virtual machines can neither be suspended nor migrated. Furthermore, the one-to-one mapping between pass-through devices and virtual machines prevent efficient utilization of coprocessors, which has a negative impact on cost effectiveness due to the high energy consumption of such hardware. Although projects exist that maximize resource utilization by providing unused GPU resources to other compute nodes [27] or that save energy by shutting down inactive compute nodes [28], a large gap between the capabilities of GPU and CPU virtualization still exists.

In the ensuing paragraphs, the state of the art of coprocessor virtualization is evaluated based on several characteristics. Most work deals with GPU compute devices, however the general techniques are applicable to other coprocessor classes as well. For desktop-based GPU virtualization, Dowty and Sugerman [29] define four characteristics that are to be considered: performance, fidelity, multiplexing and interposition. Regarding cloud-based virtualization, the aforementioned enumeration is missing isolation as an important characteristic. In order to establish coprocessors in cloud computing, one of the most crucial characteristics is that multiple tenants have to be properly isolated. Since the focus is set on coprocessors and thus compute-based capabilities instead of interactive graphics, fidelity can mostly be ignored for our use case. Last but not least, performance should not suffer severely from the virtualization overhead. However, without isolation, multiplexing and interposition capabilities, performance is worthless in the cloud computing use case.

1) Isolation: Thus far, isolation is only addressed by approaches that make use of mediated pass-through strategies like Intel GVT-g (formerly called gVirt) [30] and NVIDIA GRID. While the latter is a commercial closed-source implementation, the implementation details of GVT-g are publicly available as an open source project. In Intel's approach, each virtual machine runs the native graphics driver. In contrast to regular pass-through, mediated pass-through uses a trap-and-emulate mechanism is used to isolate virtual machine instances from each other. The main drawback is that the implementation of the mediated pass-through strategy has to be tailored to the specifications of each supported GPU, which again requires detailed knowledge about the GPU design. Overall, this approach is only feasible for the manufacturers of GPUs themselves.

With rCUDA [31]–[33], vCUDA [34], gVirtuS [35], GViM [36], VirtualCL [37] and VOCL [38], many approaches exist which are based on call forwarding. Originating from the field of High Performance Computing, the call forwarding approach uses a driver stub in the guest operating system which redirects the calls to a native device driver in the privileged domain. Since isolation is barely an issue in the HPC domain, none of the existing approaches implement isolation mechanisms.

2) Multiplexing: Sharing a single GPU among multiple virtual machines is possible for all aforementioned implementation strategies. In the faction of mediated pass-through implementations, both Intel GVT-g and NVIDIA GRID support multiplexing in order to serve multiple virtual machines with a single GPU. As for isolation, a trap-and-emulate mechanism in the hypervisor coordinates devices accesses from multiple virtual machines. On the side of call forwarding approaches, the implementation of multiplexing capabilities with low overhead is very a tough challenge. In the privileged domain, additional logic has to be implemented that schedules requests from different guests. So far, only vCUDA [34] provides such multiplexing mechanisms.

3) Interposition: While mediated pass-through approaches excel call forwarding strategies in both isolation and multiplexing, interposition is hard to achieve for mediated pass-through. Although an implementation is possible in theory [39], it is not feasible in practice as it is susceptible to the slightest variations on the hardware level. With vCUDA [34] and VOCL [38] on the other hand, multiple projects based on call forwarding exist that successfully implement interposition capabilities. Again, a piece of middleware is required in the hypervisor which carefully tracks the state of each virtual GPU instance. With such capabilities at hands, virtual machines can be suspended and even live-migrated to other virtual machine hosts.

D. Best practices for secure coding

Over the last years, several best practice collections and frameworks dealing with improving the security of information technology were established and maintained by companies and public authorities alike.

1) Critical Security Controls: The term "Security Fog of More" was established by Tony Sager, a chief technologist of the *Council on CyberSecurity*. He noticed that security professionals are confronted with a plethora of security products and services. These choices are influenced by compliance, regulations, frameworks and audits e.g. the "Security Fog of More". As a consequence, one of the main challenges today is making an educated choice. Sager wants to help security professionals by providing a framework for security choices

called *Critical Security Controls* [21] (see Figure 3) that spans 20 different areas of IT security containing suggestions for each of these areas with a focus on scalability of the solutions.

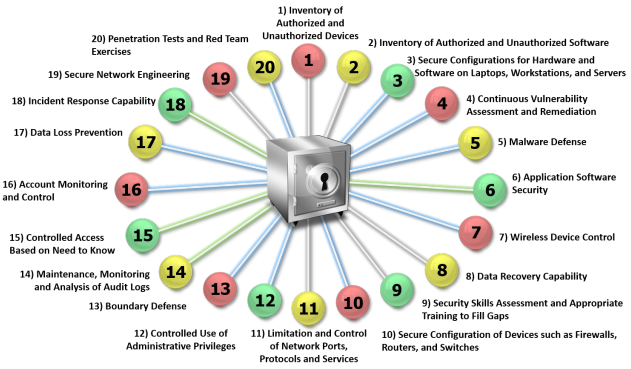


Fig. 3: The Critical Security Controls framework categorizes security threats in 20 classes. Source: [40]

2) Open Web Application Security Project: The Open Web Application Security Project (OWASP) is a non-profit organization founded 2004 with the goal of improving software security. The OWASP houses a wide range of security related projects centered around all aspects of software development driven by a large community of volunteers. We will provide a brief overview over a selection of the most popular OWASP projects:

a) OWASP Developer Guide: The OWASP Developer Guide was the first project pursued by OWASP. In its latest revision, the guide describes general concepts about developing secure software without a focus on specific technologies. The guide covers topics such as Architecture, Design, Build Process and Configuration of secure software and is targeting developers as its target audience. The instructions can be used as additional guidelines for penetration testers as well.

b) OWASP Testing Guide: The OWASP Testing Guide is a best practice collection for penetration testing of web applications and services. The guide covers the software development process as well as testing approaches for different parts of web applications (e.g. Authentication, Encryption or Input Validation).

c) OWASP Top 10: The OWASP Top 10 is a list maintained by security experts which contains the 10 most prevalent security flaws in web applications. The goal of this list to establish a security awareness in IT companies to prevent the occurrence of the most common vulnerabilities in their applications.

IV. Discussion

The state of the art presented in the previous section has demonstrated that a vast variety of approaches exist that can be accommodated under the headline *security-aware approaches for cloud-based storage and processing technologies*. While soft approaches such as best practice collections are beneficial for everyday use, they are of limited use for technically oriented research prototypes. On the other end of the scale are hardware security features such as TXT and SGX.

The Software Guard Extensions is an interesting new feature that can be used to evaluate problems that require the execution of crucial code in untrustworthy requirements. However, it should be noted that until the day of writing, no commercially available processor implements the SGX feature. Moreover, it is even unclear when such a processor can be expected to become available. Under the bottom line, it seems like SGX provides various research opportunities, however the focus for near future projects should be shifted to different topics.

With virtualization being a key technology in cloud computing, it is important to keep an eye on new virtualization concepts. With the advent of containerization, a new approach to virtualization has surfaced that tries to minimize the performance overhead caused by an additional level of context switches. While containers have already achieved a certain prevalence rate, unikernels are a recent re-discovery of an old concept. Unikernels should be considered as a direct competition to containers, since they also address mitigation of virtualization overhead while maintaining a thorough level of isolation. Even though there is a certain risk that unikernels might be a fashionable trend, eventual benefits over traditional virtual machines and containerization approaches should be evaluated. With boot times of tens of milliseconds, the use of unikernels might enable new degrees of dynamic resource utilization and improved power management.

In the subject area of virtualization, server-based virtualization of coprocessor resources, most importantly *Graphics Processing Units* (GPUs), is another aspect that has been neglected in the past. High operational costs are caused by poorly utilized devices. Even though some approaches exist that allow resource multiplexing, the near absence of proper isolation has been a deal-breaker for the cloud computing scenario thus far.

V. Outlook

Recalling the topics presented and discussed in Sections III and IV, many approaches exist for providing increased levels of security in the use case of cloud computing. While best practices collections may be beneficial for everyday use, they are of limited use for technically oriented research interests. It seems as if technological improvements like the Software Guarded Extensions (SGX) are an interesting target for further research efforts. However, the uncertain date of availability of the technology enforces a postponed examination of the topic. Regarding new virtualization approaches, there is a certain risk that unikernels are a fashionable trend that might disappear rather sooner than later. However, the crucial role of virtualization in cloud computing suggests that unikernels and containers should be evaluated more thoroughly. From a functional perspective, these new virtualization approaches do have the potential to improve both security aspects as well as performance. Regarding the non-function side of the topic, unikernels might enable us to improve both dynamic resource utilization and power management strategies. Last but not

least, employing coprocessor resources in cloud computing is a topic that requires extensive research efforts. In order to move from dedicated devices to truly shared resources, security is a major concern that has not been solved yet. Lightweight isolation mechanisms have to be researched that provide tight levels of isolation while inducing bearable levels of overhead compared to native hardware.

REFERENCES

[1] C. Neuhaus, M. von Löwis, and A. Polze, "A dependable and secure authorisation service in the cloud," in *CLOSER*, 2012, pp. 568–573.

[2] J.-C. Fabre, Y. Deswarte, and B. Randell, *Designing secure and reliable applications using fragmentation-redundancy-scattering: an object-oriented approach*. Springer, 1994.

[3] C. Neuhaus, F. Feinbube, A. Polze, and A. Retik, "Scaling software experiments to the thousands," in *CSEDU 2014 - Proceedings of the 6th International Conference on Computer Supported Education, Volume 1, Barcelona, Spain, 1-3 April, 2014*, 2014, pp. 594–601.

[4] C. Neuhaus, F. Feinbube, and A. Polze, "A platform for interactive software experiments in massive open online courses," *Journal of Integrated Design and Process Science*, vol. 18, no. 1, pp. 69–87, 2014.

[5] F. Westphal, S. Axelsson, C. Neuhaus, and A. Polze, "VMI-PL: A monitoring language for virtual platforms using virtual machine introspection," *Digital Investigation*, vol. 11, no. S-2, pp. S85–S94, 2014.

[6] LibVMI Project, "LibVMI," http://libvmi.com, accessed: 2015-07-17.

[7] VMware, Inc., "VProbes Programming Reference," Tech. Rep., 2011.

[8] C. Neuhaus, F. Feinbube, D. Janusz, and A. Polze, "Secure Keyword Search over Data Archives in the Cloud: Performance and Security Aspects of Searchable Encryption," in *5th International Conference on Cloud Computing and Services Science, .* ACM, 2015.

[9] E.-J. Goh *et al.*, "Secure indexes." *IACR Cryptology ePrint Archive*, vol. 2003, p. 216, 2003.

[10] Xen Project, "The Next Generation Cloud: The Rise of the Unikernel," http://xenproject.org, Tech. Rep., 2015.

[11] S. Soltesz, H. Pötzl, M. E. Fiuczynski, A. Bavier, and L. Peterson, "Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors," in *ACM SIGOPS Operating Systems Review*, vol. 41, no. 3. ACM, 2007, pp. 275–287.

[12] M. G. Xavier, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange, and C. A. De Rose, "Performance evaluation of container-based virtualization for high performance computing environments," in *Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on*. IEEE, 2013, pp. 233–240.

[13] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and linux containers," Tech. Rep., 2014.

[14] A. Madhavapeddy and D. J. Scott, "Unikernels: Rise of the virtual library operating system," *Queue*, vol. 11, no. 11, p. 30, 2013.

[15] A. Madhavapeddy, R. Mortier, C. Rotsos, D. Scott, B. Singh, T. Gazagnaire, S. Smith, S. Hand, and J. Crowcroft, "Unikernels: Library operating systems for the cloud," in *ACM SIGPLAN Notices*, vol. 48, no. 4. ACM, 2013, pp. 461–472.

[16] D. Schatzberg, J. Cadden, O. Krieger, and J. Appavoo, "A way forward: enabling operating system innovation in the cloud," in *Proceedings of the 6th USENIX conference on Hot Topics in Cloud Computing*. USENIX Association, 2014, pp. 4–4.

[17] D. R. Engler, M. F. Kaashoek *et al.*, *Exokernel: An operating system architecture for application-level resource management*. ACM, 1995, vol. 29, no. 5.

[18] D. E. Porter, S. Boyd-Wickizer, J. Howell, R. Olinsky, and G. C. Hunt, "Rethinking the library os from the top down," *ACM SIGPLAN Notices*, vol. 46, no. 3, pp. 291–304, 2011.

[19] A. Kivity, D. Laor, G. Costa, P. Enberg, N. Har'El, D. Marti, and V. Zolotarov, "Osv—optimizing the operating system for virtual machines," in *2014 usenix annual technical conference (usenix atc 14)*, vol. 1. USENIX Association, 2014, pp. 61–72.

[20] A. Madhavapeddy, R. Mortier, R. Sohan, T. Gazagnaire, S. Hand, T. Deegan, D. McAuley, and J. Crowcroft, "Turning down the lamp: software specialisation for the cloud," in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, HotCloud*, vol. 10, 2010, pp. 11–11.

[21] Council on CyberSecurity, "The Critical Security Controls for Effective Cyber Defense Version 5.0," https://www.sans.org/, Tech. Rep., February 2014.

[22] Intel Corporation, "Intel trusted execution technology white paper," http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/trusted-execution-technology-security-paper.pdf, online, Accessed 31.07.2015.

[23] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative instructions and software model for isolated execution," in *Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy*, ser. HASP '13. New York, NY, USA: ACM, 2013, pp. 10:1–10:1.

[24] Intel Corporation, "Intel© Software Guard Extensions Programming Reference," Tech. Rep., Oct. 2014.

[25] S. Benedict, "Performance issues and performance analysis tools for hpc cloud applications: a survey," *Computing*, vol. 95, no. 2, pp. 89–108, 2013.

[26] K. Mantripragada, A. Binotto, L. Tizzei, and M. Netto, "A feasibility study of using hpc cloud environment for seismic exploration," in *77th EAGE Conference and Exhibition 2015*, 2015.

[27] P. Markthub, A. Nomura, and S. Matsuoka, "Using rCUDA to Reduce GPU Resource-assignment Fragmentation caused by Job Scheduler," in *15th International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2014.

[28] P. Lama, Y. Li, A. M. Aji, P. Balaji, J. Dinan, S. Xiao, Y. Zhang, W.-c. Feng, R. Thakur, and X. Zhou, "pVOCL: Power-Aware Dynamic Placement and Migration in Virtualized GPU Environments," in *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, Jul. 2013, pp. 145–154.

[29] M. Dowty and J. Sugerman, "Gpu virtualization on vmware's hosted i/o architecture," *ACM SIGOPS Operating Systems Review*, vol. 43, no. 3, pp. 73–82, 2009.

[30] K. Tian, Y. Dong, and D. Cowperthwaite, "A full GPU virtualization solution with mediated pass-through," in *Proc. USENIX ATC*, 2014.

[31] J. Duato, F. D. Igual, R. Mayo, A. J. Peña, E. S. Quintana-Ortí, and F. Silla, "An Efficient Implementation of GPU Virtualization in High Performance Clusters," in *Euro-Par 2009 Parallel Processing Workshops*, ser. Lecture Notes in Computer Science, H.-X. Lin, M. Alexander, M. Forsell, A. Knüpfer, R. Prodan, L. Sousa, and A. Streit, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 6043, pp. 385–394.

[32] J. Duato, A. J. Pena, F. Silla, R. Mayo, and E. S. Quintana-Orti, "rCUDA: Reducing the number of GPU-based accelerators in high performance clusters," in *2010 International Conference on High Performance Computing & Simulation*. IEEE, Jun. 2010, pp. 224–231.

[33] J. Duato, A. J. Pena, F. Silla, J. C. Fernandez, R. Mayo, and E. S. Quintana-Orti, "Enabling CUDA acceleration within virtual machines using rCUDA," in *2011 18th International Conference on High Performance Computing*. IEEE, Dec. 2011, pp. 1–10.

[34] L. Shi, H. Chen, J. Sun, and K. Li, "vCUDA: GPU-accelerated high-performance computing in virtual machines," *Computers, IEEE Transactions on*, vol. 61, no. 6, pp. 804–816, 2012.

[35] G. Giunta, R. Montella, G. Agrillo, and G. Coviello, "A GPGPU transparent virtualization component for high performance computing clouds," in *Euro-Par 2010-Parallel Processing*. Springer, 2010, pp. 379–391.

[36] V. Gupta, A. Gavrilovska, K. Schwan, H. Kharche, N. Tolia, V. Talwar, and P. Ranganathan, "GViM: GPU-accelerated Virtual Machines Vishakha," in *Proceedings of the 3rd ACM Workshop on System-level Virtualization for High Performance Computing - HPCVirt '09*. New York, New York, USA: ACM Press, Mar. 2009, pp. 17–24.

[37] A. Barak and A. Shiloh, "The VirtualCL (VCL) Cluster Platform."

[38] S. Xiao, P. Balaji, Q. Zhu, R. Thakur, S. Coghlan, H. Lin, G. Wen, J. Hong, and W.-c. Feng, "VOCL: An Optimized Environment for Transparent Virtualization of Graphics Processing Units," in *Proceedings of 1st Innovative Parallel Computing (InPar)*, 2012, pp. 1–12.

[39] E. Zhai, G. D. Cummings, and Y. Dong, "Live migration with pass-through device for Linux VM," in *OLS08: The 2008 Ottawa Linux Symposium*, 2008, pp. 261–268.

[40] F. T. Insider, "Continuous Diagnostics and Mitigation Addresses "Foundational" Issues Identified by SANS," *http://www.federaltechnologyinsider.com/cdm-addresses-foundational-issues-identified-sans/*, May 2014.

OntQA-Replica: Intelligent Data Replication for Ontology-Based Query Answering (Revisited and Verified)

Lena Wiese
Research Group Knowledge Engineering
Institut für Informatik
Georg-August-Universität Göttingen
Goldschmidtstraße 7
37077 Göttingen
wiese@cs.uni-goettingen.de

Abstract

We extend previous work in the OntQA-Replica project by validating the results in a larger database cluster (10 nodes) and a larger dataset (four times larger than previously). The project aims to improve the performance of ontology-based query answering in distributed databases by employing a preprocessing procedure (including a clustering step and a fragmentation step): for efficient query answering, data records that are semantically related are grouped in the same data fragment based on a notion of similarity in the ontology. At the same time, the OntQA-Replica project supports an intelligent data replication approach that minimizes the amount of resources used and enables an efficient recovery in case of server failures.

1 Introduction

In a cloud storage system, a distributed database management system (DDBMS; see [4]) can be used to manage the data in a network of servers. The decisive features are replication (for recovery and scalability purposes) and load balancing (data distribution according to the capacities of servers). Previous work in the project aimed at facilitating the partitioning of large data tables based on similarity of values in an ontology or taxonomy. Hence due to its preprocessing nature, it provides an efficient means to support a form of flexible query answering. Flexible query answering [1] offers mechanisms to intelligently answer user queries going beyond conventional exact query answering. If a database system is not able to find an exactly matching answer, the query is said to be a failing query. Conventional database systems usually return an empty answer to a failing query. In most cases, this is an undesirable situation for the user, because he has to revise his query and send the revised query to the database system in order to get some information from the database. In contrast, flexible query answering systems internally revise failing user queries themselves and – by evaluating the revised query – return answers to the user that are more informative for the user than just an empty answer. One way to obtain such informative answers is to use an ontology to return answers that are related to the original search term according to some notion of similarity. For example, in an electronic health record, when searching for the term cough, the terms bronchitis and asthma might be similar to cough and might be returned as related answers. Unfortunately, finding related answers at runtime by consulting the ontology for each query is highly inefficient.

2 Ontology-Driven Fragmentation

Our aim is to extend previous work [3] to become a scalable method for flexible query anwering. The basic idea is to apply a clustering procedure to partition the original tables into fragments based on a *relaxation attribute* chosen for anti-instantiation. Finding these fragments is achieved by grouping (that is, *clustering*) the values of the respective table column (corresponding to the relaxation attribute); for this a notion of similarity

Next, the table is split into fragments according to the clusters found such that the following formal definition applies.

Clustering-based fragmentation: Let A be a relaxation attribute; let F be a table instance (a set of tuples); let $C = \{c_1, \dots c_n\}$ be a complete clustering of the active domain $\pi_A(F)$ of A in F; let $head_i \in c_i$; then, a set of fragments $\{F_1, \dots, F_n\}$ (defined over the same attributes as F) is a *clustering-based* fragmentation if

- Horizontal fragmentation: for every fragment F_i, $F_i \subseteq F$
- Clustering: for every F_i there is a cluster $c_i \in C$

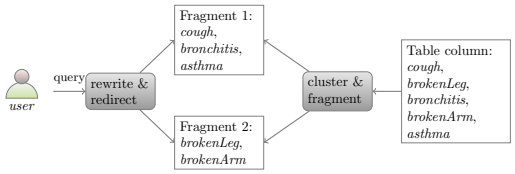


Figure 1: Ontology-driven fragmentation

such that $c_i = \pi_A(F_i)$ (that is, the active domain of F_i on A is equal to a cluster in C)

- Threshold: for every $a \in c_i$ (with $a \neq head_i$) it holds that $sim(a, head_i) \geq \alpha$
- Completeness: For every tuple t in F there is an F_i in which t is contained
- Reconstructability: $F = F_1 \cup \ldots \cup F_n$
- Non-redundancy: for any $i \neq j$, $F_i \cap F_j = \emptyset$ (or in other words $c_i \cap c_j = \emptyset$)

More formally, we apply the clustering approach described in [3] (or any other method to semantically split the attribute domain into subsets) on the relaxation attribute, so that each cluster inside one clustering is represented by a *head* term (also called prototype) and each term in a cluster has a similarity *sim* to the cluster head above a certain threshold α. We then obtain a clustering-based fragmentation for the original table F into fragments.

3 Evaluation with larger Dataset in larger Cluster

Our experimental evaluation – the OntQA-Replica system – was run on a distributed SAP HANA installation with 10 database server nodes (and hence on a larger cluster than in previous experiments) provided by the Future SOC Lab of Hasso Plattner Institute. All runtime measurements are taken as the median of several (at least 5) runs per experiment.
The example data set consists of a table (called *ill*) that resembles a medical health record and is based on the set of Medical Subject Headings (MeSH [2]). The table contains as columns an artificial, sequential *tupleid*, a random *patientid*, and a *disease* chosen from the MeSH data set as well as the *concept* identifier of the MeSH entry. We varied the table sizes during our test runs. The smallest table consists of 56,341 rows (one row for each MeSH term). We increased that table size by duplicating the original data up to 12 times (hence four times more than in previous runs), resulting in 230,772,736 rows. A clustering is executed on the MeSH data based on the concept identifier (which orders the MeSH terms in a tree); in other words, entries from the same subconcept belong to the same cluster. One fragmentation (the clustered fragmentation) was obtained from this clustering and consists of 117 fragments which are each stored in a smaller table called *ill-i* where i is the cluster ID. To allow for a comparison and a test of the recovery strategy, another fragmentation of the table was done using round robin resulting in a table called *ill-rr*; this distributes the data among the database servers in chunks of equal size without considering their semantic relationship; these fragments have an extra column called *clusterid*.
In order to manage the fragmentation, several metadata tables are maintained:

- A **root** table stores an ID for each cluster (column *clusterid*) as well as the cluster head (column *head*) and the name of the server that hosts the cluster (column *serverid*).
- A **lookup** table stores for each cluster ID (column *clusterid*) the tuple IDs (column *tupleid*) of those tuples that constitute the clustered fragment.
- A **similarities** table stores for each head term (column *head*) and each other term (column *term*) that occurs in the active domain of the corresponding relaxation attribute a similarity value between 0 and 1 (column *sim*).

The original query has to be rewritten in order to consider all the related terms as valid answers. We tested and compared three query rewriting procedures:

- lookup table: the first rewriting approach uses the lookup table to retrieve the tuple IDs of the corresponding rows and executes a JOIN on table *ill*.
- extra clusterid column: the next approach relies on the round robin table and retrieves all relevant tuples based on a selection predicate on the clusterid column.
- clustered fragmentation: the last rewriting approach replaces the occurrences of the *ill* table by the corresponding *ill-i* table for clusterid i.

3.1 Identifying the matching cluster

Flexible Query Answering intends to return those terms belonging to the same cluster as the query term as informative answers. Before being able to return the related terms, we hence have to identify the matching cluster: that is, the ID of the cluster the head of which has the *highest* similarity to the query term.
The relaxation term t is extracted from the query and then the top-1 entry of the similarities table is obtained when ordering the similarities in descending order:

```
SELECT TOP 1 root.clusterid
 FROM root, similarities
 WHERE similarities.term='t'
 AND similarities.head = root.head
 ORDER BY similarities.sim DESC
```

All query rewriting strategies require the identification of the matching cluster previously. This is done using one query over the similarities table as described above. That table has 6,591,897 rows (56341 rows of the basic data set times 117 cluster heads). The runtime measurements for this query show a decent performance of at most 24 ms.

3.2 Query Answering without Derived Fragments

Assume the user sends a query

```
SELECT mesh,concept,patientid,tupleid
 FROM ill WHERE mesh ='cough'.
```

and 101 is the ID of the cluster containing cough. In the first strategy (lookup table) the rewritten query is

```
SELECT mesh,concept,patientid,tupleid
 FROM ill JOIN lookup
 ON (lookup.tupleid = ill.tupleid
 AND lookup.clusterid=101).
```

In the second strategy (extra clusterid column) the rewritten query is

```
SELECT mesh,concept,patientid,tupleid
 FROM ill-rr WHERE clusterid=101
```

In the third strategy (clustered fragmentation), the rewritten query is

```
SELECT mesh,concept,patientid,tupleid
 FROM ill-101
```

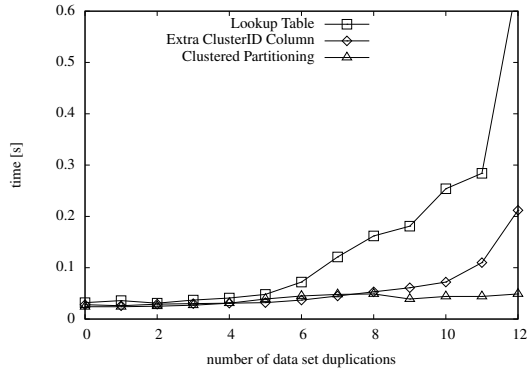


Figure 2: Queries without derived partitioning

The runtime measurements in Figure 2 in particular show that the lookup table approach does not scale with increasing data set size. The extra cluster-id column performs better, but does not scale either, when the data set becomes very large. The approach using clustered partitioning outperforms both by having nearly identical runtimes for all sizes of the test data set. Note, that after duplicating the data set 12 times it is 4096 times as large as the basic data set.

3.3 Query Answering with Derived Fragments

We tested a JOIN on the patient ID with a secondary table called *info* having a column *address*. The original query is

```
SELECT a.mesh,a.concept,a.patientid,
 a.tupleid,b.address
 FROM ill AS a,info AS b
 WHERE mesh='cough'
 AND b.patientid= a.patientid
```

In the first strategy (lookup table) the rewritten query is

```
SELECT a.mesh,a.concept,a.patientid,
 a.tupleid,b.address,lookup.clusterid
 FROM ill AS a,info AS b,lookup
 WHERE lookup.tupleid=a.tupleid
 AND lookup.clusterid=101
 AND b.patientid= a.patientid.
```

In the second strategy (extra clusterid column) the rewritten query is

```
SELECT a.mesh,a.concept,a.patientid,
 a.tupleid,b.address
 FROM ill AS a,info AS b
 WHERE a.clusterid=101
 AND b.patientid=a.patientid.
```

In the third strategy (clustered fragmentation), the rewritten query is

```
SELECT a.mesh,a.concept,a.patientid,
 a.tupleid,b.address
 FROM ill-101 AS a
 JOIN info-101 AS b
 ON (a.patientid=b.patientid).
```

We devised two test runs: test run one uses a small secondary table (each patient ID occurs only once) and test run two uses a large secondary table (each patient ID occurs 256 times). For the first rewriting strategy (lookup table) the secondary table is a non-fragmented table. For the second strategy, the secondary table is distributed in round robin fashion, too. For the last rewriting strategy, the secondary table is fragmented into a derived fragmentation: whenever a patient ID occurs in some fragment in the *ill-i* table, then the corresponding tuples in the secondary table are stored in a fragment *info-i* on the same server as the primary fragment.

Figure 3 presents the runtime measurements for queries with derived fragments with the small secondary table (one matching tuple in the secondary table for each tuple in the primary table). It can be observed that the necessary join operation causes all three approaches to perform significantly worse. The clustered partitioning strategy still shows the best performance with being roughly twice as fast as the other ones. While the lookup table approach performed worst without derived fragments, it performed better than the extra cluster-id column strategy when tested with derived fragments using small secondary tables.

However, as can be seen in figure 4 both approaches are clearly outperformed by the clustered partitioning strategy when the secondary table is large (256 matching tuples in the secondary table for each tuple in the primary table). It delivers feasible performance up to

6 – 7 data set duplications, while the lookup table and extra cluster-id column approaches fail in doing so after only 2 – 3 data set duplications.

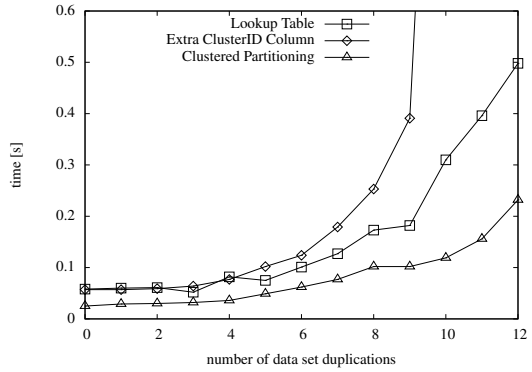


Figure 3: Queries with derived partitioning (small secondary tables)

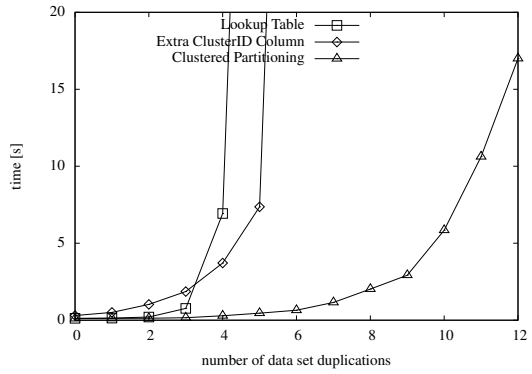


Figure 4: Queries with derived partitioning (large secondary tables)

3.4 Insertions and Deletions

We tested the update behavior for all three rewriting strategies by inserting 117 new rows (one for each cluster).

Any insertion requires identifying the matching cluster i again (Section 3.1). Each insertion query looks like this for mesh term m, concept c, patientid 1 and tupleid 1:

```
INSERT INTO ill
 VALUES ('m','c',1,1).
```

In the first rewriting strategy, the lookup table has to be updated, too, so that two insertion queries are executed:

```
INSERT INTO ill
 VALUES ('m','c',1,1).
INSERT INTO lookup
 VALUES (i,1).
```

For the second rewriting strategy, the extra clusterid column contains the identified cluster i:

```
INSERT INTO ill-rr
 VALUES ('m','c',1,1,i).
```

For the third rewriting strategy, the matching clustered fragment is updated:

```
INSERT INTO ill-i
 VALUES ('m','c',1,1).
```

After the insertions we made a similar test by deleting the newly added tuples.

Deletions require queries of the basic form

```
DELETE FROM ill WHERE mesh='m'.
```

In the first rewriting strategy, the corresponding row in the lookup table has to be deleted, too, so that now first the corresponding tuple id of the to-be-deleted row has to be obtained and then two deletion queries are executed:

```
DELETE FROM lookup
 WHERE lookup.tupleid
 IN (SELECT ill.tupleid FROM ill
 WHERE mesh='m').
DELETE FROM ill WHERE mesh='m'
```

For the second rewriting strategy, no modification is necessary apart from replacing the table name and no clusterid is needed:

```
DELETE FROM ill-rr WHERE mesh='m'
```

For the third rewriting strategy, the matching clustered fragment i is accessed which has to be identified first (Section 3.1):

```
DELETE FROM ill-i WHERE mesh='m'
```

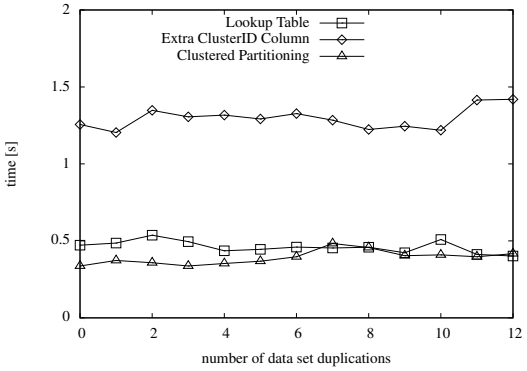


Figure 5: Insertion

As shown in Figure 5, the runtime for insertions appears to be constant for all approaches. Interestingly only the round robin approach performs worse by factor 2.5.

Figure 6 presents the measurements for deletions. Here the runtimes for the extra cluster-id column and clustered partitioning approach is constant and on a similar level, while the lookup table strategy performs roughly 4 times worse due to its higher complexity. Starting from a certain data set size the deletion time of this approach even begins to grow significantly further.

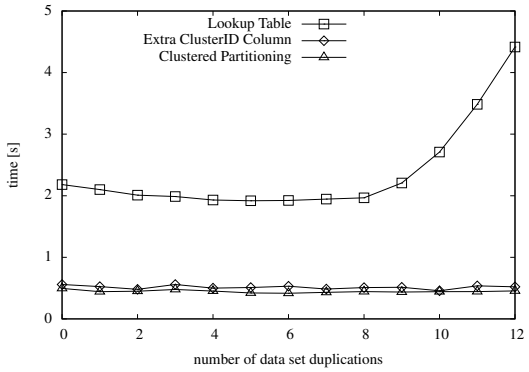


Figure 6: Deletion

3.5 Recovery

Lastly, we tested how long it takes to recover the clustered fragmentation by either using the lookup table or the extra column ID. For the lookup table approach this is done using the following query on the original table and the lookup table by running for each cluster i:

```
INSERT INTO c_i SELECT mesh, concept,
patientid, ill.tupleid FROM ill
 JOIN lookup
 ON (lookup.tupleid=ill.tupleid)
 WHERE lookup.clusterid=i
```

for each cluster i.
For the round robin fragmented table with the extra clusterid column the query for each cluster i is as follows:

```
INSERT INTO c_i
 SELECT mesh, concept, patientid,
tupleid FROM ill-rr
 WHERE clusterid=i
```

In both cases this results one c_i table per cluster.

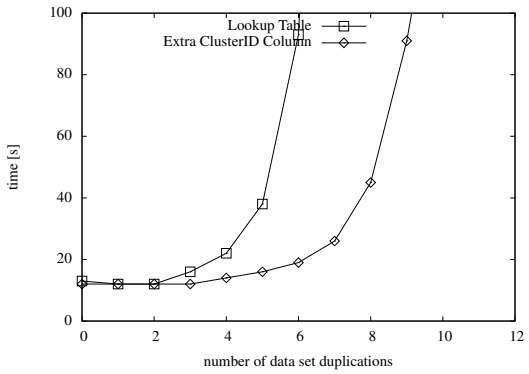


Figure 7: Recovery

As can be seen in figure 7 both recovery procedures become unfeasible very quickly with the approach for the extra cluster-id column strategy being able to handle 2 – 3 data set duplications more in an acceptable timeframe.

4 Conclusion and Future Work

With the larger cluster and the larger dataset, we verified the result that – due to the small size of the partitioned tables – the runtime performance is best for the clustered partitioning approach and the overhead of metadata management is negligible. In particular, it outperforms the lookup table approach that stores for each cluster the corresponding tuple IDs does not scale well as the data set size grows.
Current work covers the following extensions:

- We are adding data locality constraints to the system so that fragments that are accessed together frequently can be placed on the same server.
- We are adding the feature of a more dynamic replication based on common subfragments and optional conflict constraints.
- We are investigating dynamic adaptation of the clustering: whenever values are inserted or deleted, the clustering procedure on the entire data set might lead to different clusters.
- Comparing R and the PAL for obtaining a clustering of terms.

References

[1] K. Inoue and L. Wiese. Generalizing conjunctive queries for informative answers. In *Flexible Query Answering Systems*, pages 1–12. Springer, 2011.
[2] U.S. National Library of Medicine. Medical subject headings. http://www.nlm.nih.gov/mesh/.
[3] L. Wiese. Clustering-based fragmentation and data replication for flexible query answering in distributed databases. *Journal of Cloud Computing*, 3(1):1–15, 2014.
[4] L. Wiese. *Advanced Data Management: For SQL, NoSQL, Cloud and Distributed Databases*. DeGruyter/Oldenbourg, 2015.

Natural Language Processing for In-Memory Databases: Boosting Biomedical Applications

Mariana Neves
Hasso-Plattner-Institut
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
mariana.neves@hpi.de

Abstract

Clinicians and researchers in the biomedical domain often need to look for information about particular disease, gene or protein. The scientific publications are valuable source of information but looking for a specific answer and fact among the millions of abstracts available at the PubMed database might take many hours or days. Natural language processing allows linguistic and semantic processing of textual documents while the in-memory database technology has the potential of speeding up this process to support the development of real-time applications. In this report, I describe the activities that me and my students performed in the last six months when making use of a large SAP HANA instance which belongs to the HPI Future SOC Lab resources.

1 Introduction

The current data deluge demands fast and real-time processing of large datasets to support various applications, also for textual data, such as scientific publications. In-memory database (IMDB) technology allows scalability of traditional methods to process large document collections. In this report, I give an overview of the resources that I am currently using in the HPI Future SOC Lab, the projects that makes use of these resources and future work I plan to carry out in future.

I rely on an instance of 1 Tb of memory of the SAP HANA database from the HPI Future SOC lab, into which I imported around 15 millions scientific publications (titles and abstracts) which were retrieved from the PubMed database[1]. This large collection of documents is currently being used for various research activities and applications related to natural language processing (NLP) and text mining, as described in the next sections of this report.

[1] http://www.ncbi.nlm.nih.gov/pubmed

2 Linguistic processing

There is a variety of preliminary NLP tasks (cf. Figure 1) [2] which are usually carried out in documents in many NLP applications. For instance, each document needs to split by sentences and the later separated by tokens (words). The next step usually consists in classifying each token with the corresponding part-of-speech (POS) tag, i.e., assigning whether the word is an adjective, verb, noun, etc. The POS tags are used in various NLP tasks, e.g., for named-entity recognition, i.e., for identifying named entities, such as genes and disease names in textual documents. Examples of more advanced linguistic processing include chunking and semantic role labeling (SRL). Chunking, also called shallow parsing, consists in splitting the sentence into chunks or phrases, followed by classification in predefined classes, such as noun or verbal phrases. This is an important input for many NLP applications, and specially for relationship extraction. Finally, SRL consists in identifying predefined predicates, usually a verb, and their corresponding arguments, usually nouns or noun phrases. This information is particularly helpful for applications which need to rely on advanced linguistic processing, such as question answering.

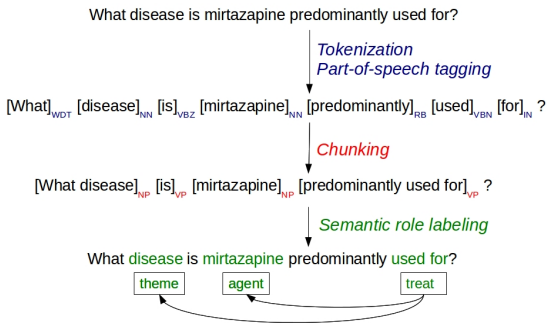


Figure 1: Example of the various linguistic processing tasks which were applied to the sentence "What disease is mirtazapine predominantly used for?".

From the above described NLP linguistic tasks,

tokenization and POS tagging are automatically computed in HANA when creating a full text index for a column which contain textual documents. However, the returned POS tags and words are frequently incorrect when processing documents from the biomedical domain, as specific methods trained in biomedical documents are usually necessary to obtain precise results. Moreover, chunking and semantic role labeling is still inexistent in the HANA database, which hinders its utilization for some NLP applications, such as question answering and information extraction.

During the Master Project "Ask your database" in the HPI summer term, three students (Fabian Eckert, David Heller and Thomas Hille) analyzed the suitability of the existing NLP features in HANA in the scope of the development of a question answering system for biomedicine. The students evaluated the tokenization feature of the SAP HANA database and developed a post-processing step to correct some of the words which were incorrectly tokenized. The problem occurs frequently due to the complexity of the biomedical nomenclature which includes many characters other than letters, e.g., hyphens, numbers, slashes, such as in gene and chemical names.

Additionally, the students implemented new methods for chunking and SRL in the HANA database. Both approaches relied on the machine learning (ML) algorithms available in the Predictive Analysis Library (PAL), such as Support Vector Machines, Naive Bayes and Decision Trees. When using supervised learning algorithms, a first step is training a model based on manually annotated data, followed by the evaluation of the model on unknown data. Therefore, the students used the Genia [3] and the BioProp [4] corpora for training and evaluating the chunking and SRL components, respectively, and obtained 90% and 80% precision for the tasks, respectively.

3 Question answering

Question answering (QA) [1] is the task of posing questions, instead of only keywords, to a system and getting exact answers in return, instead of only potentially relevant documents. For instance, for the question "What disease is mirtazapine predominantly used for?" derived from the BioASQ dataset[2], the user expects to receive one or more disease names in return, e.g., "depression", instead of documents which match the question keywords. Besides providing exact answers, QA systems can also return tailored summaries for the questions for providing more details and context for the answer. QA systems usually include there main steps [2]: (a) question processing and generation, (b) retrieval of relevant passages (sentences) which are relevant to the query, and (c) extraction of the answer and generation of an adequate summary to complement the answer.

I had previously worked on QA using the SAP HANA database, but my research was limited to the passage retrieval component. Further, I did not use a local copy of the PubMed database and relied on a small collection of around 100,000 publications previously retrieved from PubMed. During the Master Project "Ask your database" in the HPI summer term, six students (Tim Draeger, Daniel Dummer, Alexander Ernst, Pedro Flemming, Ricarda Schüler and Frederik Schulze) worked in the development of a QA system. The team implemented new methods as stored procedures for all QA steps and developed a Web interface[3] for the system (cf. Figure 2).

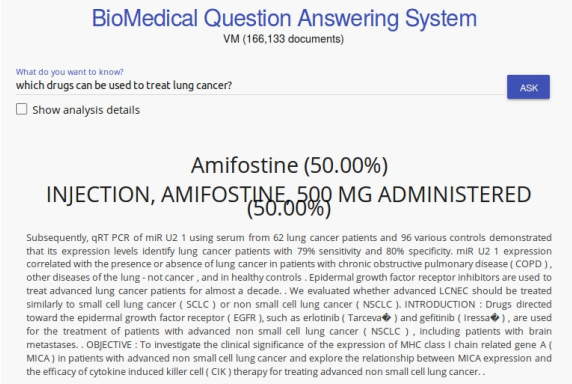


Figure 2: Screen-shot of the question answering system for biomedicine.

The students also integrated various terminologies from UMLS database into HANA, which were used to detect entities of various types (genes, diseases, chemicals, etc) both in the questions and in the documents. For this purpose, they used the built-in dictionary-based named-entity recognition feature available in HANA. For the question processing step, they relied on the chunking method developed by their colleagues (cf. above) and implemented methods to predict the question type (yes/no, factoid, list or summary) and the target of the question. e.g., a gene or disease name, and to construct a query derived from the question. For the passage retrieval step, the students implemented information retrieval methods to select and rank relevant sentences to the query. Finally, in the answer extraction step, the students created methods for extracting the answers, based on the previously recognized entities, and for automatically retrieving and ordering sentences to generate a tailored summary which provide further answer to the question. All steps of the QA system were evaluated on the BioASQ dataset [5].

[2] http://bioasq.org/

[3] http://vm-hig-mp2015.eaalab.hpi.uni-potsdam.de:9050/

Despite the success of the Master Project, the student have reported various problems when using the HANA database for developing the QA system and, especially, when dealing with the huge collection of around 15 millions documents from PubMed. Indexing these documents into the database requires carefully running a sequence of tasks which consists in: (i) creating the table, (ii) creating an empty full text index on the empty table, (iii) partitioning the index, and (iv) importing the documents gradually. This procedure is even more complex due to the need of having a duplication of the data, as two identical columns were created in the table into which the publications were imported. This is due a limitation in HANA that only allows one index per column, when we need two of them, one for the linguistic processing (words, sentences, POS tags) and one for the semantic processing (e.g., detection of gene and disease names). Currently, the QA system is running on a small collection of around four millions documents instead of the 15 millions available in PubMed, due to the impossibility of loading all PubMed publications.

4 Multilingual processing

I explored the multilingual text analysis features of HANA in the scope of constructing a parallel corpus in collaboration with three other colleagues: Dr. Aurélie Névéol (LIMSI-CNRS, France), Dr. Antonio Jimeno (IBM research, Australia) and Dr. Karin Verspoor (University of Melbourne, Australia). I have crawled and collected scientific publications from the Scielo database[4], namely titles and abstracts, in four languages: English (EN), French (FR), Portuguese (PT) and Spanish (ES).

I imported the documents into the HANA database and utilized built-in features for language recognition, sentence splitting and tokenization. I also performed a comparison of the sentence splitting output returned by HANA and by the Apache OpenNLP Java library[5] for some selected sentences and I observed that results were better when using HANA. The corpus is composed of many thousands of titles and abstracts for each language pair (EN/FR, EN/PT, EN/ES) (cf. Figure 3) and it will be used in a shared task in the 2016 edition of the Workshop of Machine Translation (WMT)[6].

5 Future work

Future work to be carried out in my HANA instance in the Future SOC Lab will include activities related to the Master Project HP/N "Learning to Note" and the Bachelor Project HP2 "Text Mining for Biomedical Applications" during the next winter term and year, respectively.

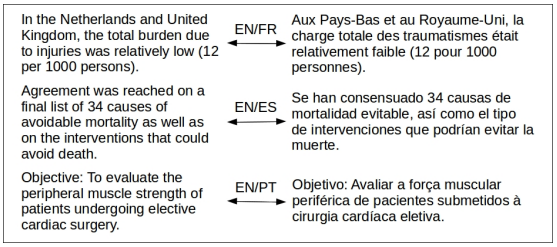


Figure 3: Examples of parallel sentences in four languages: English (EN), French (FR), Portuguese (PT) and Spanish (ES).

In the scope of the Master Project HP/N, four students will develop an intelligent annotation tool and implement information extraction methods based on semi-supervised algorithms. This approach needs to rely on large collection of textual data, i.e., the biomedical publications from PubMed, and on machine learning algorithms, making HANA a suitable framework for implementing such methods.

In the scope of the Bachelor Project HP2, five students will perform improvements on the question answering system developed during the Master Project "Ask your database" during the summer term. The students will also be in charge of adapting the system to the needs of a client, the American Society of Clinical Oncology (ASCO), in collaboration with the SAP Innovation Center Potsdam (ICP).

Additionally, we will also use the HANA instance for running an application on sentiment analysis whose prototype was developed by the student Fabian Eckert during the Seminar "In-Memory Databases: Applications in Healthcare" in the summer term. This application allows users to visualize the evolution of the opinion expressed in the scientific publications for associations between certain diseases and consuming certain foods.

[4] http://www.scielo.br/
[5] https://opennlp.apache.org/
[6] http://www.statmt.org/wmt15/

References

[1] S. J. Athenikos and H. Han. Biomedical question answering: A survey. *Computer Methods and Programs in Biomedicine*, 99(1):1 – 24, 2010.

[2] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall International, 2 revised edition, 2013.

[3] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182, 2003.

[4] R. Tsai, W.-C. Chou, Y.-S. Su, Y.-C. Lin, C.-L. Sung, H.-J. Dai, I. Yeh, W. Ku, T.-Y. Sung, and W.-L. Hsu.

Biosmile: A semantic role labeling system for biomedical verbs using a maximum-entropy model with automatically generated template features. *BMC Bioinformatics*, 8(1):325, 2007.

[5] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, et al. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138, 2015.

Extending Analyze Genomes to a Federated In-Memory Database System For Life Sciences

Matthieu-P. Schapranow, Cindy Perscheid
Hasso Plattner Institute
Enterprise Platform and Integration Concepts
August–Bebel–Str. 88
14482 Potsdam, Germany
{Schapranow|cindy.perscheid}@hpi.de

Abstract

Cloud computing has become a synonym for elastic provision of shared computing resources operated by a professional service provider. However, data needs to be transferred from local systems to shared resources, such as our Analyze Genomes platform, for processing, which might result in significant process delays and the need to comply with special data privacy acts. Based on the concrete requirements of life sciences research, we share our experience in integrating existing decentralized computing resources to form a federated in-memory database system. Our approach combines advantages of cloud computing, such as efficient use of hardware resources and provisioning of managed software, whilst sensitive data is stored and processed on local hardware only.

1. Project Idea

Latest medical devices, such as Next-Generation Sequencing (NGS) machines, generate more and more fine-grained diagnostic data in a very short period of time. In the scope of the Analyze Genomes project, we have built a platform for analyzing such data and combining it with scientific data from distributed data sources [6, 3]. Using our cloud services currently requires transferring the own data from local sites to our shared computing resources prior to its execution. However, especially processing of NGS data can involve 750 GB and more per patient sample [5]. As a result, even with increasing network bandwidth, sharing data results in a significant delay due to data duplication when following state-of-the-art analysis models. In addition, sharing of medical data requires very specific handling and exchange is limited, e.g. due to legal and privacy regulations, such as the Data Protection Directive of the European Union [2]. The collaboration of international life science research centers and hospitals all over the globe is important to support the finding of new scientific insights, e.g. by sharing of selected knowledge about existing patient cases. However, nowadays collaborations face various IT challenges, e.g. heterogeneous data formats and requirements for data privacy. In the past two Future SOC lab periods, we therefore focused on an approach we call Federated In-Memory Database (FIMDB) [7]. The system combines local computing resources of research facilities and Analyze Genomes services as managed services in a unique hybrid cloud approach as depicted in Figure 1. We attempt to build a unique cloud setup integrating decentralized computing resources, which reside on local sites with the data, whilst the service provider manages algorithms and apps remotely in our system. In cooperation with a research facility in Berlin, we have started to set up our FIMDB, by connecting their cloud infrastructure with the Future SOC lab's resources. In the following, we document our current progress in setting up such

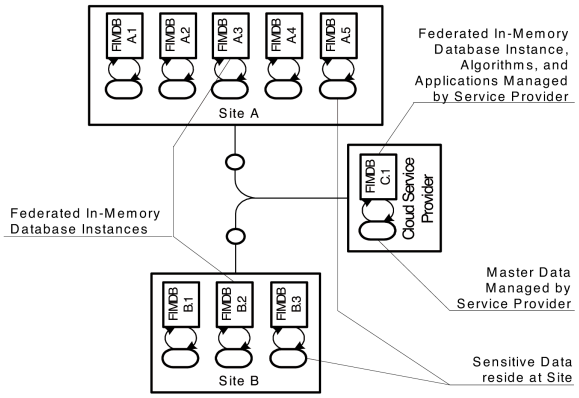


Figure 1: Computing resources and data reside on local sites whilst the service provider manages algorithms and apps remotely in our system.

a system and outline further steps that need to be undertaken.

2. Realization

In the last Future SOC lab period, we had accomplished to establish a Virtual Private Network (VPN) connection between the Future SOC and the research facility's network. We also configured the remote directories, and set up two database instances on two of their local computing nodes. These were then connected to our database landscape located at the Future SOC lab's cluster. We have conducted further steps in order to adapt our Analyze Genomes system to also use the resources provided by the research facility. These include the subscription to and configuration of our managed services. In addition, we needed to extend our worker framework by further functionality. As we now have data that must reside on the local computing resources, our scheduler must be able to assign specific computation resources for a pipeline execution.
Setting up such a system is a challenging task, which takes its time to be finished. As a result, we were not able to test and benchmark our FIMDB system by the end of the current Future SOC Lab period. Therefore, we concentrate on describing our research steps conducted to set up our system for further evaluation.

2.1 Subscription to Managed Service

The services for processing and analyzing genome data, which are offered by our Analyze Genomes platform, are accessible via a web application. On the one hand, we at Analyze Genomes act as a service provider by hosting the project website and its applications at our side. On the other hand, users from research sites act as customers by using the services offered by us via the web application. We support local user accounts, i.e. users need to register with a user name and password. In addition, we have already integrated existing authentication providers, such as OAuth, to allow successful authentication using credentials of existing accounts from social networks, e.g. Google, Facebook, LinkedIn [1].

Service Provider A database administrator, who creates distinct database schemata for each research facility and their data, represents the service provider. A database schema can be seen as a dedicated space for the research facility, containing everything from database tables to functions and stored procedures, which belong to the research facility. Figure 2 on the left depicts an example schema whose content is structured in

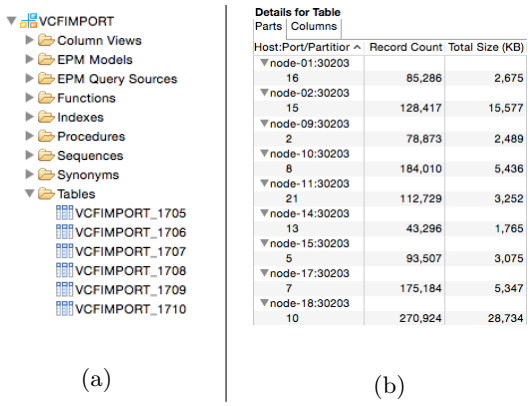


Figure 2: (a) A FIMDB database schema groups data entities, e.g. database tables, functions, and stored procedures; (b) selected partitioning details of a database table of the FIMDB.

folders and containing all sequences, functions, tables, or indices, respectively. The service provider is responsible for keeping each database schema isolated from the rest, i.e. tenant-specific data is separated to ensure data privacy [4]. Only the research facility to which the database schema belongs to will be granted access by the database administrator. The latter is responsible for managing all users and their assigned roles. In addition, he also makes sure that the data is partitioned correctly and that the partitions are distributed to the right sites. Although the data of the research sites is integrated into our FIMDB system, the data itself will be distributed exclusively to the computing resources of the research facility. By that, we can ensure that the data remains at the research sites for both data storing and processing all the time.

Customer An application administrator represents the customer from the research facility. On behalf of the research facility, the application administrator subscribes to the managed services of our Analyze Genomes service portfolio. As a result, the administrator is granted administration rights for the selected application and the specific department. This includes maintaining user groups and access rights within the application for users. In addition, the application administrator must map these application users to corresponding database users, which are maintained by the service provider. Figure 2 on the right depicts selected details of a database table partitioned across the FIMDB. For example, the first line describes that chunk 16 of the database table, which has a cardinality of 85,286 entries and a size of 2.6MB, is

stored on the Future SOC computing resource named node-01.

2.2 Configuration of Selected Service

For configuring the services offered by our Analyze Genomes platform, end users at research sites are able to maintain their personal profiles and tailor application settings. In our application scenario, each user has to define a local home directory. This directory then contains all data that belongs to the research site: On the one hand, it is the data that has been produced by them and that is analyzed by our services. On the other hand, it is the data that is generated by our services during the analysis process.

2.3 Adaptations to the Worker Framework of Analyze Genomes

The backend of our Analyze Genomes platform consists of a set of workers operating on distinct nodes on the cluster. Once a pipeline is triggered for processing genome data, it is split up into atomic tasks, which are assigned and delivered to a particular worker. A dedicated scheduler component is part of the Analyze Genomes worker framework. The scheduler guarantees a smooth pipeline execution and is able to flexibly react on errors occurring during execution, e.g. restart of jobs or pipeline execution on another worker. When starting all workers, one of them will also possess the role of the scheduler and thereby be responsible for assigning tasks to workers and managing the execution of the distinct pipeline steps. In our original system setup, those tasks were assigned to any worker who had free capacities. With the use case of the FIMDB system, we have to adapt this procedure, as we need to satisfy the increased privacy requirements of the research sites. In order to ensure that the data of a research site is processed exclusively on local machines within our FIMDB system, we need to adapt and extend our backend, e.g. user administration and scheduling capabilities.

User Administration In order to execute a pipeline on a particular research site, a scheduler needs to be able to recognize it. Analyze Genomes already provided a rudimental user administration - that is, the user had to be logged in in order to use our services. Pipeline executions therefore could be traced to the corresponding user who had started it. With the current status of the system, however, the scheduler is not able to associate a user with the execution on a particular set of nodes. Instead, all computing resources are used for pipeline execution. With the setup of our FIMDB system, we now need to identify corresponding computation resources for a particular user when a pipeline is executed. For that, we introduce user groups to our system. When a research site is included into our FIMDB system, the database administrator creates a corresponding user group. Each user from that research site will be assigned exclusively to this group. No researcher can belong two more than one group of a research facility, as this implies that data from one research facility could potentially be executed on the nodes belonging to another research facility. With the help of user groups, our scheduler is able to identify whether the pipeline of a particular user needs to be executed on a subset of the computing resources of the complete FIMDB system.

Resource Allocation The standard scheduling routine applied in Analyze Genomes' worker framework assigns tasks to any worker that is not working on any task. However, with our FIMDB system setup, we need to make sure that tasks of a pipeline get executed only by a subset of specific workers belonging to the facility of the task's creator. Thus, we need to extend our scheduler to use his knowledge on the user and his user group to distribute open tasks correctly. For that, we have introduced a mapping of workers and their nodes to corresponding user groups. As each research facility or department uses their own computational resources exclusively, and each research facility or department makes up one user group, we have a one to one mapping from user groups to nodes. By that, once a user triggers a pipeline execution, the pool of available workers for the scheduler reduces by those running on the nodes that are assigned for the user group the user belongs to.

3. Conclusion

We are convinced that sharing knowledge is the foundation to support research cooperation and to discover new insights cooperatively. Therefore, we aim at realizing our approach of an FIMDB system, which enables research sites to make use of cloud services whilst their sensitive data remains on-site. By this, research sites can make use of an existing technical infrastructure, and can avoid transferring their huge data amounts to another site, thus preventing a significant delay in the course of processing big medical data sets.
In this report, we provided insights on the steps undertaken after the infrastructural requirements, such as an existing VPN connection and database

instance added to the landscape, have been fulfilled. We have outlined the subscription to and configuration of our managed services, and also provided insights into the adaptations necessary for the worker framework in order to guarantee that data is processed in the research sites. Setting up such a system is challenging, and we have not yet fully deployed the system; we are aiming at running first tests on it in the next Future SOC lab period together with our partners.

References

[1] D. Hardt. RFC6749: The OAuth 2.0 Authorization Framework. `http://tools.ietf.org/html/rfc6749/`, 2012. last accessed: Mar 27, 2015.

[2] R. Fears, H. Brand, R. Frackowiak, P.-P. Pastoret, R. Souhami, and B. Thompson. Data protection regulation and the promotion of health research: getting the balance right. *QJM*, 107(1):3–5, 2014.

[3] H. Plattner and M.-P. Schapranow, editors. *High-Performance In-Memory Genome Data Analysis: How In-Memory Database Technology Accelerates Personalized Medicine*. Springer-Verlag, 2014.

[4] J. Schaffner. *Multi Tenancy for Cloud-Based In-Memory Column Databases*. Springer, 2013.

[5] M.-P. Schapranow, F. Häger, C. Fähnrich, E. Ziegler, and H. Plattner. In-Memory Computing Enabling Real-time Genome Data Analysis. *International Journal on Advances in Life Sciences*, 6(1-2), 2014.

[6] M.-P. Schapranow, F. Häger, and H. Plattner. High-Performance In-Memory Genome Project: A Platform for Integrated Real-Time Genome Data Analysis. In *Proceedings of the 2nd Int'l Conf on Global Health Chall*, pages 5–10. IARIA, Nov 2013.

[7] M.-P. Schapranow, C. Perscheid, A. Wachsmann, M. Siegert, C. Bock, F. Horschig, F. Liedke, J. Brauer, and H. Plattner. A federated in-memory database system for life sciences. *Business Intelligence for the Real Time Enterprise (BIRTE) Workshop at VLDB*, 2015.

Interactive Product Cost Simulation on Coprocessors

Christian Schwarz
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
christian.schwarz@hpi.de

Christopher Schmidt
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
christopher.schmidt@student.hpi.de

Abstract

Enterprise systems evaluate complex data dependencies within their applications, requiring computational power of today's modern CPUs. Furthermore, in businesses, simulations support decision makers as they can better estimate the effect of changes. When modeling enterprise specific application logic as system of equations, hybrid architectures can be leveraged to speed up execution times. Nevertheless, today's enterprise systems mainly ignore the availability of modern coprocessors, such as NVIDIA's Tesla. The project evaluates the applicability of coprocessors for in-memory database centric enterprise applications on the example of product cost calculation. We explore how to utilize the compute capabilities of graphical processing units in the enterprise specific simulation scenario. For this purpose, we present a CUDA-enabled implementation of the matrix inversion, exploiting features of the business logic modeled in the matrix to speed up the calculation. Afterwards we evaluate the performance of our implementation and compare it with a CPU-based implementation and a GPU-accelerated library implementation.

1 Motivation

In the manufacturing industry changes in raw material prices have a high impact on the production costs of manufactured goods. Cost center rates, such as electricity expenses or wages, influence the production costs of these goods through the manufacturing process.

With a product cost simulation prototype developed at the Enterprise Platform and Integration Concepts group at the Hasso Plattner Institute this challenge is addressed, simulating the impact of changed material prices and cost center rates for a manufacturing company, returning the expected new product costs for all products of the company within sub-seconds to enable interactive applications. The complex manufacturing steps for products, including multiple input materials and several stages of intermediates can be modeled in a system of linear equations. The resulting linear system can be presented as a matrix, which enables to use common matrix operations, including matrix inversion and matrix vector multiplication to calculate the simulated product costs. Both operations have been studied extensively and proven to be suitable for parallel execution across multiple cores [1, 2], speeding up the execution times. Using the available knowledge about the structure of the matrix, we can further improve the algorithm for the inverse calculation.

The faster execution time achieved through the parallel execution of the operations, results in a higher CPU usage during the runtime. In a real business setting, running the product cost simulation would therefore lower the overall system's transactional throughput, presenting a potential bottleneck for the everyday business. To overcome this drawback, we investigate how graphical processing units (GPUs) can be utilized to prevent this disadvantage. Moving the matrix operations on to the GPU is valuable due to two reasons. First, the operations performed on the matrix are data-parallel problems, as the same algebraic operations are executed on different data points of the matrix. GPUs provide a high amount of parallel computing resources that can speed up the calculation process in comparison to CPUs, from a cost and performance perspective. Second the SIMT [3] architecture of modern graphical processing units is designed especially for problems of this kind, where one instruction is executed by multiple threads on different data. Therefore using a GPU will result in load reduction on the CPU during calculation, and in speed-up of the execution time. For the implementation presented in this paper we use a NVIDIA Tesla K20Xm graphics card [4], provided by the HPI Future SOC Lab.

The remainder of this report is organized as follows. Section 2 gives an overview about related work regarding the matrix operations. Section 3 gives a brief overview of the product cost simulation, including the data sources and the algorithms used for the computation. Section 4 presents our implementation of the matrix inversion, based on CUDA [5]. In Section 5, we give an overview about our preliminary results, compared to two different methods. We draw a conclusion

and give an outlook on the next steps in Section 6.

2 Related Work

The performance of matrix inversion on graphical processing units has already been investigated by Ezzatti et. al [6]. They compared implementations on CPU, GPU and hybrid versions, using a LU factorization as implemented in LAPACK [7] and the Gauss-Jordan elimination procedure, which is also the foundation for the implementation presented in this report. Their results indicate that the Gauss-Jordan elimination is a well-suited algorithm for parallel computing. Additionally they show that hybrid implementations, exploiting the underlying platform features can still outperform pure GPU versions. One problem arising with computations on the GPU is the limited memory available. With a growing matrix size, this issue will become a challenge that needs to be addressed. To reduce the memory required during the inversion Das-Gupta [8] proposes a modified version of the Gauss-Jordan elimination calculating the inverse within the original matrix space. Another possible approach to solve this issue is to calculate subsets of the entries of the inverse [9].

A different approach to the simulation problem is to use linear solvers instead of an inversion and matrix-vector multiplication. Krüger and Westermann [10] have proposed a framework for implementing linear algebra operators on GPU and used it to implement a high-performant sparse conjugate gradient solver. More work on linear solvers on GPU includes Tomov et. al [11], who present solver implementations on hybrid systems, which are GPU accelerated. They integrate their work into the Matrix Algebra on GPU and Multicore Architectures (MAGMA) project [12]. In contrast to linear solvers, the use of matrix inversion is beneficial for the simulation scenario, based on relatively small inversion overhead in comparison to the amount of matrix vector multiplication. In addition, the inverse matrix can be reused for other scenario relevant aspects, such fast traversal of hierarchies for data selection and visualization.

3 Calculation of Product Costs

The model underlying the product cost simulation is based on a system of linear equations, which can be expressed in a matrix, as shown in Equation 1. The resulting matrix is organized as depicted in Figure 1.

The transformation of the material dependencies in the manufacturing process into a matrix format allows us to solve the equation, which calculates the production costs for all products at once, using standard matrix operations.

3.1 Business Knowledge-Based Matrix Inversion

Equation 1 shows the two operations necessary for the computation, which is the inverse calculation of the matrix and afterwards a matrix vector multiplication. Due to the major benefits of using structural knowledge of a matrix for the inverse calculation, we focus on this step. Standard libraries, such as NVIDIA's cuBLAS [13], already provide highly parallel implementations for matrix vector multiplication, running on the GPU.

A common approach to calculate the inverse of a dense non-singular matrix is to apply the Gauss-Jordan elimination. This procedure first augments the matrix to the right with the identity matrix. In the next step, the matrix is transformed into the reduced row echelon form, which is achieved by applying elementary row operations. The transformation will turn the left part of the matrix into the identity matrix and the right part, which has been augmented, will contain the inverted matrix.

For the specific matrix described in Figure 1, we use the knowledge of the structure of the matrix to reduce the number of operations needed during the Gauss-Jordan elimination. At first, we see that the diagonal values are set to *1* for the bill of material quadrant and differ in the cost center load quadrant only. Knowing that the non-zero value in these rows is stored in the diagonal only, we need to divide them and the equivalent position in the augmented identity matrix. As a following step, all entries in the bill of operation quadrant are eliminated. We can parallelize all prior operations on item level. The last quadrant of the matrix, which needs to be reduced, is the bill of material. In this matrix quadrant, we profit from the lower triangular format of the sub matrix.

To avoid locking, the operation on the secondary rows, meaning the check for non-zero values and the subtraction, is chosen for parallelization.

3.2 Dataset

The data is provided by a real manufacturing company. It is limited to contain data for a single country the company operates in. It consists of *17,304* materials and *327* cost center activities, resulting in a matrix with a dimension of *17,631*. The materials are connected with each other by *34,578* bill of material entries and with cost center activities by *30,838* bill of operation entries. All data is retrieved via ODBC and stored within a SAP HANA instance, which was provided by the HPI Future SOC Lab. SAP's HANA provides business functions required by the prototype, such as foreign currency conversion [14].

$$\begin{pmatrix} ManufactoringCosts \\ CostCenterRates \end{pmatrix} = \begin{pmatrix} BillofMaterial & BillofOperation \\ 0 & CostCenterLoad \end{pmatrix}^{-1} * - \begin{pmatrix} MaterialPurchasePrice \\ CostCenterCosts \end{pmatrix} \quad (1)$$

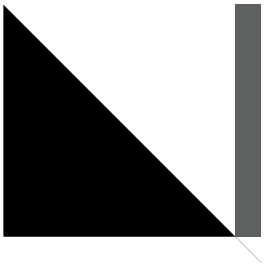


Figure 1: The product cost simulation matrix: Bill of Material quadrant (black) is a lower triangular matrix. Bill of Operation (dark grey) in general matrix format. The Cost Center Load quadrant (light grey) has values different to 0 on its diagonal. White spaces contain zero values.

4 GPU-powered Matrix Inversion

In the following description of our implementation, we reuse the three proposed separate steps of the inversion, according to the matrix quadrants. Therefore, we have a separate kernel for each of the quadrants.
The original matrix, which is processed during the inversion, is stored in a one-dimensional array in row-major ordering. In addition to the original matrix a second matrix, the augmented identity matrix using the same format is processed during the operation. It is simpler for later processing steps, for example the matrix vector multiplication, to keep both matrices separated. The matrices are copied to the GPU device, using `cudaMemcpy()` after the memory has been allocated using `cudaMalloc()`. Once the calculation is finished the inverted matrix is currently transferred back into host memory, which is achieved using cudaMemcpy(), for further processing. In future, we avoid this transfer, by performing the matrix-vector multiplication on the same device, reusing the data.

4.1 Invert Cost Center Load

The cost center load inversion is implemented in a kernel.The kernel accepts pointers to the matrix data and to the augmented identity matrix, which later stores the inverted data of the matrix. Each executed kernel operates on one cost center load data point. It divides the corresponding value in the diagonal of the same row in the identity matrix by the cost center load data point from the original matrix. The original matrix's value is set to *1*.
When executing this kernel the memory bandwidth becomes a bottleneck, as in the current matrix structure, the cost center loads in the diagonal are all far apart in physical memory. Therefore, no memory accesses by any threads within a warp can be combined, resulting in poor memory bandwidth. Since the cost center load is by far the smallest part of the matrix and only *327* values have to be accessed in the current product cost simulation, we have focused on improving other parts of the inversion algorithm. Nevertheless, a change in the underlying data structure, for example storing the diagonal of cost center load sub matrix as a separate array, is conceivable to improve the execution time in the future.

4.2 Invert Bill of Operation

The kernel for the inversion of the bill of operation requires the same information as the first kernel. Without any prior knowledge about the position of non-zero and zero values within this quadrant of the matrix requires processing each data point. For all data points having values different from 0 the corresponding data point within the identity matrix is calculated. Afterwards the data point's value in the original matrix is set to *0*.
Since all data points of this matrix quadrant are processed, the memory accesses for the original matrix and for the identity matrix can each be combined within a warp. The access to the memory containing the cost center load values is expensive, as no memory accesses by any threads within a warp can be combined. A solution for this issue has been proposed in the previous kernel description of the cost center load inversion 4.1.

4.3 Invert Bill of Material

When implementing an inversion for the bill of material we deal with two different access patterns for the matrix. The algorithm described in the section3.1 will loop through the matrix row by row. Since the matrix is in row-major order, the values within a row are stored in consecutive memory regions, allowing fast access on the GPU. When accessing the values within a column for the check for non-zeros, the access to the memory on the GPU will become a bottleneck, as each value has to be fetched separately. To overcome this challenge, an additional data structure for the column values is needed. Storing the complete original matrix in column-major order may not be wise due to memory limitations on the GPU. Therefore we propose a sparse data structure for the values in the lower triangular of the bill of material matrix quadrant. The data structure includes one array with the non-zero values, a second array storing the row of each value and

a third array storing the number of non-zero values for each column. Hence, the size of the first two arrays corresponds to the number of non-zero values and the third array is the size of the number of materials. Even though the memory alignment of this data structure is not optimal, yet, it allows combining the memory access within a warp and significantly improves the execution time of the bill of material.
In our current implementation, we iterate through the matrix rows on the CPU and start the bill of material inversion kernel for each row separately. In the future, this iteration could also be implemented on the GPU using CUDA's dynamic parallelism feature.

5 Performance Evaluation - Preliminary Results

In this section, we evaluate the implementation in regards to its execution time and compare the GPU implementation with different inversion implementations run on a CPU. We examine an implementation using the same knowledge about the matrix structure and a standard library's upper triangular matrix inversion algorithm. Additionally we will compare our matrix-specific implementation with a standard library's upper triangular matrix inversion algorithm also running on the GPU. For this purpose, we use the CULA dense library [15], which implements BLAS and LAPACK routines to be run on GPUs supported by NVIDIA's CUDA.

5.1 Benchmark Environment

For our performance measures, we use the following two systems. The performance measurements executed on the CPU are run on a system with SUSE Linux Enterprise Server 11 Service Pack 2 installed, equipped with a two hexacore Intel Xeon Processor E5-2640.
For the measurements executed on a GPU, we use a NVIDIA Tesla K20Xm graphics card, with CUDA compute capability 3.5, 14 streaming multiprocessors with 192 CUDA cores each and a warp size of 32. The hardware was provided by the HPI Future SOC Lab. When executing kernels a maximum number of 1,024 threads per block is allowed. The total amount of global memory available on the device is 5,760 MB and we use the CUDA driver version 7.0. The graphics card is integrated in a system with SUSE Linux Enterprise Server 11 Service Pack 3 installed, which is equipped with two quadcore Intel Xeon Processor E5620. For the measurements executed on the CPU we use the C++11 library `chrono`, using the function `std::chrono::system_clock::now()` to obtain the start time and the stop time. To calculate the execution time on the GPU, we use the `cudaEventElapsedTime()` function, which calculates the elapsed time between two cudaEvents.

5.2 Comparing GPU- and CPU-based implementations

In the first evaluation, we compare the presented GPU-accelerated implementation of the matrix inversion with three implementations running on the CPU. Two of the three CPU versions are also separated into three different steps, operating on the matrix quadrants presented in Figure 1 in the way presented in our algorithm description in Section 3.1. The first version is a standard implementation using OpenMP [16] for row level and advanced vector extensions (AVX) [17] for item level parallelization. The second CPU version, is using the triangular matrix inversion function `strtri()` from the Intel Math Kernel Library [18]. The reordering of the rows and columns required for the algorithm is not included in the measurements, as we could select the matrix in this format already from the database. For this inversion, we provide a total execution time, as the function operates on the complete matrix. The results are shown in Table 1.

	GPU	CPU	CPU_{strtri}
Cost Center Load	0.05	1.5	-
Bill of Operation	1.4	31	-
Bill of Material	254.2	2,227	-
Total	255.6	2,260	6,699
Total incl. transfer	1,088.2	2,260	6,699

Table 1: GPU vs CPU - Median execution time for 10 executions in ms

The transfer time for the GPU includes data transferred to the device, as we assume that in the future the inverted matrix can be utilized again during the matrix vector multiplication. The result is also not required by the user immediately. Comparing the total execution times for the algorithms, the GPU outperforms even the best CPU version by a factor greater than *8*. When the transfer time is included, the execution time of the GPU-accelerated version is approximately *2* times faster.

5.3 Comparing GPU-accelerated implementations

We now compare our GPU-accelerated version to a version from a library, which executes its code on the GPU. We use the Cula dense library's implementation of the LAPACK function `strtri()`. We compare us to the triangular inversion, as the matrix can be expressed in this format after reordering its rows and its columns. The reordering is not included in the measurements. We have executed each version *10* times and take the median execution time in milliseconds. We also include the transfer time in both measurements, as we cannot distinguish between execution

and transfer when using the library's implementation. Our implementation took *1,088* milliseconds compared to *2,372* milliseconds using `stritri`. Hence, we also show that our inversion, which uses available business logic, is approximately by factor *2* faster than an optimized and architecturally tuned library implementation.

6 Conclusion and Next Steps

We propose a GPU-accelerated matrix inversion algorithm for an enterprise specific application scenario. We have presented how the available business knowledge is leveraged to provide an implementation for the matrix inversion, enabling interactive applications. Preliminary measurements show that we were able to achieve a speed up of approximately factor *8* towards a CPU version, which also exploits the business knowledge and a speed up of factor *2* towards a comparable library implementation running on the GPU. These results suggest to integrate the GPU-accelerated inversion fully into the current prototype, and to exploit the GPU further for the matrix vector multiplication as well. As the prototype stands for a class of enterprise problems, we are confident that the chosen implementation can be used as a blueprint for future applications of the same class.
As next steps, we will evaluate the performance of the implemented algorithms based on an extended real world dataset.

References

[1] Bruce Hendrickson, Robert Leland, and Steve Plimpton. An Efficient Parallel Algorithm for Matrix-Vector Multiplication. *International Journal of High Speed Computing*, 7:73–88, 1995.

[2] Enrique S Quintana, Gregorio Quintana, Xiaobai Sun, and Robert van de Geijn. A Note On Parallel Matrix Inversion. *SIAM Journal on Scientific Computing*, 22(5):1762–1771, May 2000.

[3] Erik Lindholm, John Nickolls, Stuart Oberman, and John Montrym. Nvidia Tesla: A Unified Graphics and Computing Architecture. *Micro, IEEE*, 28(2):39–55, April 2008.

[4] NVIDIA Corporation. *Tesla K20 GPU Accelerator, Board Specification*. NVIDIA Corporation, July 2013.

[5] NVIDIA Corporation. *CUDA C Programming Guide*, 7.0 edition, March 2015.

[6] Pablo Ezzatti, Enrique S Quintana-Orti, and Alfredo Remon. Using graphics processors to accelerate the computation of the matrix inverse. *The Journal of Supercomputing*, 58(3):429–437, April 2011.

[7] Edward Anderson, Zhaojun Bai, Christian Heinrich Bischof, Laura Susan Blackford, James Weldon Demmel, Jack J Dongarra, Jeremy J Du Croz, Sven Hammarling, Anne Greenbaum, A McKenney, and Danny C Sorensen. *LAPACK Users' Guide*. Third Edition. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, August 1999.

[8] Debabrata DasGupta. In-Place Matrix Inversion by Modified Gauss-Jordan Algorithm. *Applied Mathematics*, 04(10):1392–1396, 2013.

[9] Patrick R Amestoy, Iain S Duff, Yves Robert, Francois-Henry Rouet, and Bora Ucar. On computing inverse entries of a sparse matrix in an out-of-core environment. *SIAM Journal on Scientific Computing*, 34(4):1975–1999, July 2012.

[10] Jens Krüger and Rüdiger Westermann. Linear Algebra Operators for GPU Implementation of Numerical Algorithms. *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH 2003*, pages 908–916, July 2003.

[11] Stanimire Tomov, Rajib Nath, Hatem Ltaief, and Jack Dongarra. Dense Linear Algebra Solvers for Multicore with GPU Accelerators. *IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum*, pages 1–8, April 2010.

[12] The University of Tennessee. *MAGMA User Guide*, 1.7.0 edition, September 2015.

[13] NVIDIA Corporation. *cuBLAS Library User Guide*, 7.5 edition, September 2015.

[14] Franz Färber, Sang Kyun Cha, Jürgen Primsch, Christof Bornhövd, Stefan Sigg, and Wolfgang Lehner. SAP HANA Database - Data Management for Modern Business Applications. *ACM Sigmod Record*, 40(4):45–51, December 2011.

[15] NVIDIA Corporation. *CULA Tools: GPU-Accelerated Libraries*, October 2015.

[16] Barbara Chapman, Gabriele Jost, and Ruud van der Pas. *Using OpenMP: Portable Shared Memory Parallel Programming*. The MIT Press, December 2007.

[17] Intel Corporation. *Intel Architecture Instruction Set Extensions Programming Reference*. Intel Corporation, August 2015.

[18] Intel Corporation. *Intel Math Kernel Library Reference Manual - C*. Intel Corporation, 11.3 edition, August 2015.

ActOnAir: Data Mining and Forecasting for the Personal Guidance of Asthma Patients

Matthias Scholz
Hochschule Mainz
Lucy-Hillebrand-Straße 2
55128 Mainz
matthias.scholz@hs-mainz.de

Nikolai Bock
Hochschule Mainz
Lucy-Hillebrand-Straße 2
55128 Mainz
nikolai.bock@hs-mainz.de

Gunther Piller
Hochschule Mainz
Lucy-Hillebrand-Straße 2
55128 Mainz
gunther.piller@hs-mainz.de

Klaus Böhm
Health & media
Dolivostraße 9
64293 Darmstadt
klaus.böhm@health-media.de

Abstract

Goal of the project ActOnAir is a novel approach for the capture, combination and analysis of bio signals and environmental data, to provide personal guidance for individuals, who suffer from asthma and need to reduce their exposure to air pollutants. This contribution outlines the corresponding system architecture and discusses first results from a prototype implementation. Central part is the novel usage of data mining and forecast algorithms, which is implemented on the in-memory platform SAP HANA.

1 Introduction and Motivation

Air pollution is a major health risk. Estimates from the World Health Organization reveal that approximately one in eight of total global deaths result from the exposure to air pollutants. In particular people with asthma are affected by worsening air quality. Approximately 235 million persons suffer from asthma with more than 2 million deaths every year [1]. Although there is plenty of research on the influence of environmental factors upon the appearance of asthma symptoms (see e.g. [2]), detailed personal guidance for everyday life is hard to provide. Reasons are:

- A comprehensive and fine granular capture of the symptoms of individuals and their exposure to environmental factors is missing. Many pollutants are only measured in a few places and difficult to relate to individual persons.
- Individual analyses and situational predictions are absent. The dependence of asthma attacks on pollution thresholds, environmental aspects and individual dispositions are not considered in a holistic way.

In the near future available information on personal bio signals and the environment will strongly increase due to the growing number of mobile and stationary sensors. Therefore, it is all the more important to develop methods and tools for a reliable provisioning of situation specific, individual guidance.

The situation described above shall be improved significantly though a new hard and software system "ActOnAir". A corresponding project has been started early 2015. This paper summarizes current challenges, the system architecture of the new solution as well as first implementation results.

2 State of the Art

Since around 10 years local and fine granular measurements of air pollutants are topic of a growing number of projects, e.g. CitiSense [3] or Copenhagen Wheel [4]. On the other hand, new concepts and trends, like People as Sensors [5] and Quantified Self, yield more and more information on personal wellbeing. Pre-requisite of a comprehensive analysis of environmental and personal data is their meaningful integration. Due to the inherent heterogeneity of data sources this is often a challenge. The ActOnAir

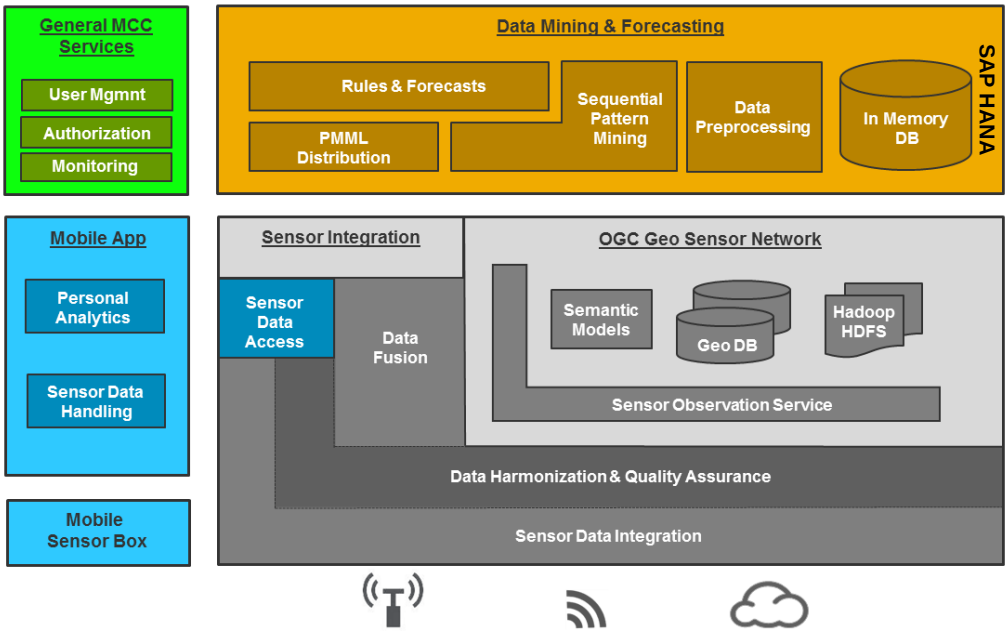


Figure 1: ActOnAir system architecture

system addresses this key issue through extendable integration services.

Data mining of sensor information becomes increasingly important for personalized analyses in health care. The development of methods which cover all potentially relevant data and ensure an appropriate interpretation and further processing of results, is high on the agenda [6]. A good starting point for asthma monitoring is the work of Lee et al. [7]. Still missing are methods for a systematic improvement of personalized results, as well as for the provisioning of real-time guidance. ActOnAir addresses both issues.

3 System Architecture

A suitable system architecture to overcome the challenges mentioned above, is shown in Figure 1. It consists of five main building blocks:

Mobile Sensor Box: To measure the individual exposure of persons to air pollutants, the ActOnAir system provides a wearable box with a set of sensors. Amongst others it contains measuring units for particulate matter emissions PM10 and PM2.5, for ozone, temperature and relative humidity. Captured data are transmitted via Bluetooth to the mobile applications of end users and from there to the Sensor Integration module.

Sensor Integration and Geo Sensor Network: This component is responsible for the processing of heterogeneous sensor data. The module Sensor Integration provides services for a flexible plug-and-play integration and combination of data from different sources. Harmonized sensor data are persisted within the Geo Sensor Network. It is based on standards of the Sensor Web Enablement from the Open Geospatial Consortium (OGC), like the Sensor Observation Services (SOS) and SensorML [8].

The component Data Fusion enables the combination of sensor data, personal bio signals and generally available environmental information. For this purpose all data are projected onto an appropriate spatial-temporal reference.

Data Mining and Forecasting: This building block receives harmonized and interpolated sensor data from the Data Fusion component. During preprocessing, data are assigned to suitable patient segments. Data mining analyses then yield frequent sequential patterns for days with asthma attacks, as well as for times without discomfort. These patterns serve as input for the derivation of rules for forecasts. Corresponding results are finally provisioned to end user applications in the form of PMML documents.

The system shall be able to automatically identify potential improvements for individual forecasts, e.g. based on newly measured sensor data. It may then appropriately optimize patient segments, calculate new rules and distribute them accordingly.

Mobile Application: The user or patient interacts through a smartphone application with the ActOnAir system. The mobile app calculates and displays personal forecasts. For this purpose it interprets the rules from the data mining component using current sensor data and personal bio signals.

In addition, the smartphone application enables a regular recording of personal symptoms through an asthma diary. It typically covers information on peak

expiratory flow, cough, wheeze, shortness of breath, chest tightness and further indications. Finally, the smartphone application acts as control unit for all mobile and personal sensors.

General MCC Services: This module provides common mobile cloud computing services for, e.g., user management, authorization as well as communication and application monitoring.

The interplay of these major components of the ActOnAir system is illustrated through a typical data flow: The smartphone app of a patient collects information on her asthma symptoms, as well as sensor data captured by her Mobile Sensor Box. All data are transferred to the component System Integration. They are then technically and semantically harmonized. Finally, they are persisted in the Geo Sensor Network. Here also publicly available environmental data, e.g. from weather and air quality monitoring stations, are stored. In the module Data Fusion general environmental data are extrapolated to appropriate temporal and spatial coordinates and combined with data from individuals.

Next, the completed data sets are provided to Data Mining and Forecasting. After appropriate preprocessing, e.g. segmentation and binning, frequent patterns and rules for forecasting are derived. The rules for forecasting are transferred to the smartphone application of the patient. Here they are used for real-time guidance. Inputs are: actual symptoms, most current sensor data from the Mobile Sensor Box and possibly other wearable devices of the patient, general environmental data with proper temporal and spatial reference. The latter are retrieved on request from interfaces of the component Sensor Data Access.

4 Prototype Implementation

This section summarizes selected details of a prototype implementation of the ActOnAir software system. For the Geo Sensor Network the SOS framework of 52°north [9] is used. It implements the actual OGC SOS standards and appropriate information models for air quality. The SOS framework allows a physical, as well as a virtual integration of data from different sources. Based on current OGC standards [8], the system offers a uniform API for outgoing and incoming data communication, covering also spatio-temporal filter operations. The component Sensor Integration uses SpringXD [10], which builds upon Spring Integration and the messaging patterns of Hohpe and Woolf [11]. The first proof of concept successfully implements adapters for generally available data services for environmental information. These typically use different formats, like CSV, JSON or NetCDF. All basic services for data processing and communication with mobile applications and the component for data mining have been established and tested.

The module Data Mining and Forecasting is built upon the in-memory platform SAP HANA. The performance of SAP HANA is beneficial for the expected high data volumes. In particular, initial explorative analyses, as well as automated systematic improvements of forecasting models require short response times. Also the Predictive Analysis Library (PAL) within the HANA platform is of significant advantage, as ActOnAir needs to use and combine several data mining algorithms in a performant way.

To exploit the possibilities of HANA, the programming model has been chosen appropriately: a denormalized data model is used; table variables are applied as far as possible to avoid write operations; data intensive application logic is largely embedded into the database with SQLScript; stored procedures – e.g. for data pre-processing – are parallelized wherever applicable.

For sequential pattern mining an R implementation of the SPADE algorithm is applied [12]. Frequent patterns for sensor data, environmental information and personal symptoms are then taken as attributes for the computation of decision trees [7]. For this purpose the CART algorithm of PAL is used [13]. The corresponding results are provided to the smartphone applications of end users as PMML document through an OData interface.

This multi-step analysis procedure has been successfully tested with sample data for selected asthma symptoms and generally available environmental information.

5 Next Steps

In the area of sensor data management algorithms for the harmonization and fusion of diverse sensor data will be implemented next. Necessary is also the completion of the mobile application for end users. Furthermore, an authorization concept based on OAuth will be explored for the overall data flow.

For data mining and forecasting test runs with realistic data are scheduled. To fine-tune the applied algorithms, a medical evaluation of results is of particular importance. Potential adjustments include the segmentation of patients, binning of data during preprocessing, pruning of frequent patterns and the simplification of decision trees.

In addition, concepts for an automated optimization of personal predictions will be investigated. Results from a retrospective comparison of predictions with corresponding event data from patients, or significant changes in volume and quality of sensor data could be important triggers.

In Q3 2016 the complete system should be available for final evaluation.

Supported by the Federal Ministry for Economic Affairs and Energy

References

[1] WHO: 7 Million Premature Deaths Annually Linked to Air Pollution. http://www.who.int/mediacentre/news/releases/2014/air-pollution/en/, 2014. Last accessed 17th September 2015

[2] Schachter E. N. et al.: Outdoor air pollution and health effects in urban children with moderate to severe asthma. Air Quality, Atmosphere & Health 2015 (4):1-13, 2015

[3] CitiSense: http://sosa.ucsd.edu/confluence/display/CitiSensePublic/CitiSense, 2010. Last accessed 17th September 2015

[4] CopenhagenWheel: http://senseable.mit.edu/copenhagenwheel, 2014. Last accessed 17th September 2015

[5] Sagl G., Resch B., Blaschke T.: Contextual Sensing: Integrating Contextual Information with Human and Technical Geo Sensor Information for Smart Cities. Sensors 2015 (15): 17013-17035, 2015

[6] Sow D. et al.: Mining of Sensor Data in Health Care: A Survey. In: Aggrawal C. C. (ed.): Managing and Mining Sensor Data. Springer, Heidelberg, 2014

[7] Lee C. H. et al.: A Novel Data Mining Mechanism Considering Bio-Signal and Environmental Data with Applications on Asthma Monitoring. Computer Methods and Programs in Biomedicine 2011, 101 (1): 44-61, 2011

[8] OGC: OGC Standards and Supporting Documents. http://www.opengeospatial.org/standards, 2015. Last accessed 17th September 2015

[9] 52n SOS: Sensor Web Community. http://52north.org/communities/sensorweb/index.html, 2015. Last accessed 17th September 2015

[10] Spring XD: http://projects.spring.io/spring-xd, 2015. Last accessed 17th September 2015

[11] Hohpe G., Woolf B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, Boston, 2003

[12] Buchta C., Hahsler M., Diaz D.: Package 'arulesSequences'. https://cran.r-project.org/web/packages/arulesSequences/arulesSequences.pdf, 2015. Last accessed 17th September 2015

[13] SAP PAL: SAP HANA Predictive Analysis Library. http://help.sap.com/hana/sap_hana_predictive_analysis_library_pal_en.pdf, 2015. Last accessed 17th September 2015

BICE: A Cloud-based Business Intelligence System

Oliver Norkus, Brian Clark, Björn Friedrich, Babak Izadpanah,
Florian Merkel, Ilias Schweer, Alexander Zimak, Jürgen Sauer
Carl von Ossietzky Universität Oldenburg
Escherweg 2
26121 Oldenburg
{firstname.surname}@uni-oldenburg.de

Abstract

At the University of Oldenburg, a research project is working on the standardization of business intelligence in the cloud. A reference architecture for BI-Cloud systems and services was defined and elaborated. In this paper we introduce our cloud–based BI–system which was build up on the base of this reference architecture. We report from the implementation of the prototype under the use of hardware provided by the Hasso-Plattner-Institute (HPI). Furthermore we present our use-case and discuss our scenario-based evaluation.

1 Introduction

The combination of business intelligence (BI) and cloud computing (CC) is discussed increasingly in science and industry. The absence of standards and transparency encourages many organizations skepticism and incomprehension. The wide use in width remains off. First products of various tool manufacturers exist on the market - but they are heterogeneous and partly in an early maturity. They can not be combined in any way, or integrate with existing systems [2], [5].
This is because, among other things, that some fundamental issues regarding the enterprise and software architecture have not been clarified yet. This underlines the potential of standards [1].
Therefore, in our research the standardization of BI in the cloud is addressed. Thereby, the aim is to increase transparency and to promote the standardization of BI in the cloud. Operationalized, the objective is to define and elaborate artifacts for the description and analysis, comparison, evaluation and uniform design of cloud-based BI systems and services.

For this reason we have designed an integrated architectural model for BI in the cloud. This consists of two artifacts:

- First, the taxonomy for describing and comparing BI-Cloud services [4] and
- secondly, the architecture for realizing, for comparison and evaluation of BI-Cloud systems [5].

To determine the feasibility we realized a prototype based on this architecture and to evaluate the concepts we applied the architecture and the prototype in a case study. In this contribution we introduce our cloud–based BI–system which was build up on the base of the reference architecture. Moreover, we report from the implementation of the prototype under the use of hardware provided by the Hasso-Plattner-Institute (HPI). And at least, we present the results of our case study, discuss our scenario-based evaluation and describe further work. But first, let's short explain our understanding of *BI in the cloud*.
BI in the cloud is a disruptive technology. It has precursor and roots in both domains BI and CC. BI in the cloud offers new technology combinations to provide individual and differentiated configurable, scalable and flexible analytical services. Analytical applications, services and systems can be deployed by using CC. In general, the resulting IT services will be summarized as Business Intelligence as a Service (BIaaS) [5]. More specifically, it can differ between [4]:

- BI-Software as a Service: Visualization (e.g., Report, Dashboard, Scorecards) or Modell (e.g., self service BI, data mining) as a BI-Cloud service
- BI-Platform as a Service: Data Warehouse (DWH) platform as a cloud platform for developing and configuration of BI-Cloud software services.
- BI-Infrastructure as a Service: Storage and computing components for the infrastructural provision of BI-Cloud services.

The rest of this paper is structured as follows: In the next section, see 2, we introduce our case study and therein requirements and business drivers..

2 Use case

The considered use case originates from energy domain in the field of electricity trading. To identify the requirements and business drivers we performed interviews with expert from the energy industry INF-Paper. For the sake of brevity, the results are summarized concentrates in the following.
Energy suppliers are chiefly interested in the costs a customer needs to pay per kilowatt–hour so that they realize a profit. In the initial situation, this calculation is handcrafted with the help of spreadsheet software. This calculation is named breakeven analysis or contribution margin control and the result is the cumulative cost. The calculation of the cumulative costs is needed when energy suppliers make offerings to prospective customers. The cumulative cost per kilowatt–hour is the important part of any offering. If a contract between a customer and an energy supplier is up for extension, the current cumulative cost must be analyzed to calculate the cost for the new offer. The process of calculation lasts up to a few days and has to be completed before the sales pitch takes place. This is disadvantageous for the field managers. When they visit a customer site for a customer pitch, it would be impossible for them to calculate a new value of the cumulative cost when necessary. They have to return to the office and find a new appointment with the customer. This is time–consuming for both sides [3].

The use case yields to the following requirements [3]:

- **Availability** System redundancy is required to guarantee high availability.
- **Accessibility** Field managers are often on site for customer pitches. Thus, the service should be available on a wide range of devices and must be available over the Internet.
- **Accounting** Given that BI systems are expensive to set up and maintain, it is necessary for clients to pay for their usage. The charges can be calculated in respect to the usage.
- **Performance** The duration of calculation, and data transport to the user should be minimized. Poor performance leads to decreased user acceptance of BI systems. While the quality of mobile Internet connections an unavoidable factor, keeping transmitted data small and presenting overview before detail helps to mitigate this.
- **Modularity** There are multiple processes related to the calculation of analysis. There is the potential for modifications or integrations of further analysis. Modularity enables the effective management of the system.
- **Expandability** There are multiple processes related to the calculation of analysis. There is the potential for modifications or integrations of further analysis. Modularity enables the effective management of the system.
- **User empowerment** There are multiple processes related to the calculation of analysis. There is the potential for modifications or integrations of further analysis. Modularity enables the effective management of the system.
- **Web standards** Nearly every mobile device or computer supports web technologies like HTML and JavaScript. The best way to bring the service to a wide variety of devices is to use the accepted standards.

The requirements *availability* and *accessibility* lead to the two more computer science related requirements *scalability* and *flexibility*. The hardware resources have to be provisioned as required. If one server is overburdened, another server must be started to take the load away from the overburdened server. Through resource scaling, the service's full quality is available to every user.
In the next section, we present our conceptual approach and our prototype for a cloud–based contribution margin dashboard.

2.1 Prototype

As a feasibility study of the reference architecture, we have created a prototype covering the requirements of the case study outlined above (see 2). In order to satisfy the identified requirements and to follow the reference we implemented a BI-Cloud system, providing a contribution margin dashboard for electricity trading.
The complete architectural approach for the use case regardless of the technology can be found in [3], the generic architecture is summarized [5]. The following sections only discuss the technological architecture of the implementation. The developed prototype is based on state of the art technologies provided by the Hasso-Plattner-Institute (HPI). The virtual machines (VM) required for running the components are running and managed using HP Converged Cloud, which adheres to the Open Stack architecture. The in-memory database system is a shared SAP HAHA instance. In figure 1 the realized architecture of the prototype is shown.
For the foundation infrastructure, BICE uses an SAP HANA instance for a Database, as well as a HP Converged Cloud blade for computing hardware. An Open Stack platform has been installed on the HP Converged

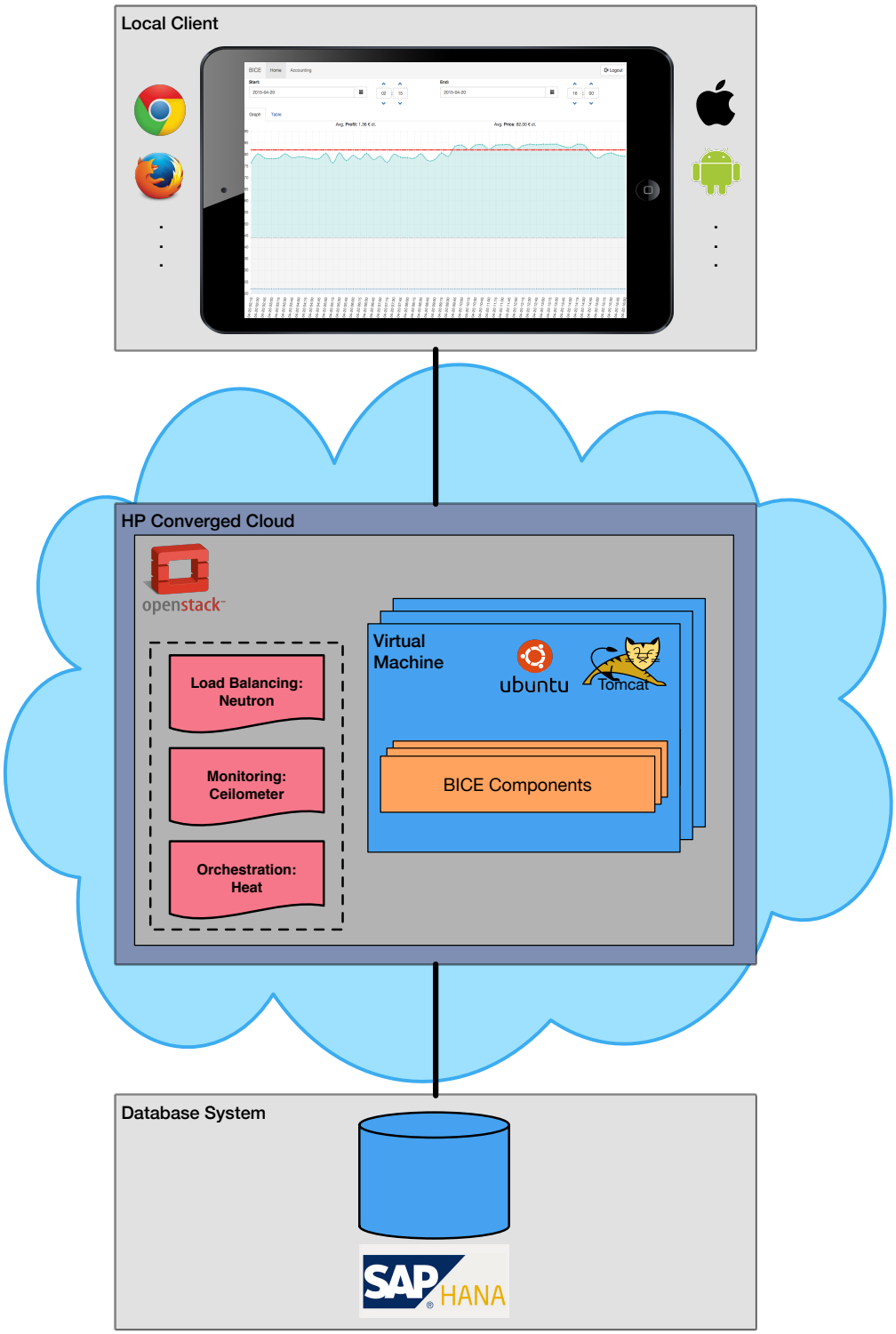


Figure 1: Architecture of the prototype

Cloud, to provide virtual machines while handling orchestration, scaling, and load balancing. The individual software components of the BICE system run inside these virtual machines using Ubuntu/Tomcat.
All relevant data (consumption, energy, labor, and network usage costs) are provided on the SAP HANA database system. The salesperson has the opportunity to view the contribution margin and the total cost of any customer site in tabular or graphical form, taking into account the desired filter settings, such as time range and interval.

3 Evaluation

As evaluation methods were the scenario-based and expert-based evaluation used. For testing were extensive pseudonymous test data provided by our industry partner from the German energy sector. During the design and development early feedback from the experts to the different stages of development has been obtained. For final testing the experts have assumed the role of the users. In the form of cognitive walk through the use case the system was fully tested. The experts independently verified the functionality and usability. It was found, and for the sake of brevity here just summarized, that the functionality of the prototype fulfilled the requirements. Furthermore, the reference architecture has made very good use as an implementation based. With the use of the architectural model as an implementation template risks were minimized and the efficiency of the development process has been increased. The project team, especially the architects and developers, were supported by the realization and development.

4 Conclusion and further challenges

Based on the reference architecture, a prototype was developed in order to show them the basic feasibility. This prototype has been used and applied in the context of a case study. Within this case study and a scenario-based evaluation the results were tested and rated. In general, BI in the cloud appears to be an adequate technological answer for the questions from the business in the area of agility, flexibility and cost efficiency. The realized prototype fulfilled the requirements of the case study as expert- and scenario-based evaluations showed. The deployment templates from the reference architecture were well suited to implement a cloud-based BI system.
Since the concentration was on the feasibility and functionality, of course, the issue of security, especially in the cloud context, experience a consideration. The expansion of the reference architecture towards a comparison and assessment model for BI in the cloud is another next step. The suitability as deployment basis is shown. The distribution of the integrated reference architecture is a long-term step.

Regarding the overall goal of standardization BI in the cloud, we consider our work at this point as an important step in the direction of unification of the architecture of cloud-based BI systems.

References

[1] O. Norkus. An approach for the standardization of business intelligence in the cloud. In U. Aßmann, B. Demuth, T. Spitta, G. Püschel, and R. Kaiser, editors, *Proceedings of Software Engineering & Management*, number 239 in LNI, pages 299–303. Bonner Köllen Verlag, 03 2015.

[2] O. Norkus and H.-J. Appelrath. Towards a business intelligence cloud. In *Proceedings of the Third International Conference on Informatics Engineering and Information Science (ICIEIS2014)*, pages 55–66. SDIWC, 09 2014.

[3] O. Norkus, B. Clark, F. Merkel, B. Friedrich, J. Sauer, and H.-J. Appelrath. An approach for a cloud-based contribution margin dashboard in the field of electricity trading. In D. Cunningham, P. Hofstedt, K. Meer, and I. Schmitt, editors, *Informatik 2015*. Bonner Köllen Verlag.

[4] O. Norkus and J. Sauer. A taxonomy for describing bi cloud services. In *Proceedings of the International Conference on Semantic Web Business and Innovation*, pages 1–12. SDIWC, 2015.

[5] O. Norkus and J. Sauer. Towards an architecture of bi in the cloud. In G. Silaghi, J. Altmann, and O. Rana, editors, *Economie of Grids, Clouds, Systems and Services (GECON2015)*. Springer, 09 2015.

Project Report on "Large-Scale Graph-Databases based on Graph Transformations and Multi-Core Architectures"

Hasso Plattner Institute

Prof.-Dr.-Helmert-Strae 2-3
14482 Potsdam, Germany

Matthias Barkowsky
Matthias.Barkowsky@student.hpi.uni-potsdam.de

Henriette Dinger
Henriette.Dinger@student.hpi.uni-potsdam.de

Lukas Faber
Lukas.Faber@student.hpi.uni-potsdam.de

Felix Montenegro
Felix.Montenegro-Retana@student.hpi.uni-potsdam.de

Abstract

Updating large scale graph data interactively accessed by multiple users applying graph transformations requires a high throughput. This can be achieved by concurrently executing multiple queries. In order to obtain consistency of the graph data, different synchronisation strategies for graph transformations were prototypically implemented and evaluated using the Future Soc Lab. This project report presents the conceptual basics and results of the project.

1 Introduction

Large scale graph databases keep gaining influence on the web, for example in social networks [2]. Accordingly, the amount of requested updates on the graph data becomes hard to handle with requests potentially incoming faster than they can be processed. To obtain the possibility of interactively querying [8] in a multi-user setting the throughput of queries needs to be increased. This can be achieved using parallelism on multicore architectures. Then, the problem of ensuring consistency in the data needs to be addressed.
Therefore, a framework for graph databases, which allows integration of different synchronisation strategies for synchronising access to graph data, was developed and implemented in the project [1]. In our project, queries are performed as graph transformations [7], which consist of a left side graph, which is matched onto the graph data and replaced by a right side graph. We distinguish two types of queries. On the one hand queries are considered as reading queries if their execution does not have any side effect on the data. In that case their left hand side and their right hand side are equal. On the other hand there are writing queries whose execution does alter the graph data. There are transformation engines like Henshin[1] or Story Diagram Interpreter[2] capable of performing such graph transformations.
Their work consists of the search for matches and replacing them. We distinguish engines in one-stage and two-stage engines. Two-stage engines like Henshin are able to execute a query in two steps: First, the matches are found. This can be done without any side effects. In a second step these matches are replaced. In contrast one-stage engines like Story Diagram Interpreter lack this option only offering an execution in one step.

2 Implementation and Synchronisation

The framework offers adapters for different transformation engines which are operating on graph data

[1] http://www.eclipse.org/henshin/
[2] http://www.hpi.uni-potsdam.de/giese/public/mdelab/mdelab-projects/story-diagram-tools/

generated by the Eclipse Modelling Framework[3] [5]. These enable the usage of said transformation engines for executing queries, allowing users to formulate queries in their transformation language of choice. To assure a consistent state during parallel executions, accesses to the databases must be synchronised. In this project, multiple synchronisation strategies are implemented and evaluated.

2.1 Reference Strategy (GL)

For evaluating the developed strategies, the Global Lock Synchronisation Strategy, which only allows sequential execution, is used as a reference. The strategy allows an execution after acquiring a lock for the entire graph database, so no two queries will be executed simultaneously [6].

2.2 Class Lock Versioning Synchronisation Strategy (CLV)

The Class Lock Versioning Strategy is a synchronisation strategy that uses multiple versions of the stored data to allow reading said data while simultaneously performing writing operations. Therefore, different versions of the same node can exist at the same time without overwriting each other. Instead of directly modifying the data, writing operations create new versions. For reading operations, an appropriate existing version is chosen to execute the operation on. Conflicting versions created by parallel executed writing queries are avoided because each query locks all classes of nodes it targets beforehand.
By only letting reading queries read from versions of the data that are consistent and not modified by parallel executed writing queries, reading and writing queries can be executed in parallel without the risk of reading inconsistent data. To ensure consistency of the data stored in the database, the strategy uses mutual exclusion of writing queries with type granularity. Thus, only writing queries that operate on differently typed nodes are allowed to be executed in parallel [3].

2.3 Two Phase Node Locking Synchronisation Strategy (TPNL)

TwoPhaseNodeLock is a synchronisation strategy that seperates reading and writing accesses on the graph in reading and writing phases. Thus, reading accesses can run simultaneously in reading phases since writing accesses must wait for a writing phase. A phase ends when no more reading or writing accesses are present. Writing accesses are allowed to run simultaneously in a writing phase but must additionally aqcuire a lock on a node befor changing it in order to preserve consistency. For preventing deadlocks, queries must sort requestet nodes befor acquiring locks on them. Therefore, these nodes must have been identified by the query in a previous reading phase [9].

[3] http://www.eclipse.org/modeling/emf/

2.4 Version Node Lock Synchronisation Strategy (VNL)

VersionNodeLock combines the approaches of locking nodes and versioning. Queries start reading on an invariable version of the graph. Thus, reading accesses do not need to be synchronised. Results are written on copies of the nodes. Queries must sort and lock nodes before committing written versions of them [9].

3 Work on the Future SOC Lab

3.1 Used Future SOC Lab Resources

All experiments were executed on a Fujitsu RX600S5 - 2 architecture with 4 Xeon X7550 processors, each having eight cores with 2GHz, providing 1024 GB of RAM and running an OpenJDK Runtime Environment (IcedTea 2.5.1) in Java version 1.7.0_65 and Ubuntu 14.04.01 LTS.

3.2 Setting

We conducted our experiments on a graph resembling interconnected pages with page ranks. Each page links other pages and is linked by other pages. Additionally each page has a page rank dependant of the page rank of the pages that link to this page. The corresponding ecore model is shown in Figure 1.

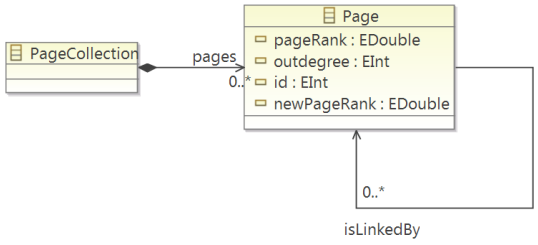


Figure 1: Ecore model for the page graph

In this example, we use a page graph consisting of 1,000,000 nodes and 10,000,000 edges. The edges are uniformly distributed. This means that each node has ten outgoing edges, the number of incoming edges is distributed randomly.
We defined five queries that we use to query the graph. They differ in the actions they perform:

Reading The first type of queries is given an id attribute. It will match on a page with an equating id and will return the id attributes of all pages that link to that page.

Range Reading Unlike the previous query, the second type of queries target a large number of nodes. They are given two decimals parameters. The first parameter describes a lower bound of page rank values and the second page rank describes an upper bound of page rank values. The query will return all page ids of pages whose page rank is within the interval defined by these two bounds.

Updating Updating queries recalculates the page rank for one page. The query identifies the page with a page id parameter.

Creating The fourth type has a page id parameter. All pages with an equating id are matched. The query creates a new page with the same id attribute for each matching page. This query inserts both nodes and edges into the graph.

Deleting The last type of queries matches on pages whose id equals to a given id parameter. Then it will delete all matched pages.

We constructed two benchmarks that use the queries introduced above.

Read Load Benchmark

In a multi-user setting many parallel reading queries are performed on the database. Simultaneously there may be writing queries that access the same data. To allow for an interactive use of the database writing queries need to be executed fast while there are reading queries that would target the same data, what we test in this benchmark.
In this benchmark 16 reading queries are queried in parallel, 11 being simple reading queries and 5 being range reading queries. When any query is finished it is immediately restarted. In addition, 200 writing queries also need to be executed. They are split up into 160 updating, 20 creating and 20 deleting queries. The benchmark will only terminate after all writing queries are successfully terminated.
Figure 2 shows the execution time using different strategies.

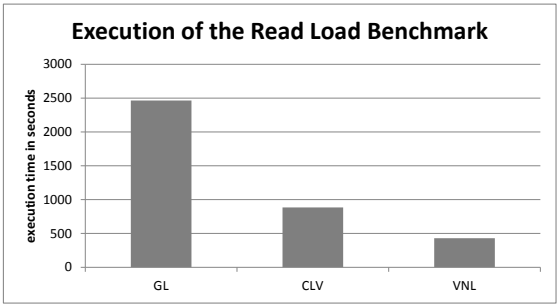


Figure 2: Average execution time of one query using the Read Load Benchmark and different synchronisation strategies

It can be seen that the both version-based strategies require less time to execute the given writing queries. This is due to their ability to execute the reading queries completely in parallel without hindering the writing queries. Being able to parallelize the writing queries, too, the strategy VNL achieves the best performance. While the strategy GL takes more time to execute the updates, it still completes the execution in a moderate time because it strictly obeys the FIFO principle.
The benchmark was also run with the strategy TPNL, but took a considerably longer time (>7200 seconds) to complete because of the reading queries blocking the execution of the writing queries.

Write Benchmark

In a multi-user setting, different types of queries, such as reading, updating, creating or deleting queries, might be queried. This benchmarks tests the performance of strategies when executing a mix of such queries at the same time. It contains 200 reading queries. On the one hand, these reading queries consist of 195 simple reading queries. They are organized in groups of 5 queries. Each groups starts to execute when all queries of the previous group are finished. That way, the reading queries are spread throughout the benchmark. On the other hand, the benchmark contains 5 range reading queries, that are queried at the beginning of the benchmark. Additionally the benchmark contains 300 writing queries with 150 updating queries, 75 creating queries and 75 deleting queries.
Figure 3 shows how much time the queries needed depending on the strategy.

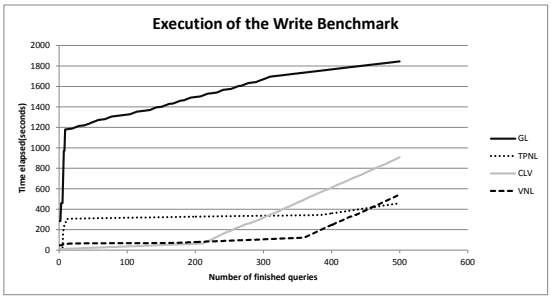


Figure 3: Execution progress of the Write Benchmark using different synchronisation strategies

As indicated, GL does not execute any queries in parallel. This is most significantly reflected that the five long running reading queries are executed sequentially instead of executing them in parallel. Each of these queries need around four minutes to complete. Therefore the graph of GL quickly grows to 1200 in the beginning. The other strategies can execute these queries in parallel. TPNL need these queries to terminate be-

fore writing queries can be executed. The graph remarkably increase in the beginning when the five complex queries need to finish. The version strategies CLV and VNL can execute writing queries in parallel to reading queries. Therefore their execution time grows continuously.
The average time needed to complete further queries notably increases around 200 finished queries for CLV. This strategy can process reading queries in parallel. Thereafter only writing queries are left that all of these strategies can only process sequentially.
Unlike the previous strategies, TPNL and VNL can execute about 350 queries before the average time needed to execute further queries significantly increases. This is because they can execute and updating queries in parallel. The remaining queries mostly consist of creating and deleting queries that can not be executed in parallel with other creating and deleting queries. This results in these queries remaining after most other queries are finished. They are tried to be executed in parallel but each query will invalidate the matches of the other queries causing them to be restarted. Therefore the remaining queries finish sequentially.
As CLV the VNL strategy use the multi-version code. They need more time to sequentially execute queries as TPNL.

4 Conclusion

By looking at our test results, it becomes apparent that a highly parallel execution is desirable and greatly increases the throughput of the developed database. To maximise the amount of concurrently executed queries while retaining a consistent state of the stored data, an efficient synchronisation strategy has to be used. We presented different possible solutions on how to maximise concurrent modifications of graph data. We implemented and evaluated a global lock strategy, a version strategy, a node locking strategy and a combination of version and node locking synchronisation strategy.
The test results showed that TPNL introduces mechanisms to allow for parallel execution of multiple reading queries. CLV introduces a versioning approach that allows for simoultaneous execution of both reading and writing queries. VNL combines the benefits of both approaches.

5 Further Steps

Further steps may include the optimisation of the database, for example by adding the option to create and utilise indices. While this does not increase parallelism when executing multiple queries, the execution of a single query could be sped up, thus increasing overall performance. Some of the implemented strategies do not provide support for graph transformation concepts such as NACs (Negative Application Conditions) and PACs (Positive Application Conditions), which can also be adressed in future work.
Furthermore, since the execution of graph transformations as a two-stage process in this context has significant advantages over a one-stage execution, it might be desirable to implement means to split the execution process of otherwise one-stage transformation engines.

References

[1] Large-scale graph-databases based on graph transformations and multi-core architectures. `http://hpi.de/fileadmin/user_upload/hpi/dokumente/studiendokumente/bachelor/bachelorprojekte/2014_15/BA_Projekt_G1_FG_Giese_Framework_Graphendatenbanken.pdf`. Accessed: 04.08.2015.

[2] R. Angles. A Comparison of Current Graph Database Models. *Proceedings of the 2012 IEEE 28th International Conference*, pages 171–177, April 2012.

[3] M. Barkowsky. Implementing Parallel Execution of Queries over Graph Databases using Versions. Bachelor's Thesis, Hasso Plattner Institute, University of Potsdam, June 2015.

[4] Barkowsky, Matthias, Dinger, Henriette, Faber, Lukas, Montenegro, Felix. Anforderungsdokument. Hasso Plattner Institute, University of Potsdam, 2014. Bachelor's Project.

[5] Barkowsky, Matthias, Dinger, Henriette, Faber, Lukas, Montenegro, Felix. Designdokument. Hasso Plattner Institute, University of Potsdam, 2014. Bachelor's Project.

[6] L. Faber. Development of a Benchmarking Framework and Implementation of Reference Strategies for Evaluation of Synchronisation Strategies for Graph Transformatio. Bachelor's Thesis, Hasso Plattner Institute, University of Potsdam, June 2015.

[7] Habel, Annegret, Pennemann, Karl-Heinz . Correctness of high-level transformation systems relative to nested conditions. *Mathematical Structures in Computer Science*, 19, 2009. Cambridge University Press.

[8] Miller, Robert B. Response time in man-computer conversational transactions. In *Proc. AFIPS Fall Joint Computer Conference*, pages 267–277, 1968.

[9] F. Montenegro. Synchronisation of Graph Transformations by Mutual Exclusion of Overlapping Queries. Bachelor's Thesis, Hasso Plattner Institute, University of Potsdam, June 2015.

Analyzing the Global-Scale Internet Graph at Different Topology Levels: Data Collection and Integration

Benjamin Fabian
Institute of Information Systems
Humboldt University of Berlin
Spandauer Straße 1
10178 Berlin, Germany
bfabian@wiwi.hu-berlin.de

Georg Tilch
Institute of Information Systems
Humboldt University of Berlin
Spandauer Straße 1
10178 Berlin, Germany
tilchgeo@wiwi.hu-berlin.de

Abstract

Traceroute data from global-scale mapping projects can be used to generate comprehensive graphs at different abstraction levels of the Internet. In our project we aim to integrate several large data sets and create a map of the Internet. Moreover, we will conduct graph analyses with respect to identifying important nodes and assess Internet robustness. In the first phase of the project, the data integration and graph creation have been successfully completed. We also conducted first statistical analyses of the resulting Internet graphs.

1 Internet Topology

The purpose of the Internet as a globe-spanning network is to enable connectivity among the billions of connected machines, i.e., each device should be able to communicate with every other device. Reliable information about the Internet topology is crucial to the development of effective routing algorithms, security purposes, robustness analyses, resilience management, and designing countermeasures against global surveillance.

The Internet topology can be represented at different levels, some of which relate to the classical conceptual OSI (Open Systems Interconnection) model. The Internet is often considered at five granularities, illustrated in Figure 1. Those are the IP-interface, router, Point-of-Presence (PoP), Autonomous System (AS), and Internet Service Provider (ISP) levels.

The IP-interface level gives the most fine-grained resolution. Every router has by definition at least two interfaces while backbone routers may have many more. Each interface is assigned with one to many IP addresses. Therefore, each router usually has multiple IP addresses depending on the configuration of that machine. End-hosts, however, typically only have one (dynamically assigned) IP address.

At this level of granularity, each IP address appears as one node in the corresponding graph while an edge refers to a single hop network link (layer 3) between them. This implies that each router appears many times in an IP-interface graph. Interfaces are depicted as solid black dots in Figure 1.

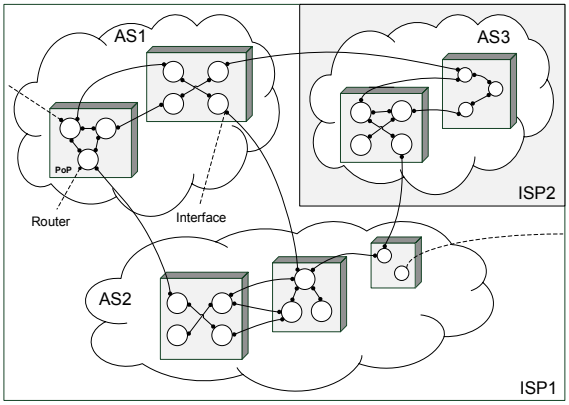


Figure 1: Levels of the Internet Topology

Note that this granularity of topology disregards devices and connections working at lower OSI layers, such as hubs or switches. This ignoring of data link and physical layers comes at the price of some lost information and potential mistakes while inferring actual topologies: For example, machines connected to each other via a switch appear to have a direct edge with one another, while in reality there is one link from each to the switch. Another drawback is the practice of tunneling across different routers, which can significantly distort the inferred topologies.

It is unfortunately not trivial to infer lower layer topologies, which is why network research has mainly concentrated on the IP-interface and higher levels. The largest advantage of this level is that its topology can relatively easy be measured. The *traceroute* tool can be used to infer the connections between two IP addresses; therefore it has become a standard in topology research.

This project aims at developing a method for creating a large integrated graph at the IP-interface level as the basis for subsequent examinations. Such future analyses include the search for bottlenecks and weak points in the entire Internet topology as well as in the

topological connectivity of individual firms and services such as cloud provisioning; so far we have only conducted such investigations at the AS level [9, 11, 12]. Such graph-based analyses can be enriched by business and industry classification [13] as well as geographic and geopolitical information [14].

2 Research Approach

Even though topology discovery has been subject to earlier research [1-4], details of the macroscopic picture still remain unclear. This project aims at advancing the understanding of the Internet topology by integrating empirical data into a multi-leveled graph model. The main emphasis of our research project is placed on both generating and analyzing a combined global-scale Internet graph at different topological abstraction levels (i.e., IP-interface, Point-of-Presence PoP, Autonomous Systems AS).

Different global-scale Internet mapping projects [2-4, 6] provide the raw data for this project. Direct adjacencies of *traceroute* paths called from globally dispersed vantage points yield edges of an IP-interface level graph. In contrast to assessing the datasets individually, our project *combines* different data sources into one large graph in order to capture as much information as possible.

Relying on the results of earlier work [7], the AS-level graph can be generated with IP-to-AS mapping.

Focus will also be placed on the analysis of the PoP-level graph. With the aid of node labels and subgraph analysis, the initial IP-level graph will be further aggregated to the PoP-level. Graph and complex network analyses will be conducted on these graphs. Moreover, using rDNS and geolocation data of the clustered IP addresses, it is possible to infer the geolocation of the particular PoPs as well [4]. The characteristic geospatial information of the PoP graph is used to validate the extracted graph with "top-down" PoP-level maps from ISP information [5].

Earlier works have only concentrated on a single dataset to conduct graph analysis at different topology levels. Our novel approach attempts to combine different datasets and will process them in an integrated and coherent fashion to yield a complimentary and reliable data basis for future research. To the best of our knowledge, this will be the first time that such a comprehensive approach will be followed.

3 Related Publications & Earlier Practical Experiences

The Institute of Information Systems at Humboldt University Berlin has been conducting research in the graph analysis domain for several years. The emphasis in this respect has so far been placed on the AS level. Earlier projects included the generation and analysis of a large AS-level graph extracted from different sources. In particular, robustness analyses and vulnerability assessments of the Internet at AS-level have been conducted [9, 11-13]. Large-scale graph analysis has also been applied on the Bitcoin transaction log ("blockchain") [10] as a representation of a social network. Whereas the calculations of the AS-level graph have been possible with the means of the institute, the Bitcoin network could only be examined on subgraphs. This is due to limited computing capabilities of end-user equipment and university servers in dealing with the massive amounts of generated data. These issues are exacerbated when carrying out actual graph analyses: While some metrics are relatively easy to calculate (e.g., degrees), others such as centrality measures are prohibitively expensive with respect to computation. An additional issue with existing graph databases is that the whole graph data needs to be stored in the main memory for any calculations.

The same issues are expected in the present project: A few tests during the first phase of data acquisition and pre-processing lasted several days even though it was conducted on a capable machine (8-core CPU, 16 GB RAM, 3 TB HDD). Having these experiences in dealing with large-scale graphs, it was very helpful to use the powerful resources of the HPI Future SOC Lab for our project [20] that made the present approach feasible.

4 Project Plan

Our project requires a powerful computational configuration to enable the transformation of the huge amounts of raw data into graphs. Furthermore, a capable graph engine is needed for the topological analyses of the graph-structured data. That is why we plan to exploit the large-scale memory and multicore architecture of the HP Converged Cloud and the newly implemented SAP HANA Graph Engine. The project is structured in four phases. While the first and most of the second phase are already completed, the remaining steps are also planned to be carried out on the resources provided by the HPI Future SOC Lab [20].

The first phase consists of data acquisition and pre-processing. The collected traceroute datasets are examined individually to mitigate any peculiarities. This phase includes cleansing the datasets from anomalies (e.g., anonymous routers), bringing them into a consistent format, and combining them into a unified raw dataset.

The second phase is concerned with extracting the graph at different granularities from the cleansed and

combined raw data. First, the IP-interface-level graph will be spanned by interpreting adjacencies in traceroute paths as edges of the graph. This basic graph will then be augmented with additional information, such as timing, edge weights and rDNS lookups. Secondly, IP-to-AS mapping will be used to generate a traceroute-inferred AS-level graph. Thirdly, the IP-level graph will further be clustered into PoP-level entities, using subgraph analysis and rDNS data. This particular step is computationally expensive. Using information of the clustered IP-level nodes, appropriate measures of geolocating the PoP-level graph will be taken.

The third phase deals with the actual graph analysis of the extracted datasets, which is computationally expensive on such a massive scale. That is why we hope to exploit the promising high performance (OLAP-) graph processing capabilities of the recently implemented SAP HANA Graph Engine. With the help of such computational power, we will probably be able to examine the centrality measures, clustering coefficients, shortest paths, and (strongly) connected components in detail [8]. Those metrics describe important characteristics of the graphs.

The fourth and last phase comprises an evaluation of the PoP-level graph on recently created "top-down" PoP-level maps taken from ISP information. Both the extracted PoP-level graph and the "ground truth" maps inherently show geospatial information, which will enable the comparison of the two points of view.

5 Project Status and Results

The project has successfully achieved major milestones of the first and second phases.

5.1 Phase 1: Data Integration

During the "IPv6 Launch" on June 6, 2012, major ISPs permanently enabled IPv6 for their services [15] and since then, more and more traffic has been routed with the new system. This is the main driver why this work considers data collected during the timeframe of June, 7 – June 20, 2012 as observation period.

Choosing a point at the advent of IPv6 avoids large distortions of IPv4-collected topologies from the new protocol. The risk of wrongly inferred topologies due to infrastructures transferred during the IPv6 Launch is negligible because ISPs start implementing IPv6 with dual-stack implementations rather than with a clear-cut transition towards the new system [16].

Taking the characteristics of IPv6 introduction into consideration, this selection of the sampling period allows obtaining a comprehensive view of the IPv4 Internet without large distortions by the new routing protocol. An overview of the traceroute data sources that have been integrated in our project is given in Table 1.

	iPlane	CAIDA	Carna	DIMES	RIPE Atlas	RIPE IPv6L
Size of raw data	45.9 GiB	86.2 GiB	17.8 GiB	30.7 GiB	20.3 GiB	30.5 GiB
Number of files	2,106	1,154	1	7	35	1
Number of records	264.6 mn.	203.3 mn.	67.0 mn.	21.0 mn.	20.9 mn.	10.3 mn.
Vantage points	299	56	266,604	783	4,780	56
Destination IPs	127,566	195.7 mn.	63.0 mn.	2.3 mn.	39	4,323
Number of traces	112.9 mn.	105.6 mn.	41.8 mn.	15.1 mn.	4.1 mn.	1.8 mn.

Table 1: Integrated Traceroute Data Sources

Whereas each single dataset has some disadvantages, the *combination* of all of them is expected to yield reliable results. Merging data from diverse topology discovery projects could change the individual drawbacks to advantages because the information created from various points and with diverging methods provides a complemented view on the Internet. For the investigation period, there is presumably no additional traceroute data from any large-scale measurement projects available.

A first indication of the quality of the combined dataset can be seen in exploring the conjoint traceroute characteristics: The individual raw data sum to vast **231.4 GiB** and include **587.3 million records** for the duration of the observation period.

The number of unique monitors adds up to 272,505. Most of these stem from the Carna botnet, but the merging of datasets results into an even larger network of vantage points.

Examine Figure 2, where all monitors are plotted on a world map. As one can see, some areas such as Europe, the US East Coast, and the South East of Brazil show a concentration of sources.

Again, the vast majority (97.8%) of vantage points originate in the botnet but it is illustrative to get a visual idea on the complete probing infrastructure. An interesting fact is that there are 73 duplicate monitors, i.e., machines that executed traceroute queries in more than one project.

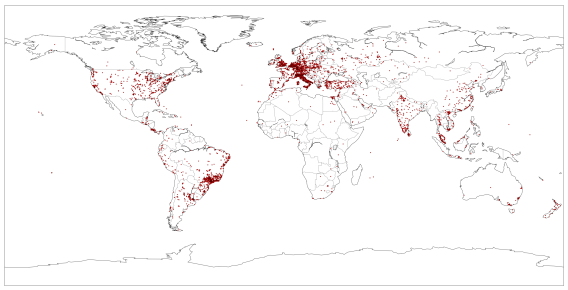


Figure 2: Geographic Location of all Vantage Points

The 257.7 million unique destination IPs in the combined dataset targeted *all* 14.4 million routable /24 prefixes. This does not only indicate that the number of routable prefixes is accurate but also that the whole Internet in terms of subnets with the size of 256 IPs was queried. In total 40.49% of all traces reached their intended target.

This effect also emerges in Figure 3. The distribution of trace lengths is noticeably an overlap of individual distributions. One can recognize the dominance of few hops and the familiar outlier of two-hop traces (with two or three anonymous routers). There is an increased prevalence of anonymous routers in traces with many hops but generally, the impact of unknown interfaces is rather small: 91.3% of all traces traversed three or less non-responding routers, with almost half of them not seeing even one.

Overall, the **average length** of traceroutes is a small: 13.29 with a standard deviation of 6.04 hops.

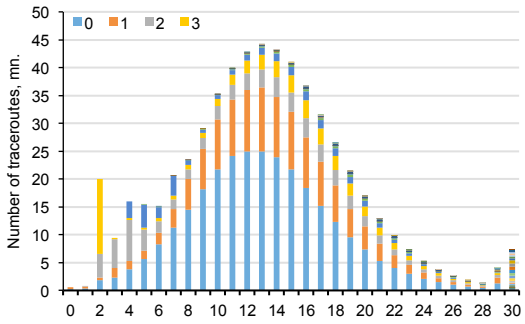


Figure 3: Distribution of Trace Lengths in Hops, Combined Dataset

Conspicuous is, however, the considerably smooth pattern of the combined hop distribution, which demonstrates the underlying principle of this project in an illustrative way: The individual datasets have distortions or peculiarities due to shortcomings in the traceroute implementation, the choice of packets, or the location of sources and targets.

However, anomalies "even out" after superimposing the individual components. Consequently, the features of the combined dataset produce results that are probably closer to reality than any of the individual approaches.

Finally, the acquisition and preprocessing of the data resulted in a combined dataset with **281.5 million** unique traces for the observation period.

To our knowledge this is the largest and most diverse dataset in a traceroute-based topology discovery project so far and it establishes a thorough basis for the following graph extraction and analysis.

5.2 Phase 2: Graph Extraction

The idea of how a graph is extracted from raw traceroute data is the following: adjacencies in traces are interpreted as direct edges of the IP graph. Moreover, the graphs of higher topology levels can be constructed through aggregation. The assumption is that layer-3 information contains knowledge about the higher levels of the Internet; akin to the statistical physics approach that links "microscopic dynamics and interactions […] to the statistical regularities of macroscopic physical systems" [17, Preface, p. *x*].

The source datasets suffer, however, from the shortcoming that none of them is large enough to obtain data about the whole Internet and thus necessarily only captures a subset. This limitation is apparently due to the massive effort needed (in terms of time and money) to establish a sufficiently large measurement infrastructure.

Therefore, this project chooses a different approach for graph extraction. Recall that in the large-scale topology discovery projects "the outcome of many traceroute measurements should be merged" [18, p.5] to extract an IP level graph. Therefore, there is no conceptual break in merging the raw traceroutes from different datasets as well – even if their collection strategies differ. In the end, only direct IP adjacencies of the traceroute records determine how an edge is determined and the combination of raw data from different projects behaves just like the installation of additional monitors and choosing more destinations as targets.

The data from all datasets are processed in exactly the same fashion to alleviate the risk that differences in data aggregation influence the properties of the graphs.

The combination of diverging data thus yields a more comprehensive picture of the actual Internet topology and can alleviate individual measurement biases.

In Table 2, the fundamental statistics of the extracted graphs are shown (for the largest connected components, LCC).

	IP	Router	PoP	AS	ISP
Nodes	3,255,088	2,806,857	53,348	33,752	31,030
Edges	8,544,788	5,039,348	102,591	122,561	113,489
Avg. degree $\langle k \rangle$	5.2501	3.5907	3.8461	7.2624	7.3148

Table 2: Size Metrics of the Graphs (LCC)

5.3 Use of Hardware Resources

The procedures to convert the raw data into graphs include the loading, parsing, filtering, and extracting edges from the massive amounts of data and are resource-demanding. Furthermore, the calculation of graph metrics is computationally highly expensive. That is why there is a need for massive computing power.

The hardware provided by the Future SOC Lab so far included three HP Converged Cloud Blades with 24 x 64-bit CPUs running at a frequency of 1.2 GHz on Ubuntu 14.04. Each of the three machines had 64 GiB memory and was equipped with 1 TiB HDD. Furthermore, 3 TiB were mounted on every machine to store intermediate and final results. This configuration permits an extensive parallelization of tasks. The calculated results would not have been possible without the support of the HPI. For the intense calculations of phase 3, the use of HANA could become crucial.

6 Conclusion

The phases 1 and 2 of the project, data integration and graph extraction, have been completed. In future work, we aim to conduct the further steps of the project, in particular the graph centrality and robustness analyses.

References

[1] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP Topologies With Rocketfuel," *IEEE/ACM Trans. Networking*, vol. 12, no. 1, pp. 2– 16, 2004.

[2] KC Claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov, "Internet Mapping: From Art to Science," in *Proceedings of the Cybersecurity Applications and Technology Conference for Homeland Security (CATCH '09)*, Los Alamitos, CA: IEEE, 2009, pp. 205–211.

[3] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: An Information Plane for Distributed Services," in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI '06)*, Berkeley, CA: USENIX, 2006, pp. 367–380.

[4] D. Feldman and Y. Shavitt, "Automatic Large Scale Generation of Internet PoP Level Maps," in *Proceedings of the 2008 IEEE Global Telecommunications Conference (GLOBECOM 2008)*, Piscataway, NJ: IEEE, 2008, pp. 1–6.

[5] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE J. Select. Areas Commun,* vol. 29, no. 9, pp. 1765–1775, 2011.

[6] Carna Botnet, *Internet Census 2012: Port scanning /0 using insecure embedded devices.* Available: http://internetcensus2012.bitbucket.org/paper.html. Accessed 26 Oct 2015.

[7] Team Cymru, *IP to ASN Mapping.* Available: http://www.team-cymru.org/IP-ASN-mapping.html. Accessed 26 Oct 2015.

[8] M. Rudolf, M. Paradies, C. Bornhövd, and W. Lehner, "The Graph Story of the SAP HANA Database," in *Proceedings of the 15th GI-Conference on Database System in Business, Technology and Web (BTW 2013)*, Bonn: Gesellschaft für Informatik, 2013, pp. 403–420.

[9] A. Baumann and B. Fabian, "How Robust is the Internet? – Insights from Graph Analysis," in *Proceedings of the 9th International Conference on Risks and Security of Internet and Systems (CRiSIS 2014)*, Trento, Italy, Springer, LNCS 8924, 2014.

[10] A. Baumann, B. Fabian, and M. Lischke, "Exploring the Bitcoin Network," in *10th International Conference on Web Information Systems and Technologies (WEBIST 2014)*, 2014, pp. 369–374.

[11] B. Fabian, A. Baumann, and J. Lackner, "Topological Analysis of Cloud Service Connectivity," Computers & Industrial Engineering, vol. 88, pp. 151–165, October 2015.

[12] A. Baumann and B. Fabian, "Vulnerability Against Internet Disruptions – A Graph-based Perspective," Proceedings of the 10th International Conference on Critical Information Infrastructures Security (CRITIS 2015), Berlin, Germany, October 2015, Springer LNCS.

[13] A. Baumann and B. Fabian, "Who Runs the Internet? Classifying Autonomous Systems into Industries," Proceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST), Barcelona, Spain, April 2014.

[14] A. Baumann and B. Fabian, "Towards Measuring the Geographic and Political Resilience of the Internet," International Journal of Networking and Virtual Organisations 12/2013; 13(4):365-384.

[15] Internet Society: *World IPv6 Launch*. Available: http://www.worldipv6launch.org/. Accessed 26 Oct 2015.

[16] Internet Society: *World IPv6 Launch Measurements*. Available: http://www.worldipv6launch.org/measurements/. Accessed 26 Oct 2015.

[17] R. Pastor-Satorras and A. Vespignani, "Evolution and Structure of the Internet: A Statistical Physics Approach," Cambridge University Press, 2004.

[18] R. Motamedi, R. Rejaie, W. Willinger, "A Survey of Techniques for Internet Topology Discovery," IEEE Communications Surveys & Tutorials, Vol. 17 (2), 2014, 1044 – 1065.

[19] Y. Shavitt, E. Shir ,"DIMES: Let the Internet Measure Itself," ACM SIGCOMM Computer Communication Review 35(5): 71–74.

[20] HPI Future SOC Lab. Available: https://hpi.de/forschung/future-soc-lab.html. Accessed 26 Oct 2015.

Comparison of Feature Extraction Approaches for Image Classification

Christian Hentschel
Hasso-Plattner-Institut
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
christian.hentschel@hpi.de

Timur Pratama Wiradarma
Hasso-Plattner-Institut
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
pratama.wiradarma@student.hpi.uni-potsdam.de

Harald Sack
Hasso-Plattner-Institut
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
harald.sack@hpi.de

Abstract

Deep Convolutional Neural Networks (CNN) have recently been shown to outperform previous state of the art approaches for image classification. Recent efforts focus on finetuning CNN models trained on outside data to novel datasets. In this report we investigate to what extent this can be successful when the outside data and the new data to be classified differ.

1 Introduction

Content-based image classification is an important means to address the challenge of search and retrieval in large datasets such as community and stock photo galleries. Typically, manual tagging is impossible due to the large amount of visual information and performance improvements in automatic categorization of visual data have rendered automatic tagging an alternative worth considering.

In this report, we compare two different approaches for image classification – Bag-of-Visual-Words (BoVW) and Convolutional Neural Networks (CNN) – in terms of the obtained classification accuracy. While BoVW has proven to provide good results in the past in recent benchmarking initiatives (cf. e.g. [6]) they have been largely outperformed by CNNs. Their success should mainly be attributed to the fact that different from standard image classification approaches such as BoVW, which use hand-crafted Feature Extractors, CNNs consider the process of feature extraction as part of the model training process. Thus, the derived features usually represent the data much better and therefore lead to more accurate models.

While the idea of using CNNs in image classification is not new (cf. e.g. [4]) only recently the availability of large scale computing power as well as large training datasets manually pooled from web resources such as Facebook or Flickr made CNNs successful. Typically, a CNN model for image classification today is trained on a GPU[1] which offers a high degree of parallelization.

While the requirement of powerful GPU hardware will most likely diminish as an obstacle in future, manually labeling large amounts of training data cannot be easily substituted. For many classification scenarios it is simply impossible to provide a reasonably large amount of training data as it requires manual labeling of images, which is a tedious and time consuming task. Recent efforts have therefore been focusing on reusing CNN models that have been pre-trained on outside data for classification of novel datasets. However, it is still unclear how pre-trained models perform as generic feature extractors especially in cases where the new dataset to be classified differs visually from the dataset used to train the CNN model. In this report we evaluate these new techniques and to compare their performance on visually different datasets to the performance obtained by Bag-of-Visual-Words approaches.

Based on our experience in the preceding Future SOC Lab project [7] where we successfully built an infrastructure that allowed us to train CNN models using different dataset sizes and evaluate their individual accuracy on the ImageNet dataset [6] we reuse these individual models as feature extractors and fine-tune them in order to classify different datasets. As this requires several training and testing runs, we largely benefited from the Future SOC resources in order to conduct our experiments in reasonable time. An own imple-

[1] Nvidia created the parallel computing platform and programming model called CUDA (Compute Unified Device Architecture, [1]) meant to be a general purpose architecture not limited to computer graphics.

mentation of BoVW extraction and model training is available and has been tuned to exploit modern multi-processor architectures such as provided by the Future SOC Lab. For training our own CNN models we rely on existing implementations [2] which have successfully been adapted to the FutureSOC Lab environment in the preceding project. These implementations improve computation speed by exploiting modern GPU hardware (such as the Nvidia Tesla architecture provided in Future SOC) in order to speed up the model training processes.

2 Experimental Setup

The dataset provided by the "WikiArt.org – Encyclopedia of fine arts" project[2] contains images that have visual characteristics different from the object and scene categories in the ImageNet dataset, which we used to train a Convolutional Neural Network. The Wikipaintings dataset[3] is abed on the "WikiArt.org" project and contains a collection of paintings from different epochs, ranging from Renaissance to Modern Art movements. Figure 1 shows example images for some classes. As one can see, the images exhibit visual characteristics different from real-world object images, with strong strokes and unique color compositions.

All paintings are manually labeled according to the respective art epoch. We used the dataset to generate groundtruth data by selecting 1000 images for training and 50 images for testing. Since the number of images in the Wikipaintings collection varies a lot (e.g., for some categories less than 100 images are available while others provide more than 10,000 images), only the classes that have at least 1050 images were chosen from the entire classes (i.e., 22 classes). Based on these classes, we generated training subsets by randomly selecting 5, 10, 20, 40, 60, 80 and 100% of the entire training data (while keeping the test data fixed). We tested three different approaches to classify Wikipaintings test images using the different training set splits: Following our findings in [7] we trained Support Vector Machine classifiers based on Improved Fisher Vector (IFV) image encodings (see [5]). Furthermore we trained CNNs using only the available training data. Finally, we used the best performing model from our previous experiments on the ImageNet dataset (using all training data from all classes) and fine-tuned that model to the Wikipaintings dataset.

2.1 Fine-tuning CNNs

In our experiments, the CNN architecture proposed by Krizhevsky, et al in the ImageNet 2012 competition was chosen [3]. A replication of the model architecture is provided by the Caffe framework [2], with small modifications in the order of pooling and normalization steps (i.e., pooling is applied before normalization). This setting helped to speed up the forward run without reducing accuracies.

In order to fine-tune a pre-trained CNN model to a new target dataset, the last layer (i.e., the 3th fully-connected layer in the architecture we employed) is replaced with the new target outputs – 22 outputs for the Wikipaintings scenario. Additionally, the learning rate on the last layer is multiplied by a factor of 10. Moreover, based on some experimental results, pre-trained models converge faster if using smaller epochs (i.e., 30 epochs). The reminder of the training parameters – such as the initial learning rate, momentum or batch sizes – is the same as when training the entire network from scratch (i.e. without using outside data).

3 Comparison of Fine-tuned CNNs and IFV models

In order to evaluate the results achieved by the different approaches, we computed the mean average precision scores (MAP, averaged over all classes) and plotted them as a function of the training dataset size used to train the model (see Fig. 2).

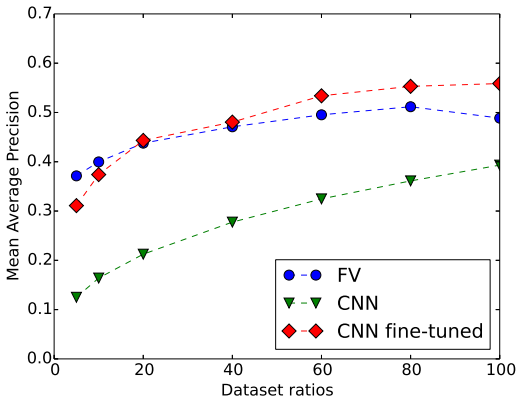


Figure 2: A comparison of IFV, CNN (trained from scratch) and CNN (pre-trained using ImageNet data) on the Wikipaintings dataset. Reported scores are MAP scores as a function of increasing trainingset sizes.

The figure shows that, by adding more positive sample images per class the classification performance increased for all approaches. The best performing results is achieved by the pre-trained CNN model, whereas CNNs trained from scratch performed worse than SVM models using IFV encodings.

In scenarios with very small amount of training data available (e.g. considering 5% – 10% of the original training data, resulting in 50 – 100 images per

[2] WikiArt.org – Encyclopedia of fine arts, http://www.wikiart.org

[3] A list of image URLs can be obtained from http://sergeykarayev.com/vislab/datasets.html

Figure 1: Example images from Wikipaintings dataset.

class) the models based on IFV outperform both CNN-based approaches with the biggest difference at 5% (about 6% difference in achieved MAP score). By incrementing the number of training images, the CNN model based on outside data improves and outperforms the IFV based approach when using about 20% of the available training data. The highest score (i.e., MAP=55.9%) is achieved by the pre-trained CNN model trained on the entire training set. Moreover, while the IFV models converged at 80% and even dropped at 100% (from 51.2% to 48.9% MAP score) both CNN approaches improve with increasing number of training images without showing any sign of saturation. A similar finding with IFV saturating at around 80% of the entire training data has also observed in previous experiments when comparing CNNs and IFV on ImageNet data (see [7]).

Our findings show that pre-trained CNN models need a certain amount of training images to adopt to a new dataset. Improved Fisher Vector encodings can be a competitive alternative when only limited amounts of data is available. Finally, we observe that the overall achieved average precision scores are significantly lower when comparing to classification of real-world objects and scenes as in ImageNet. Whereas CNNs as well as IFV-based approaches were able to achieve near perfect results (AP=100%) for the most easy categories of the ImageNet dataset none of the classification models could achieve near perfect results for any of available classes. This is an indicator of the Wikipaintings classification scenario being more complicated than ImageNet.

4 Conclusions and Future Work

In this report we have shown an experimental setup for comparison of three state-or-the art image classification approaches: Convolutional Neural Networks trained with and without outside data as well as Improved Fisher Encodings and Support Vector Classifiers. Reported results underline the superior performance of pre-trained CNN models when at least 100 training images are available per category. In scenarios with fewer training data available, IFV have proven to outperform CNN-based approaches by up to 6% MAP. Future work will focus on improving CNN models by adding data augmentation techniques. CNNs have been reported to benefit from simple data manipulation tricks such as rotating, flipping and blurring training images. By that means, the available training data can be increased at no additional labeling cost.

Running our experiments in reasonable amount of time required modern GPU and CPU hardware as provided by the HPI FutureSOC lab. Training CNNs was conducted on two Nvidia Tesla K20X GPUs and IFV based encodings were computed using a highly parallelized implementation running on a 128 core machine using up to 1TB of main memory.

References

[1] NVIDIA, CUDA Parallel Computing Platform. `http://www.nvidia.com/object/cuda_home_new.html`. Accessed: 2015-10-02.

[2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[5] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6314 LNCS(PART 4), 2010.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[7] T. Wiradarma, C. Hentschel, and H. Sack. Comparison of image classification models on varying dataset sizes. Technical report, Hasso-Plattner-Institut, Future SOC Lab report, 2015.

Optimization of Data Mining Ensemble Algorithms on SAP HANA

David Müller, Sabrina Plöger, Christoph M. Friedrich and Christoph Engels
University of Applied Sciences and Arts Dortmund, Department of Computer Science
Emil-Figge-Str. 42, D-44227 Dortmund
david.mueller@fh-dortmund.de, sabrina.ploeger@fh-dortmund.de,
christoph.friedrich@fh-dortmund.de, christoph.engels@fh-dortmund.de

Abstract

Ensemble methods (like random forests, quantile forests, gradient boosting machines and variants) have demonstrated their outstanding behavior in the domain of data mining techniques.

This project focuses on optimization of a self-implemented ensemble method on SAP HANA to combine a powerful environment with a fully developed data mining algorithm.

1 Project Idea

In the first four FSOC Lab periods the University of Applied Sciences and Arts Dortmund successfully addressed the topic *Data Mining on SAP HANA* with their projects *Raising the power of Ensemble Techniques* and *Performance Optimization of Data Mining Ensemble Algorithms on SAP HANA* [11][18]. The initial project idea was to compare different opportunities, which enable the usage of predictive analytical techniques on SAP HANA.

SAP is offering the Predictive Analysis Library (PAL), which contains more than 40 well-known algorithms in the fields of classification analysis, association analysis, data preparation, outlier detection, cluster analysis, time series analysis, link prediction and others [27].

In the first project period very accurate predictions could be achieved by using PAL's decision tree implementation [13]. On the other hand performance problems for certain functions in combination with special datasets occurred as the PAL implementation was relatively new and the high potential of the HANA architecture was not fully exploited [5]. Furthermore, no ensemble methods were part of the comprehensive selection of algorithms offered by PAL, yet [27].

In the second period the project team focused on the implementation of an ensemble method on HANA by using the SAP internal language L [12][19].

As this implementation could not match the expected performance advantages, a new project has been initiated in order to write and implement the random forest algorithm in C++ by using the SAP HANA AFL SDK, to utilize HANA's powerful capabilities for CPU-intensive algorithms [18][23].

The project idea of the last and the resent periods was and is to optimize the self-implemented random forest in C++ [20][21][22].

Why Ensemble Methods?

Predictive statistical data mining has evolved further over the recent years and remains a steady field of active research. The latest research results provide new data mining methods which lead to better results in model identification and behave more robustly especially in the domain of predictive analytics. Most analytic business applications lead to improved financial outcomes directly, for instance demand prediction, fraud detection and churn prediction [1][2][10][14][15][30]. Even small improvements in prediction quality lead to enhanced financial effects. Therefore the application of new sophisticated predictive data mining techniques enables business processes to leverage hidden potentials and should be considered seriously.

Especially for classification tasks ensemble methods (like random forests) show powerful behavior [6][7][28] which includes that

- they exhibit an excellent accuracy,
- they scale up and are parallel by design,
- they are able to handle
    - thousands of variables,
    - many valued categories,
    - extensive missing values,
    - badly unbalanced datasets,
- they give an internal unbiased estimate of test set error as primitives are added to ensemble,
- they can hardly overfit,
- they provide a variable importance and
- they enable an easy approach for outlier detection.

Why SAP HANA?

SAP HANA is a "flexible, data-source-agnostic toolset […] that allows you to hold and analyze massive

volumes of data in real time" [3]. It enhances data processing by sophisticated technologies like Massive Parallel Processing (MPP), in-memory computing, columnar data storage, compression and others [3][16][17][26]. Through this project the powerful capabilities of SAP HANA shall be exploited to gain fast processing of CPU-intensive predictive calculations.

Project Goal and Strategy

The overall project idea is to optimize the C++ random forest implementation on SAP HANA. On the one hand the usage of this method leads to longer runtimes for one special data set in comparison to the PAL C4.5 function. Therefore, the reason has to be identified to optimize the existing code. On the other hand the implemented algorithm is not aligned to a special decision tree specification. Thus, the implementation has to be adjusted to such specification, to make a comparison between this method and other algorithms more valid.

The project consists of the following milestones:

- Optimize source code. Find options to make implementation more reliable and stable.
- Identify specifications of the C4.5 method by Quinlan [24].
- Construct a concept for adjusting the random forest method to the specifications of the C4.5 algorithm.
- Adjust the random forest implementation in C++.
- Compare the C++ method with the PAL C4.5 algorithm, to get valid statements about accuracy and runtime results of the C++ implementation.

2 Used Future SOC Lab Resources

For this project a HANA environment revision nine with the latest PAL distribution and the access to use the HANA AFL SDK is needed.

3 Project Findings and Impacts

Impacts on the project and its results are listed in this chapter, as well as the project findings.

3.1 Adjustment of implementation code for stable application

The random forest implementation is the outcome of an ongoing research project. Tests have shown, that the current version doesn't run stable for all possible input data, yet. Consequently sources of errors are identified and corrected. This comprises specially the prediction method for multiple tree inputs. The prediction function was redesigned and now uses linked tree node objects. This allows not only a more stable runtime behavior but also reduces the prediction runtime.

3.2 Adjustment to the C4.5 by Quinlan

The comparison between the implemented method und the C4.5 specification by Quinlan shows some opportunities for adjusting the current implementation. Main differences are the randomization, error based pruning, windowing and the handling of missing values [24].

In this project period the focus is put on the randomization method. The implementation is adjusted to a random feature selection with a maximum consideration of $log_2(quantity\ attributes)+1$ attributes at each node of a tree.

3.3 Datasets for Testing

Four datasets are picked for the testing, comprising

- Car-Purchase: The goal of this dataset is to predict if the car purchased at an auction is a good or bad buy. The dataset consist of 72,666 observations, 13 discrete and 13 numeric attributes [8].
- Connect4: Predicting if player one wins the game or not. The dataset consist of 67,557 observations and 42 discrete attributes [29].
- Covertype: Predicting the forest cover type from cartographic variables. The dataset consist of 581,012 observations, 44 discrete attributes and 10 numeric attributes [4].
- Pokerhand: Each record is an example of a hand consisting of five playing cards. The class describes the poker hand. The dataset consist of 1,025,010 observations and 10 discrete attributes [9].

Those datasets are heterogeneous in the number of observations, the number of attributes, the distribution of discrete und numeric attributes and the number of distinct values of both, discrete values and the class column. Thus they are providing a foundation for solid and applicable test results.

3.4 Performance

The C++ implementation achieves the shortest runtimes for creating a decision for data sets with just discrete attributes. This covers all tests with the data sets Connect4 and Pokerhand (see appendix 1). The training of a model can be executed up to eight times faster by the usage of the C++ decision tree.

For Covertype and Car-Purchase the PAL C4.5 method is the faster algorithm, even if the results are more similar compared to the results with just discrete data sets. It is important to mention that the PAL method is not aligned to the C4.5 specifications by Quinlan concerning the handling of numeric attributes

[31]. Tests are pending, if this different approach is responsible for better performance results compared to the C++ method.

The C++ decision tree is the fastest method for predicting with the created model for all performed tests (see appendix 2).

The random forest implementation achieves good performance results as well. Despite the application of 50 unpruned trees, the training algorithm is only three to ten times slower than creating only one tree with C++ and the prediction is just two to seven times slower. This results can be attributed to the usage of bagging, randomization and parallelization.

3.5 Prediction Accuracy

The prediction results are satisfying and the implemented algorithm runs reliably for most of the possible input variations. Depending on the dataset, parameters and the selection of test and training data, the prediction accuracy can either be better or worse compared to the PAL C4.5 decision tree. The difference in accuracy between the C++ approach and PAL is very small, as those methods work very similar (see appendix 3).

However, it is important to point out, that the random forest leads to different results. In most cases, this means a better prediction, compared to all decision trees, tested in this period (see appendix 3). For Covertype data, the random forest can increase the accuracy up to nine percent.

4 Final Results / Deliveries

The main contribution of this project is the optimization of the random forest implementation in C++. It could be verified, that the random forest leads to better accuracy results in many test cases.

5 Next steps

There are a lot of opportunities to use the project results for further improvements.

On the one hand, the functionality of the method can be extended. There are still important parts, as the handling of missing values or the opportunity of post pruning, which should be implemented. Furthermore, the PAL C4.5 is using a global discretization approach, which leads to faster processing while reaching similar accuracy results. Consequently, this approach must be examined as a powerful alternative to the current implementation.

On the other hand, the delivered random forest can be used to adapt the algorithm to other predictive models, as for example quantile forests or gradient boosting machines.

6 Conclusion

Optimizations of the ensemble technique are implemented successfully in this project period and all project goals are accomplished. The implementation offers a strong and fast predictive model. Nevertheless, there are still some opportunities to optimize the implementation with respect to performance and accuracy of prediction by application of other programming paradigms and further-developed prediction methods.

7 References

[1] R. E. Banfield; R.E., et. al.: “A Comparison of Decision Tree Ensemble Creation Techniques”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 1 (2007).

[2] S. Benkner, A. Arbona, G. Berti, A. Chiarini, R. Dunlop, G. Engelbrecht, A. F. Frangi, C. M. Friedrich, S. Hanser, P. Hasselmeyer, R. D. Hose, J. Iavindrasana, M. Köhler, L. Lo Iacono, G. Lonsdale, R. Meyer, B. Moore, H. Rajasekaran, P. E. Summers, A. Wöhrer und S. Wood: „@neurIST Infrastructure for Advanced Disease Management through Integration of Heterogeneous Data, Computing, and Complex Processing Services“, DOI:10.1109/TITB.2010.2049268, IEEE Transactions on Information Technology in BioMedicine, 14(6), Pages 1365 - 1377, (2010).

[3] B. Berg, P. Silvia: “SAP HANA An Introduction”, 2nd edition, GalileoPress, Boston (2013).

[4] J. A. Blackard (Colorado State University): Covertype Database, (1998), UCI Machine Learning Repository, URL: http://archive.ics.uci.edu/ml, accessed on 15.10.2014

[5] J.-H. Böse, SAP Innovation Center Potsdam, personal communication, Aug. 2013.

[6] L. Breiman: „RF / tools – A Class of Two-eyed Algorithms“, SIAM Workshop, (2003), URL: http://www.stat.berkeley.edu/~breiman/siamtalk2003.pdf , accessed on 11.03.2014.

[7] L. Breiman: “Random Forests”, (1999), URL: http://www.stat.berkeley.edu/~breiman/random-forests-rev.pdf, accessed on 11.03.2014.

[8] Car-Purchase Dataset, (2011), Kaggle, URL: https://www.kaggle.com/c/DontGetKicked, accessed on 15.10.2014

[9] R. Cattral (Carleton Univerity): Pokerhand Database, (2007), UCI Machine Learning Repository, URL: http://archive.ics.uci.edu/ml, accessed on 15.10.2014

[10] C. Engels: „Basiswissen Business Intelligence.“, W3L Verlag, Witten (2009).

[11] C. Engels, C. Friedrich: „Proposal - Raising the power of Ensemble Techniques“, Proposal to summer 2013 period at the HPI Future Lab, (2013).

[12] C. Engels, C. Friedrich: „Proposal - Follow up & extension activities to *the Raising the power of Ensemble*

Techniques project “, Proposal to winter 2013 period at the HPI Future Lab, (2013).

[13] C. Engels, C. Friedrich, D. Müller: „Report - Raising the power of Ensemble Techniques“, Report to summer 2013 period at the HPI Future Lab, (2013).

[14] C. Engels; W. Konen: „Adaptive Hierarchical Forecasting”. Proceedings of the IEEE-IDACCS 2007 Conference, Dortmund (2007).

[15] J. Friedman: „Computational Statistics & Data Analysis”, Volume 38, Issue 4, 28 February 2002, Pages 367–378, (2002), URL: http://dx.doi.org/10.1016/S0167-9473(01)00065-2, accessed on 11.03.2014.

[16] J. Haun, et al.: “Implementing SAP HANA”, 1st edition, Galileo Press, Boston (2013).

[17] R. Klopp: “Massively Parallel Processing on HANA”, (2013), URL: http://www.saphana.com/community/blogs/blog/2013/04/22/massively-parallel-processing-on-hana, accessed on 11.03.2014.

[18] D. Müller, C. Engels, C. Friedrich: „Proposal - Performance Optimization of Data Mining Ensemble Algorithms on SAP HANA“, Proposal to summer 2014 period at the HPI Future Lab, (2014).

[19] D. Müller, C. Engels, C. Friedrich: „Report - Follow up & extension activities to *the Raising the power of Ensemble Techniques* project“, Report to winter 2013 period at the HPI Future Lab, (2014).

[20] D. Müller, S. Plöger, C. Engels, C. Friedrich: „Proposal - Follow up & extension activities to *Performance Optimization of Data Mining Ensemble Algorithms on SAP HANA* “, Proposal to winter 2014 period at the HPI Future Lab, (2014).

[21] D. Müller, S. Plöger, C. Engels, C. Friedrich: „Proposal - Optimization of Data Mining Ensemble Algorithms on SAP HANA“, Proposal to summer 2015 period at the HPI Fu-ture Lab, (2015).

[22] D. Müller, S. Plöger, C. Engels, C. Friedrich: „Report - Follow up & extension activities to Performance Optimization of Data Mining Ensemble Algorithms on SAP HANA “, Report to winter 2014 period at the HPI Future Lab, (2014).

[23] D. Müller, S. Plöger, C. Engels, C. Friedrich: „Report - Performance Optimization of Data Mining Ensemble Algorithms on SAP HANA “, Report to summer 2014 period at the HPI Future Lab, (2014).

[24] J. Ross Quinlan: „C4.5. Programs for machine learning.“, in: Kaufmann (The Morgan Kaufmann series in machine learning), San Mateo, California (1993).

[25] R Project (2015): What is R? Online verfügbar

[26] SAP AG: “SAP HANA Developer Guide (document version: 1.0 – 27.11.2013, SPS 07)”, (2013), URL: http://help.sap.com/hana/SAP_HANA_Developer_Guide_en.pdf, accessed on 11.03.2014.

[27] SAP AG: “What´s New? SAP HANA SPS 07 - SAP HANA Application Function Library (AFL)”, (2013), URL: http://www.saphana.com/servlet/JiveServlet/download/4267-1-12720/What%C2%B4s%20New%20SAP%20HANA%20SPS%2007%20-%20AFL%20Predictive.pdf, accessed on 11.3.2014

[28] G. Seni, J. Elder: “Ensemble Methods in Data Mining”, Morgan & Claypool, San Rafael, California (2010).

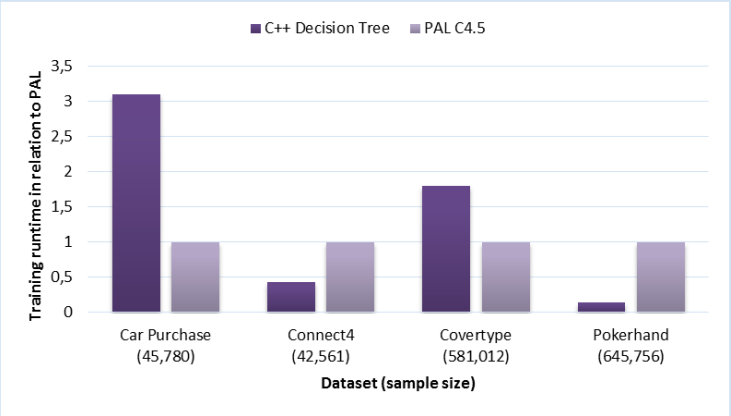
[29] J. Tromp: Connect4 Database, (1995), UCI Machine Learning Repository, URL: http://archive.ics.uci.edu/ml, accessed on 15.10.2014

[30] G. Üstünkar; S. Özögür-Akyüz; G. W. Weber; C. M. Friedrich und Y. A. Son, „Selection of Representative SNP Sets for Genome-Wide Association Studies: A Metaheuristic Approach“, DOI:10.1007/s11590-011-0419-7, Optimization Letters, Volume 6(6), Seite 1207-1218, (2012)

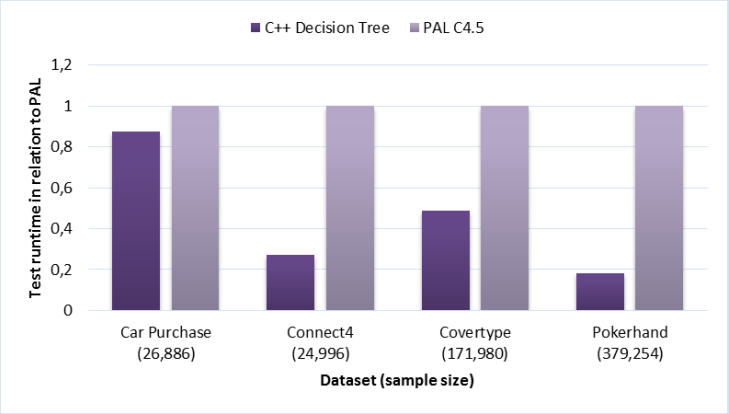
[31] P. Wang: Numeric Atribute Split in Decision Tree c4.5, Mail to Daniel Johannsen, Nov. 2014.

Appendix:

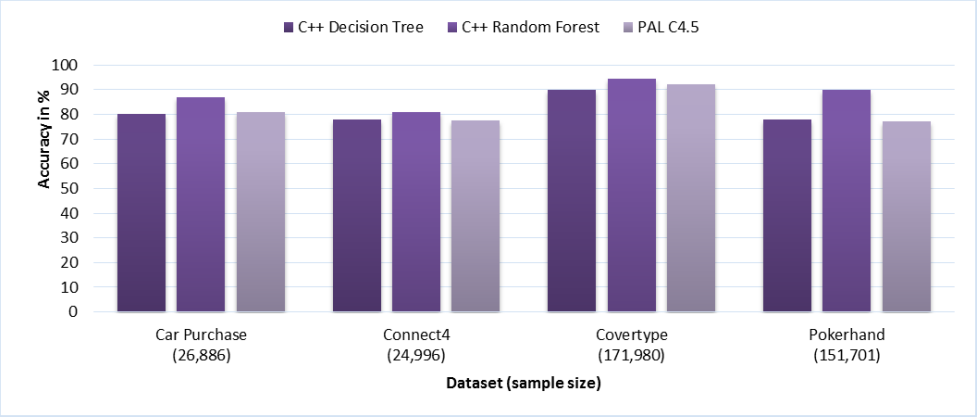
Appendix 1: Training Runtime of C++ Decision Trees and PAL C4.5 in Relation to PAL C4.5



Appendix 2: Test Runtime of C++ Decision Trees and PAL C4.5 in Relation to PAL C4.5



Appendix 3: Accuracy of C++ Decision Trees, C++ Random Forest and PAL C4.5



Simulation of User Behavior on a Security Testbed

Report for the Project "Security Monitoring and Analytics of HPI Future SOC Lab (Phase II) in 2015 Spring"

Aragats Amirkhanyan, Andrey Sapegin, Marian Gawron,
Feng Cheng, Christoph Meinel
Hasso Plattner Institute (HPI), University of Potsdam
D-14482 Prof.-Dr.-Helmert-Str. 2-3
{Aragats.Amirkhanyan, Andrey.Sapegin, Marian.Gawron,
Feng.Cheng, Christoph.Meinel}@hpi.de

Abstract

For testing new methods of network security or new algorithms of security analytics, we need the experimental environments as well as the testing data which are much as possible similar to the real-world data. Therefore, in this technical report, we present our approach of describing user behavior for the simulation tool, which we use to simulate user behavior on Windows-family virtual machines. The proposed approached is applied to our developed simulation tool, which was developed and tested based on the resources of HPI Future SOC Lab. We use the developed simulation tool for solving a problem of the lack of data for research in network security and security analytics areas by generating log dataset that could be used for testing new methods of network security and new algorithms of security analytics.

1 User Behavior States Graph

The *User Behavior States Graph (UBSG)* is the concept to describe user activities for the simulation tool. By other words, it says what we need to do and in which order. Before to learn the concept of the *UBSG*, we need to learn following entities: *State* and *Action*. They are the basement of the framework, because they are used to build the *UBSG* and to describe simulation scenarios.

1.1 State

When you work on the computer, you can say that the virtual machine (VM) has some state and it changes its own state based on your (user) manipulation. The basic example could be the changing the state of your computer when you login to the system. In one particular time, you are logged off, but after entering the credential you change the state of the computer, because now you are logged on. The *State* entity is used to define the specific state of the computer (VM). When we define the state, we should define the name, the parameters that describe the current state and the method that we can use to find out whether this state belongs to some VM.

To define the state parameters, we use the screenshot of the VM. This parameter is called the *Reference Image*. The *logon_hotkey* state is the Windows welcome page state, which you can see after the system is loaded. The *logon_user* is a state in which user is welcomed to enter his username and password. The *desktop* state is a state when user is logged on the system and he can see a desktop view. The *rdp_success* state illustrates a state of the closed successful RDP session.

Now we need to define the method how to check the state of the VM. Our implementation of the simulation tool uses the *virtual network computing (VNC)* protocol [2] to connect to the VM. To support *VNC* on the client side, we use the *vncdotool* [4] library. This library also can take a screenshot of the remote VM desktop. We use this screenshot functionality for checking a state of the virtual machine by comparing the image histogram of the taken screenshot with the predefined reference image in the *State* entity.

1.2 Action

After defining the state, we can decide which action we need to apply. For that, we have the entity called *Action*. The *Action* entity is defined by some commands which are associated with the particular action. The commands could be the typing something on the keyboard or some manipulation with the mouse. In Table 1, you can find the action names, description of the actions and snippet codes for them. Some actions could be parameterized, for example action *A3* requires two parameters: the username and the password.

As we have many states and actions, we want to be

Table 1: Actions and commands.

Action	Commands
A1	['ctrl-alt-del', 'alt-w', 'right', 'right', 'right', 'right', 'enter']
A2	['esc', 'esc']
A3	[':' + args['username'], 'tab', ':' + args['password'], 'enter']
A4	['lsuper-r', ':shutdown /l /f', 'enter']
A5	['cd /', 'powershell rdp.ps1]
A6	[':exit', 'enter']
A7	['lsuper-r', ':shutdown /l /f', 'enter']

A1 - open logon window; A2 - close logon window and return to welcome windows page; A3 - enter username and password and enter into Windows; A4 - log off; A5 - run the RDP session by running script; A6 - close the RDP session; A7 - log off.

able to reuse already defined actions. It means that some actions are not unique, but they are combined by other already defined actions. For example, we could have the action to press *Ctrl* and the action to press *W*. And to define the action *Ctrl+W* (to close the folder explorer), we do not need to define new commands, but we need to say that this new action is combined by other two, which should be undertaken together, but you can also apply them consequently. Once we defined list of the common actions, we could use them to build any new action.

1.3 The states graph

Up to this moment, we got some imagination what are the *State* and *Action*. In this paragraph, we show how to bind them to make the state graph and by this way to describe the *UBSG*.

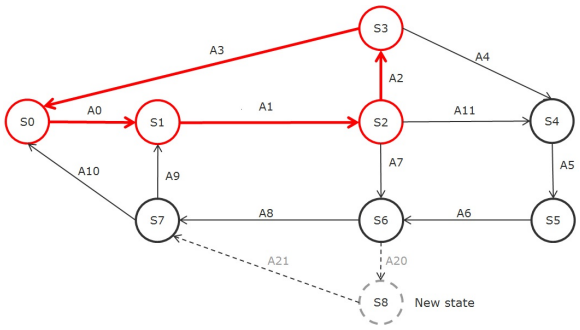


Figure 1: The User Behavior States Graph.

Figure 1 shows the *UBSG*. In this figure, nodes reflect *States* and edges reflect *Actions*. It means that if we have the state *S0* and we want to switch the state from *S0* to *S1*, we need to apply action *A0*. For example, we have the *logon_user* state and we want to change to the *desktop* state, then we need to pass the action that contains the commands of entering the username, the password and pressing the enter (the action *A3*). In the same way, we can switch to any other state in the graph.

1.4 Extending the states graph

Once we defined the graph with states and actions, it could be that we need to extend the state graph by adding new states. To do it, firstly, we need to find out which state we need. Then we need to create new *State* entity with parameters that describe this state. As we mentioned in the previous section, we use the reference image (VM screenshot) to define the state. So, we need to take a screenshot and assign the reference image to the new state.

Once we defined the state, we need to bind new state with the existing state graph. As we said before, states are nodes and actions are edges. So, to bind the new state with the state graph, we need to specify with which existing state in the graph we want to bind our new state and by which actions. We could use existing actions to make connections between states or we could define new one, if we do not have appropriate in the state graph. In Figure 1, you can see the *UBSG* with the extended branch: *S8* state, *A20* and *A21* actions.

1.5 Scenario

The goal of defining the *UBSG* is to use it to define scenarios of user behavior for the simulation tool. The defining the scenario includes the defining the particular path in the *UBSG*. In Figure 1, you can see the example of the scenario covered by the graph. According to the example, the scenario is defined by the path: *(S0,A0) - (S1,A1) - (S2,A2) - (S3,A3) - (S0)*. It is marked in the figure by wide red edges. This path means that we start with the state *S0*, then we compare the current state of the VM with the state *S0*. If states are not equal, then we have to finish the scenario, otherwise we go further and we apply the action *A0* to switch the state from *S0* to *S1*. After that, we check the state *S1* and apply action *A1* and so on until we reach the state *S0*

To have better imagination of the scenario term, we presented the example of the *UBSG*, which we used in the real simulation. Figure 2 describes all possible changes of VM. By this figure, you can see from

which state to which state you can switch and what you should do (which actions to apply) to change the state. The description of the actions you can find in Table 1.

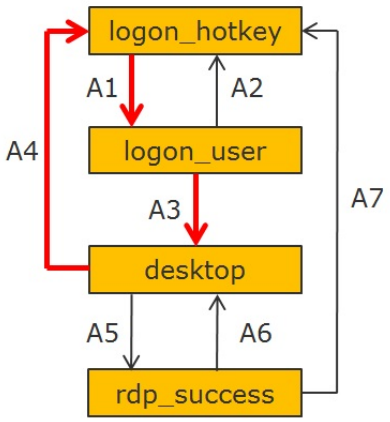


Figure 2: The User Behavior States Graph's example.

In Figure 2, we define the scenario describing the path: *logon_hotkey - logon_user - desktop - logon_hotkey*. According to the scenario and based on the *UBSG*, the simulator finds out that it should apply following actions: *A1, A3, A4*. If you look at Table 1, you can find out that this scenario is pretty simple scenario of user behavior, which is log on the system and then log off from the system. For example, if during the scenario simulation, we want to change the state from *desktop* to *logon_hotkey*, we should pass the action *A4*, which is the *Log off* action according to Table 1, and it includes the follow commands *['lsuper-r', ':shutdown /l /f', 'enter']*. You can use the same way to reveal any needed VM change.

2 Target scenarios

We are interested in using the *UBSG* and the simulation tool to produce data for research in IT security, especially to train Intrusion Detection Systems and test Anomaly Detection Algorithms. To prove our *UBSG* concept, we are aimed to develop the simulation tool based on the *UBSG* and simulate scenarios with normal and abnormal behavior to use them for analyzing attacks. Therefore, we specified the target scenarios, which we want to simulate.

2.1 Typical setup

For the first challenge, we started with designing and describing the test network. Our test network contains the domain controller (DC) with installed Windows Server 2012, the wiki and database (DB) servers with Windows Server 2003 and four client computers with Windows 7 Professional 64-bit. To complete the installation, we created four user accounts: Alice, Bob, Carol and Admin.
The first part of the implementation is the deployment and the configuration of the test network environment. To set up the network, we used a dedicated server on the HPI Future SOC Lab with installed VMware ESXi [3]. The configuration of the dedicated server is as follows: HP ProLiant DL980 G7 with 8X8 CPUs x 2.26 GHz and 2 TB RAM. Firstly, we created the network to isolate our virtual machines from others hosted on the same VMware ESXi server. Afterwards, we deployed and configured all virtual machines as described above.

2.2 Normal scenario

In our configuration, all users have access to client computers, the wiki server and the database server, but only admin has direct access to the domain controller. In the case of the normal scenario, users log on client computers by their credentials and they do it several times per day. Users Bob and Carol during the workday visit the wiki server, but Alice does not visit the wiki server despite that she has access. All users have access to the database server, but only Admin uses it in the normal scenario. In turn, Admin usually logs on his computer and during the workday he logs on the domain controller, the database server and the wiki server by the remote desktop connection (RDP).

2.3 Abnormal scenario

The abnormal scenario is the scenario that differs from the normal scenario by some unusual but acceptable behaviors. In our case, the abnormal scenario includes additional user behaviors. The first abnormal behavior is that the user Alice uses the wiki server. This behavior is abnormal but not suspicious, because other users use it every day. The second abnormal behavior is that the user Bob uses the database server, but in the normal scenario only Admin uses it. This user behavior is more suspicious and could be determined as an attack or malicious behavior.

3 Simulator

3.1 Architecture

The architecture of the experimental setup is illustrated in Figure 3. We implemented the simulation tool (simulator) on the python programming language [1]. To connect from the simulator to virtual machines on the VMware ESXi server, we use the virtual network computing (VNC) protocol [2]. But to use the VNC protocol, we enabled VNC on the ESXi server, specified the VNC port for each client virtual machine and supported the VNC protocol on the application side. To use VNC in the simulator, we used the *vncdotool*

library [4]. This library also can take a screenshot of the remote VM desktop. We used this screenshot functionality for checking a state of the virtual machine by comparing image histograms with predefined reference states (*UBSG* block in Figure 3). Once we determined the current state of the virtual machine, we can specify the activities to be undertaken in each case according to the scenario description, such as sending the *ctrl+alt+delete* command to the virtual machine, entering the username and the password, opening the RDP connection and others. The RDP connection from the client computer to the database and wiki servers is implemented by invoking the PowerShell [6] script hosted on the client computer. This script can accept parameters, such as the IP address, the username, the password and session time, and it means that we can use the script to establish the RDP connection with any server by specifying parameters.

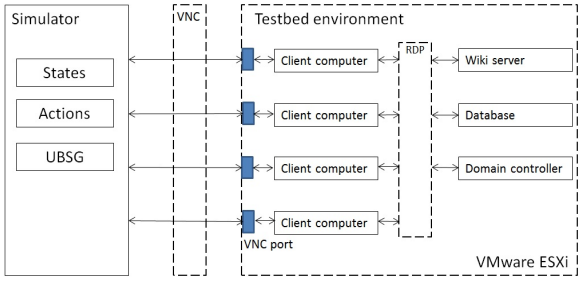


Figure 3: Experimental setup.

3.2 Simulation results

We have successfully used the simulator to simulate simple normal and abnormal scenarios. The normal scenario takes about 4 hours with 46 login and logout events into client computers and 30 RDP connections to the wiki, database and domain controller servers. The abnormal scenario takes about 6 hours with 50 login and logout events to client computers and 39 RDP connections to the wiki, database and domain controller servers. The result of running the simulator is set of real logs in the Windows Active Directory [5] of the domain controller. The domain controller produces a huge amount of log events, but we are interested only in events related to user behavior. For example, we are interested in the event with numbers 4624 (an account was successfully logged on). All the event data generated by the domain controller will then be collected, normalized, and pushed to a 1-TB HANA Instance at HPI Future SOC Lab for further analytics. Thanks to applied approach we have data that are approximately real data. And we can use the simulation tool and the *UBSG* concept to generate more sophisticated scenarios and use generated data for testing analytics approaches and for training intrusion detection systems (IDS).

4 Conclusion

We have presented a new approach for solving a problem of the lack of data for research in the area of network security and security analytics. Our proposal is based on the simulation of synthetic user behavior to generate real dataset. We introduced the concept of the *User Behavior States Graph* for simulation tools. To prove the concept of the approach, we provided the implementation of the idea, and we presented the experimental setup used for the simulation, the description of target scenarios, the architecture of the simulation tool and implementation details. As result, we have successfully used the simulation tool to generate needed data for research. The full description of the work is presented in the paper [7].
As future work, we would like to use the simulator with large amount of users to generate more sophisticated data and test the simulator under load. These experiments imply more resources and additional challenges, such as using the simulator with the network of enterprise or campus level and using several simulators simultaneously in the same network.

Acknowledgment

We would like to thank HPI Future SOC Lab for providing us with the latest and powerful computing resources, which make the testing and experiments specified in the paper possible.

References

[1] Python programming language. `https://www.python.org`. [last access: 19.10.2015].

[2] Virtual Network Computing. `http://www.hep.phy.cam.ac.uk/vnc_docs`. [last access: 19.10.2015].

[3] VMware ESXi. `http://www.vmware.com/products/vsphere-hypervisor`. [last access: 19.10.2015].

[4] Vncdotool. A command line VNC client. `https://github.com/sibson/vncdotool`. [last access: 19.10.2015].

[5] Windows Active Directory. `http://msdn.microsoft.com/en-us/library/bb742424.aspx`. [last access: 19.10.2015].

[6] Windows PowerShell. `http://technet.microsoft.com/en-us/library/bb978526.aspx`. [last access: 19.10.2015].

[7] A. Amirkhanyan, A. Sapegin, M. Gawron, F. Cheng, and C. Meinel. Simulation user behavior on a security testbed using user behavior states graph. In *Proceedings of the 8th International Conference on Security of Information and Networks*, SIN '15, pages 217–223, New York, NY, USA, 2015. ACM.

Protecting minors on social media platforms

Estée van der Walt
Department of Computer Science
Security & Data Science Research Group
University of Pretoria, South Africa
estee.vanderwalt@gmail.com

J.H.P. Eloff
Department of Computer Science
Security & Data Science Research Group
University of Pretoria, South Africa
eloff@cs.up.ac.za

Abstract

Minors are at risk on the internet and specifically social media. Not only are the dangers to them unknown but they are not mature enough to handle these threats and protect themselves. Certain laws have been proposed for their protection but these are either ignored, cannot be enforced or they are not preventative. With this big data science research project we are proposing an identity deception indicator (IDI) for pointing out the uncertainty in trusting individuals on social media sites and protecting minors with this knowledge.

1 Project idea

The upsurge of internet use by minors have provided online predators, like pedophiles, with an unlimited source for soliciting new victims [1] [2]. This could be ascribed to the fact that, according to the authors, minors are likely to not be as protective about their personal information and tend to be more trusting of people they have just met [3] .

The idea of this research project is to use advances made in the fields of big data [4] and data science [5], cyber-security [6] and human factors [7] [8] to explore with an identity deception indicator towards protecting minors on social media sites like Twitter.
The project has been divided into various processes which consists of initially identifying, collecting and cleaning big data. Thereafter potential variables are identified and through inspection, machine learning and deep learning techniques enhanced to culminate in the creation of an identity deception indicator.

Figure 1 illustrates all processes in the research project and the current status of the research after the last semester.

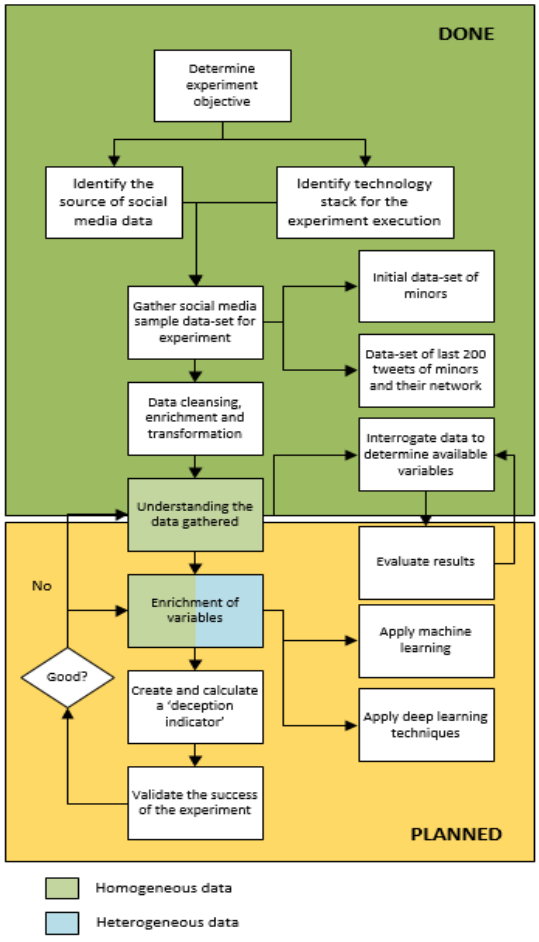


Figure 1: The project process diagram

1.1 Main deliverables

The main deliverables of 2015 were:

- To gather a big dataset for the experiment consisting of minors.
- To collect the last 200 tweets of all identified accounts as well as their friends and followers.
- To clean, enrich and transform the data.
- To understand the data through variable inspection.
- To explore in machine learning for enrichment and addition of more variables to the research at hand.

2 Use of HPI Future SOC Lab resources

To summarize the following resources were used for the research at the HPI Future SOC lab:

- Twitter: The Twitter4j Java API was used to dump the data needed for the experiment in a big data repository.
- Hortonworks Hadoop 2.2: For the purposes of this experiment HDP Hadoop runs on an Ubuntu Linux virtual machine hosted in "The HPI Future SOC"- research lab in Potsdam, Germany. This machine contains 4TBs of storage, 8GB RAM, 4 x Intel Xeon CPU E5-2620 @2GHz and 2 cores per CPU. Hadoop is well known for handling heterogeneous data in a low-cost distributed environment, which is a requirement for the experiment at hand.

  Flume: Flume is used as one of the services offered in Hadoop to stream initial Twitter data into Hadoop and also into SAP HANA.

  Sqoop: This service in Hadoop is used to pull data from SAP HANA back to the Hadoop HDFS. Analytical results e.g. machine learning and predictive modeling for the experiment will be generated on both Hadoop and SAP HANA. These results will be stored on both platforms and Sqoop facilitates this requirement.

  Ambari: For administration of the Hadoop instance and starting/stopping the services like Flume and Sqoop.

  Hue 3.7: The Hadoop UI for executing queries and inspective HDFS data.

- Java: Java is used to enrich the Twitter stream with additional information required for the experiment at hand.
- SAP HANA: A SAP HANA instance is used which is hosted in "The HPI Future SOC"- research lab in Potsdam, Germany on a SUSE Linux operating system. The machine contains 4TBs of storage, 1TB of RAM and 32CPUs / 100 cores. The in-memory high-performance processing capabilities of SAP HANA enables almost instantaneous results for analytics.

  The XS Engine from SAP HANA is used to accept streamed Tweets and populate the appropriate database tables.

- Machine learning APIs: Various tools are considered to perform classification, analysis and apply deep learning techniques on the data. These include the PAL library from SAP HANA, libraries in Python, the Hadoop Mahout service and the Graphlab platform.
- Visualization of the results is done in HTML, Angular and CSS.

The following ancillary tools were used as part of the experiment:

- For connection to the FSOC lab we used the OpenVPN GUI as suggested by the lab.
- For connecting and configuration of the Linux VM instance we used Putty and WinSCP
- For connecting to the SAP HANA instance we used SAP HANA Studio (Eclipse) 1.80.3

3 Findings in 2015

The purpose of this phase of the research project was to gather enough data, cleanse the data and to understand the variables available to the research.

We found that building a big data set (2-4TB in volume) is challenging in an experimental or research environment. Future work for the research at hand will increase the size of the data set through various options such as:

- Simulate data
- Ask Twitter to lift the rate limit
- Use application as opposed to user authentication for better rate limits
- Investigate other possible API calls
- Pull data from multiple Twitter accounts

We also found that even when data is available a lot of effort is to be applied on cleansing the data. We found it for example helpful to

- Ignore retweets
- Only use data from twitter accounts with less than 1,000 followers to filter out tweets from news and advertising agencies.

The initial inspection of the variables were quite promising in that we found that even though most twitter accounts have their location disabled, the location could potentially be deduced from the time zone of the account.
We also found that a simple word cloud interrogation showed promise towards extracting age from tweets as it seems that many twitter users indicated their class (e.g. 5th grade). This is depicted in figure 2.

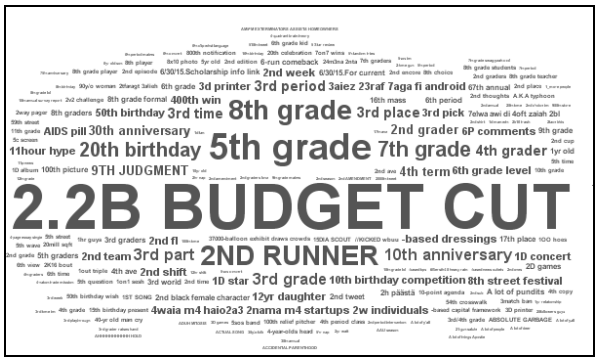


Figure 2: A word cloud of tweet content

The SAP HANA instance, virtual machine and storage was provided by the HPI FSOC research lab and the following is worth mentioning:

- There were no issues in connection.
- The lab was always responsive and helpful in handling any queries.
- The environment is very powerful and more than enough resources are available which makes the HPI FSOC research lab facilities ideal for the experiment at hand

Overall we found that the environment and its power enabled the collection of a big dataset without issue. The support of the HPI FSOC research lab is appreciated.

4 Next steps for 2016

The next steps in the project is to continue with the investigation and analysis of the variables available in the big dataset. Heterogeneous variables like the profile images contained in twitter accounts will also be considered. Machine learning and deep learning techniques will be applied to further enrich the variables for the experiment at hand.

Additional measures will then be applied, like Shannon entropy, to indicate whether the variable contributes in the value of the identity deception indicator.

The deliverables for this phase are:

- To clean to data based on initial findings from previous data interrogation
- To add more variables for experimentation
- To apply various different machines learning techniques in both SAP HANA and Hadoop
- To evaluate the results from these techniques and identify enriched variables
- To experiment with methods of identifying useful variables and weight their importance
- To produce an initial identity deception indicator per online persona

References

[1] A. Schulz, E. Bergen, P. Schuhmann, J. Hoyer, and P. Santtila, "Online Sexual Solicitation of Minors How Often and between Whom Does It Occur?," Journal of Research in Crime and Delinquency, p. 0022427815599426, 2015.

[2] R. Williams, "Children using social networks underage 'exposes them to danger," in The telegraph, ed, 2014.

[3] S. Livingstone, G. Mascheroni, K. Ólafsson, and L. Haddon, "Children's online risks and opportunities: comparative findings from EU Kids Online and Net Children Go Mobile," 2014.

[4] R. Kannadasan, R. Shaikh, and P. Parkhi, "Survey on big data technologies," International Journal of Advances in Engineering Research, vol. Vol. No. 3, Mar 2013 2013.

[5] F. Provost and T. Fawcett, Data Science for Business: What you need to know about data mining and data-analytic thinking: " O'Reilly Media, Inc.", 2013.

[6] R. Von Solms and J. Van Niekerk, "From information security to cyber security," Computers & Security, vol. 38, pp. 97-102, 2013.

[7] E. Hargittai, J. Schultz, and J. Palfrey, "Why parents help their children lie to Facebook about age: Unintended consequences of the 'Children's Online Privacy Protection Act'," First Monday, vol. 16, 2011.

[8] I. Liccardi, M. Bulger, H. Abelson, D. J. Weitzner, and W. Mackay, "Can apps play by the COPPA Rules?," in Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on, 2014, pp. 1-9.

Using Process Mining to Identify Fraud in the Purchase-to-Pay Process

- Final Report -

Galina Baader
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
galina.baader@in.tum.de

Veronika Besner
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
veronika.besner@in.tum.de

Michael Schermann
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3 85748 Garching, Germany
Michael.schermann@in.tum.de

Sonja Hecht
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
sonja.hecht@in.tum.de

Helmut Krcmar
Technische Universität München
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
krcmar@in.tum.de

Abstract

The aim of our project is to investigate the use of process mining to identify fraud in the purchase-to-pay business process in real-time. Therefore, we used the process mining tool Celonis[1] to reconstruct the as-is process from the underlying log data of our ERP system. Displaying the as-is process reveals process deviations which can further be analyzed regarding fraud. We applied our fraud detection strategy within an experiment, where the participants simultaneously hide fraud in the ERP system and analyze the fraud of the other teams. In total 9 out of 14 fraud cases were identified in real-time.

1 Introduction

Corporate fraud is a massive problem in most companies, consuming an estimated 5% of the annual revenues of a typical organization [1]. Computer assisted audit tools and techniques enabled to retrieve and analyze huge data volumes [2 ; 3] regarding fraud.

Most auditing tools use data mining based fraud detection. These methods do not take sequential information into account [2], which significantly limits such approaches. As process mining builds directly upon sequential dependencies, we concentrate on process mining for fraud detection in our research.

In most companies, fraud detection is performed once a year on a database export from the productive systems. One reason is that running fraud detection algorithms on a productive system might have a huge impact on the performance [3]. However, SAP HANA promises huge performance improvements. In our research aim to identify fraud in real-time without affecting the productive system.

2 Project Goal

The goal of the project is to determine the suitability of process mining to detect fraudulent behavior in real-time in the purchase-to-pay business process. We therefore use Celonis process mining, as the tool is able to visualize business processes based on event

[1] http://www.celonis.de/

logs from the underlying system (here SAP ERP). Deviations from the standard processes are visualized and can be investigated regarding fraudulent behavior.

The project should result in a proof of concept to show that fraudulent behavior can be efficiently found when using process mining. Furthermore, it should be analyzed, if SAP HANA can overcome performance issues, which limits the use of process mining on a productive system.

3 Project Design

The basic design of the project and its underlying dataset are presented in the following section.

3.1 Experimental setting

Fraud should be detected in real-time. As we do not have access to a productive system of an enterprise to run the process-mining tool and search for fraud, we designed an experiment, which simulates real-world conditions.

In a competition, participants of the experiment have to simultaneously commit fraud in an ERP system and detect the fraud of the other teams. For each successfully committed or identified fraud the respective team virtually gets the money. The team with the highest amount of money wins the competition. To ensure the quality of the committed fraud cases each fraud case first had to be discussed with a jury of professional accountants.

3.2 Technical Architecture

Figure 1 shows, how the different tools we used in the project are interconnected. At the client side, the SAP HANA Studio is installed and allows interaction and manipulation of data stored in SAP HANA. We created the needed event log tables for Celonis directly in HANA and gave Celonis access to the corresponding database scheme. Celonis was installed on a web server provided from the HPI.

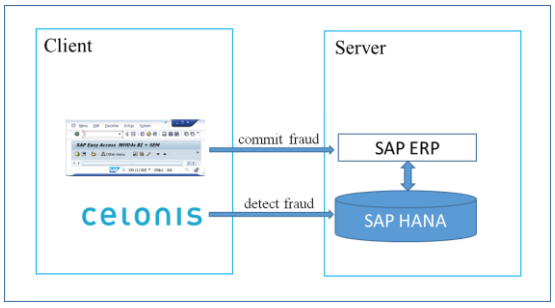


Figure 1: Project Architecture

4 Setting Up Celonis

In the following, the process of creating a detection strategy will be described.

The project consists of six main steps:

1. Load data into HANA
2. Create activity, case and process tables
3. Create data cubes and data models in Celonis
4. Create analysis
5. Detect process deviations, investigate analysis, filter and explore processes
6. Refine analysis to detect further fraud cases

In the following, these steps will be described in detail.

4.1 Relevant SAP Tables

We had access to an SAP ERP system running on SAP HANA. As fraud was hidden in the ERP system, the corresponding data were directly within the SAP HANA database. No data transfer was necessary.

Table 1 provides an overview of the most important tables that we used for the analysis.

Table Name	Table Description
EBAN	Purchase Requisition
EKKO	Purchasing Document Header
EKPO	Purchasing Document Item
EKBE	History per Purchasing Document
BSEG	Accounting Document Segment
BKPF	Accounting Document Header
LFA1	Vendor Master
CDHDR	Change Document Header
CDPOS	Change Document Items

Table 1: Important Purchase-to-Pay Tables [6]

4.2 Create Activity, Case and Process Tables

Once the data tables are imported into HANA, certain event log tables have to be created manually to display process models with Celonis. Therefore, we developed a script (using SQLScript as it is supported by HANA) that creates the respective activity, case and process table.

Activities represent every single step in every order (like purchase requisition, purchase order, invoice received etc.) and cases represent the complete process path of every order position. The third table, the process table, is a utility table and maps activities to integer activity IDs for Celonis' improvements. The ac-

tivity table is created based on the previously imported tables and afterwards the case and process tables are derived from this activity table.

In the following section, this table creation will be explained.

```
CREATE TABLE CELONIS_P2P_ACTIVITIES(
    ActivityCaseID VARCHAR(18)
    ,Activity VARCHAR(40)
    ,EventTime TIMESTAMP
    ,Sorting INTEGER
    ,EventUser VARCHAR(12));
```

Figure 2: Creation of the Activity Table

For the activity table, at first five different columns are needed. The creation of these columns can be seen in figure 2. The first one is the "ActivityCaseID" column, which is a unique number for each case. A case represents one single position of an order and all the steps belonging to this order. The "Activity" column is a textual representation of the current action, for example "Purchase Requisition", "Purchase Order", or "Invoice Receipt". The "EventTime" column provides a timestamp consisting of the date and the time when the action was performed. "Sorting" provides a classification of the normal order, in which the activities should be performed; "Purchase Requisition" should, for example, in a normal case happen before "Purchase Order", which again should happen before "Invoice Receipt". The "EventUser" is the SAP system user, who created the respective activity, for example, booked the purchase requisition or order in the system.

Three more columns are generated by a Celonis procedure. One is the "Lifecycle" column, which is derived from the "Sorting" and "Timestamp" columns and represents the position in a case's lifecycle. Another one is the "Case_Num_ID", which gives each case (defined by "ActivityCaseID") an integer number that is easier to process. Finally, a "PrimaryKey" is added for each position of the activity table.

To fill the activity table with values, for every activity different SAP tables have to be joined to meet certain criteria. To create the purchase requisition tables, for example, the tables EBAN and EKPO have to be joined, as it can be seen in figure 3.

```
INSERT    INTO    CELONIS_P2P_ACTIVITIES(
SELECT
    (EKPO.MANDT||EKPO.EBELN||EKPO.EBELP)
    AS ActivityCaseID,
    'Purchase Requisition' AS Activity,
    EBAN.BADAT AS EventTime,
    10 AS Sorting,
    EBAN.ERNAM AS EventUser
    FROM EBAN
    JOIN EKPO ON EBAN.MANDT = EKPO.MANDT
    AND EBAN.EBELN = EKPO.EBELN
    AND EBAN.EBELP = EKPO.EBELP);
```

Figure 3: Join of Tables EBAN and EKPO

Further, we join EKPO and EKKO to get the purchase orders, EKPO and EKBE to get the invoice receipts, or EKPO with CDPOS and CDHDR to get deleted positions or changed items. A deep understanding of the relevant tables is necessary. For example deletion of a position can be seen in table CDPOS (Change Document Items). The column "table name" should equal 'EKPO' to refer to the purchasing document and it must be deleted (so the fieldname equals 'LOEKZ' and the new value equals 'L').

Once we included all needed activities in the activity table, we created the case and the process tables. The creation of the process table is quite straightforward as it only maps an "Activity_ID" to each activity in the activity table.

Additionally to these four columns, one can choose to add further columns. It could, for example, be interesting to know the vendor for each case, as well as the ordered material and its price and amount.

4.3 Create Data Cubes and Data Models in Celonis

Once we have the activity, case and process tables we can create data cubes and models in Celonis. To do so we first give Celonis access to the respective database scheme in HANA, where the tables are located and choose this scheme as our data store. Then we define the tables Celonis should use to create our data model, in our case the activity, case and process tables created in step 2.

As a next step we can configure this data model and define, which tables and columns Celonis should use for each attribute. Finally we have to create at least one foreign key to connect the activity and case table.

5 Real-Time Fraud Detection

As soon as the data cube and data model is set up and configured properly, we can create our analysis. To do so we have to create a new document based on our data model, choose a suitable format and Celonis automatically creates a process model.

In figure 4 an exemplary process model can be seen. It represents the first part of the standard purchase-to-pay process going from purchase requisition to purchase order, to goods receipt and to invoice receipt without consideration of specific deviations like deleted positions or changed order of activities. As one can see most cases in the example do not have a purchase requisition, but start directly with the purchase order and not all cases go through all of the activities.

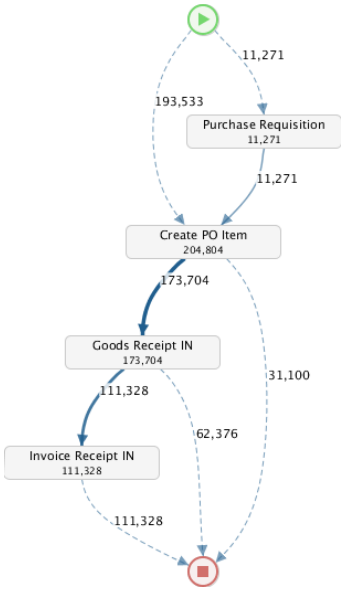


Figure 4: Exemplary Process Model

5.1 Investigate Analysis, Filter and Explore Processes

The model depicted in figure 4, e.g., is a strong simplification of all the processes that are derived from the used event log. Depending on the filter one choses, the tool displays a specific number of variants – which equal our different process paths. Within the fraud detection experiment, 16 variants of the process were identified. Many processes, for example, use a path that is similar to the one shown in figure 4, but where the invoice receipt was created before the good receipt creation. Since the order of these two activities can be swapped without consequences, this is therefore no indicator for fraudulent behaviour. A different path shows, e.g., all the deleted orders and another one all the orders where the price of the good was changed after the order was created. These deviations may, for example, be more interesting to check for occurring fraud cases. If needed, different filters can also be used, e.g., hide a certain activity or only show processes that start or end at a certain activity.

5.2 Refine Analysis to Detect Further Fraud Cases

The experiment participants analysed all process deviations in our dataset and found indicators for fraud. For example, we have identified hints that show a double issue of an invoice or the non-purchase fraud.

In double issue of an invoice the fraudster receives two or more invoices for one product to get a double payment from the vendor. We have seen 724 fraud cases where two invoices have been received for one product as it can be seen in figure 5. Analysing this data in detail reveals that in some cases the sum of the multiple invoices exceeds the amount of the prize for the ordered product, which is an indicator for fraud.

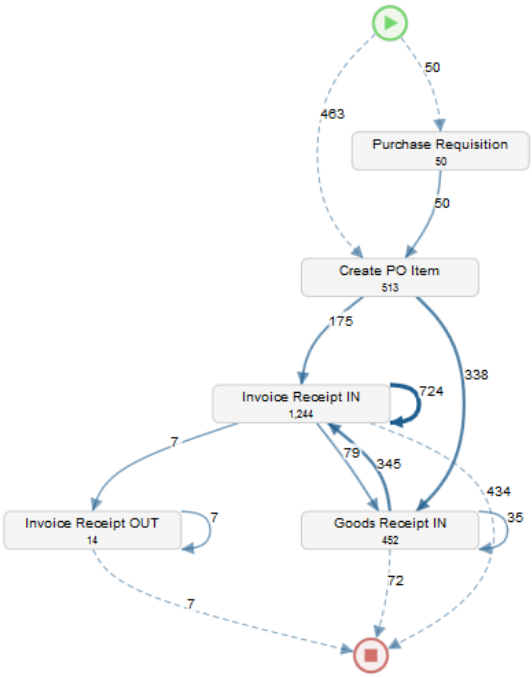


Figure 5: Double Received Invoices

As a further example we were able to identify 41 cases, where we have not received a good, but the invoice was issued as displayed in figure 6. In the so-called non-purchase fraud, a certain good is not delivered but the invoice paid. This is a further strong indicator for fraud.

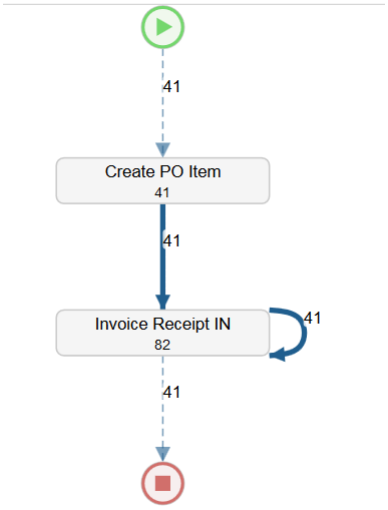


Figure 6: Invoice receipt without goods receipt

However, there are fraud cases included, that cannot be spotted through process deviations. One example is the kickback scheme. A fraudster and a complice vendor agree on an overpayment and share the overpaid

amount. This fraud can be analysed comparing different process instances and looking for deviations. Celonis further offers the possibility to create key figures using SQL. A key figure “over-average payment” can also be added to spot the respective fraud.

6 Conclusion and Outlook

We identified all necessary tables of the purchase-to-pay business process. We then created the necessary Celonis tables to be able to display the business process with all deviations. We conducted an experiment where the participants simultaneously hide and detect fraud. Out of 16 fraud cases 9 were identified in real-time. The performance of the ERP system was not throttled back. Our fraud analysis showed fraud cases like double issue of an invoice and the non-purchase fraud. We further analysed cases that cannot be identified through process deviations, but as data irregularities in certain columns or by comparing different process instances.

It should be subject to further research to identify process conformant fraud cases, by adding further hints for fraud.

Sources

[1] ACFE, „Report to the Nations on Occupational Fraud and Abuse,“ Association of Certified Fraud Examiners, Austin, Texas, USA, 2012.

[2] A. Bönner, M. Riedl und S. Wenig, Digitale SAP®-Massendatenanalyse: Risiken erkennen - Prozesse optimieren, Berlin (Germany): Erich Schmidt, 2011.

[3] D. Coderre, Computer Aided Fraud Prevention and Detection: A Step by Step Guide, John Wiley & Sons, 2009.

[4] C. Phua, V. Lee, K. Smith und R. Gayler, A Comprehensive Survey of Data Mining-based Fraud Detection Research., 2010.

[5] Y. Yannikos, F. Franke, C. Winter und M. Schneider, „3LSPG: Forensic tool evaluation by three layer stochastic process-based generation of data,“ in *Computational Forensics Vol. 6540*, Berlin/Heidelberg, Springer Verlag, 2011, pp. 200-211.

[6] "SAP Datasheet," [Online]. Available: http://www.sapdatasheet.org. [Accessed 18 03 2015].

On the Potential of Big Data Boosting Bio-inspired Optimization A Study Using SAP HANA

Bernd Scheuermann, Elisa Weinknecht
Hochschule Karlsruhe
Technik und Wirtschaft
University of Applied Sciences
Karlsruhe, Germany
bernd.scheuermann@hs-karlsruhe.de

Abstract

Meta-heuristics inspired by the principles of biology are often successfully applied to a wide range of complex optimization problems with practical relevance in business and engineering. To further boost the performance of such algorithms, this paper proposes to enhance bio-inspired optimization by a framework that provides guidance obtained through the analysis of extensive amounts of historical optimization knowledge maintained by SAP HANA. The review of related work shows that the envisaged framework makes progress beyond the state of the art bringing together the concepts of bio-inspired optimization, big data and in-memory database technology. The architecture of the framework is provided along with a discussion of the prospected advantages and challenges.

1 Introduction

Many optimization problems in industry are known to be NP-hard problem (see, e.g., [5] for a concise introduction to the intractability of optimization problems). For such problems, it is yet unknown if there exists an algorithm that is capable of finding the optimum in polynomial time. Such problems include, e.g., the Vehicle Routing Problem (VRP) [13], the Traveling Salesperson Problem (TSP) [8], or the Quadratic Assignment Problem (QAP) [2] or many other problems in production, warehouse and transportation logistics. Decades of research in the field of complexity theory suggest that solving NP-hard problems to optimality always requires exponential runtime, although it has never been proven.

Considering NP-hard combinatorial optimization problems, exact algorithms guarantee to find the optimum, but in the worst case, the search requires exponential time. On the other hand, many algorithms have been developed which afford only polynomial time. These approximate algorithms, also called heuristics, search for good solutions to the optimization problem, however, they cannot guarantee to find the optimum.

Many heuristics are problem-dependent, i.e. they exploit problem-specific knowledge and can therefore often provide good solutions in reasonably short time, although these heuristics are often very specialized for one sort of problem and can only hardly if ever be applied to others. In this context, so-called meta-heuristics play an important role, since they provide a generic approach to the creation of problem-specific heuristics. For some applications, the techniques of meta-heuristics offer the only way for an efficient optimization, when other heuristics cannot be properly adapted. A range of meta-heuristics has been inspired by principles observed in biology. Some popular examples of such bio-inspired metaheuristics include: Evolutionary Algorithms [6] (inspired by evolution and natural selection), Ant Colony Optimization [3] (inspired by the foraging behavior of ants), and Particle Swarm Optimization [7] (inspired through the dynamics of bird flocks or fish schools).

Although being very successful in a wide range of industrial-strength optimization scenarios, typically bio-inspired meta-heuristics suffer a major drawback: they ”forget” their search history. Therefore, this paper proposes an approach to resolve this weakness by designing and developing novel techniques for bio-inspired metaheuristics guided through extensive amounts of optimization knowledge managed by SAP HANA.

2 Related Work

In literature, several attempts have been made to enable meta-heuristics to memorize previously created solutions and to tackle the above-mentioned forgetfulness. Implicit memory, e.g. diploidy, is implemented by using redundant representations of solutions [9, 4]. Alternatively explicit memory stores specific information about useful solutions, which can be

re-inserted later again [14, 15]. An overview of implicit and explicit memory can be found in [1]. Abstract memory schemes [10, 11] store the abstraction of good solutions in the memory instead of good solutions themselves. This yields learning processes, which are functionally similar to machine learning and are claimed to improve the optimization performance of Evolutionary Algorithms. Simões and Costa [12] introduce variable-size memory for Evolutionary Algorithms with two populations: the main population searches the best solution, the other variable-size population serves as memory. In cyclic environments, the performance of the algorithm is increased by applying a genetic operator inspired by the biological conjugation process.
The review of the state-of-the-art suggests that all previous approaches to memory-based meta-heuristics restricted themselves to using as little main memory as possible and refrained from using database technology for storage and data persistence. This can be attributed to several reasons:

1. the high cost and limited capacity of main memory in the past,
2. the intractability of parallel programming needed to efficiently explore vast amounts of optimization knowledge,
3. the bottleneck experienced when accessing traditional disc-resident databases.

3 Using SAP HANA as Knowledge Store in Bio-inspired Optimization

On the one hand, the forgetfulness of traditional search heuristics is purpose and belongs to the strategy of the optimization approach, i.e. the impact of unfavorable decisions in the past shall fade away, whereas more recent favorable decisions shall be rewarded so as to steer the algorithm towards promising areas within the search space. On the other hand, the deletion of the search history induces the following disadvantages:

1. the algorithm loses track of the previously traversed paths in the decision graph,
2. earlier created solutions may be unknowingly visited again and need to be re-evaluated,
3. the algorithm is unaware of the extend of search space exploration and runs at risk of getting trapped in a local optimum due to premature convergence.

To overcome these major drawbacks, it is motivated to develop and to evaluate an approach that provides the optimizer with guidance obtained through the analysis of extensive historical optimization knowledge maintained by SAP HANA. This approach involves the following steps:

- To implement a selection of bio-inspired meta-heuristics that shall be directly executed on the SAP HANA platform.
- To use the SAP HANA database and its columnar storage to log the optimization knowledge and the solutions created throughout the optimization process.
- To implement big data techniques to analyze and to explore the optimization knowledge re-siding on HANA and to use the results to further guide the optimization.
- To evaluate the algorithms developed with the aim of comparing distinct strategies of logging data to SAP HANA and analyzing the benefit of big data in bio-inspired optimization.

It is anticipated that in a representative industry-strength optimization run the optimization data being logged and explored on HANA requires a storage volume between 1 GB and 1 TB.

3.1 Architecture

It is envisaged to work towards a programming and execution framework for bio-inspired optimization exploiting the in-memory database of SAP HANA as knowledge store and its analytical engines to explore

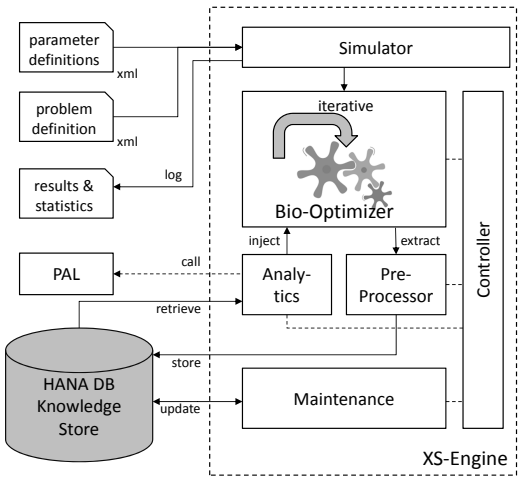


Figure 1: Architectural Overview

The framework is being implemented as native application on the SAP HANA XS Engine and consists of the following modules: A *simulator* reads the definition of the problem instance and parameter definitions of the optimization algorithm from files. These definitions are fed to the *bio-optimizer* which embodies the implementation of an iterative meta-heuristics as

introduced in the previous sections. In static environments, these definitions are read once; whereas in dynamically changing environments, the simulator produces a configurable online stream of change events that require the optimizer to adapt appropriately and to respond quickly. Typically, the algorithm continuously evolves on a set of optimization knowledge. Such knowledge is represented by raw data which depends on the algorithm and the problem considered. In the case of Evolutionary Algorithms, this raw data may consist of the genotypical representations of individuals created, along with associations describing the parent-offspring dependencies as well as fitness values and elitist solutions. With regards to Ant Colony Optimization, which is a constructive approach, solutions and their objective functions values are relevant as well. Furthermore, the distributions of pheromone trails could be memorized. In Particle Swarm Optimization, the vector representations of moving particles are memorized supplemented by coordinates describing the locations of individually and globally best solutions visited, or also including data describing the swarm topology and motion characteristics.

The *pre-processor* is responsible for extracting this raw data from the optimizer, pre-computing it and routing the results to the knowledge store residing in SAP HANA. Such pre-processing may include e.g. the filtering of memorized solutions, datatype transformations, the calculation of similarity, diversity, convergence figures, or computing the entropy of pheromone matrices. The *analytics* module evaluates the knowledge store and aims at maintaining solution diversity and identifying promising albeit unexplored areas in the search space. Here the clustering algorithms embodied in PAL (Predictive Analysis Library) may help to identify such areas. In dynamic environments, the analytics module may analyze the nature of the last dynamic change and explore historical data in HANA to find similar environments that could be re-instantiated and injected into the current optimization run. Furthermore, the capabilities of PAL may be exploited to analyze historical entries in the knowledge store thereby predicting future movements of optima. Injecting suitable solutions (or sub-populations) to predicted areas may make the algorithm better prepared and improve its responsiveness at the next change. Housekeeping tasks are performed by the *maintenance* module which could execute age-dependent or quality-dependent deletions of solutions and to automatically re-evaluate solutions at dynamic changes. Finally, the *controller* instance monitors and coordinates the interplay of the modules described above.

3.2 Challenges

Many open research questions and challenges arise. Some of them are related to the tasks of the maintenance and the controller module:

- To decide which knowledge (here solutions) should be stored in HANA and at which frequency. One may be in favor of storing the best, the above average or the latest solutions. This may reduce the amount of data stored but could prevent the algorithm from sustaining diversity and from identifying new unvisited areas.
- To determine the number and the set of solutions to be replaced when inserting new ones to the knowledge store. As opposed to earlier replacement strategies, the use of data analytics in HANA may suggest to create and maintain a larger knowledge base than before and to apply a less restrictive replacement strategy. Moreover, with HANA the calculation of the replacement candidate should also be expedited. Such calculations include e.g. a) deleting the solution which keeps the *highest variance* among the solutions in the knowledge store after deletion, b) replacing the *most similar* solution in the knowledge base, or c) determining the two solutions with pairwise *minimum distance* and deleting the worst of them from the knowledge store.
- To determine the proper set of solutions chosen from the knowledge store and when to inject them to the optimizer. In static environments, an injection may be advisable, when solution diversity drops below a threshold or when the best solution has not improved considerably for a number of iterations. In dynamic environments, also the occurrence of a change may trigger an injection of solutions.
- To determine a suitable data format for the stored knowledge in order to make best use of the row-store or column-store techniques in HANA, respectively. Here columnar tables prefer low-frequent insertions or columns with low cardinality.

4 Conclusion and Status of Work

This paper is concluded by summarizing the prospected advantages of the proposed approach:

- Logging and online-analysis of optimization knowledge shall enable the optimizer to learn from the decisions of the past and to make better-informed decisions in the forthcoming iterations of the optimization algorithm.
- Analyzing the optimization history shall enable the algorithms to discover as yet unexplored but promising regions in the search space. This shall exploit the expedited data exploration capabilities of SAP HANA.
- The optimizer may be enabled to detect and avoid previously visited solutions, which shall prevent

the algorithm from re-visiting them again and wasting computation time.

- Implementing the optimizer directly on the SAP HANA platform shall bring the algorithm to the proximity of the data thereby reducing latency and improving data throughput.

- In online optimization scenarios with dynamically changing constraints and objective functions, the optimization knowledge kept in HANA may be used to control and to maintain solution diversity, which may be readily exploited to guide the optimization process to promising areas after a change.

The development of the introduced framework is work in progress. Two biologically inspired optimizers (EA and ACO) for static environments have been implemented on SAP HANA. In initial experiments, extensive amounts of optimization knowledge have been logged to the SAP HANA database during optimization runs. A first procedure has been implemented to explore this optimization knowledge and to inject solutions into the EA optimizer following the random immigrants approach. Future research continues the development and empirical performance benchmarking to work towards the architecture and methods presented Section 3.

References

[1] J. Branke. Memory enhanced evolutionary algorithms for changing optimization problems. In *Proc. of CEC 99*, 1999.

[2] E. Cela. *The Quadratic Assignment Problem: Theory and Algorithms*. Kluwer Academic Publishers, 1998.

[3] M. Dorigo. *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, 1992.

[4] A. S. Etaner-Uyar and A. E. Harmanci. A new population based adaptive domination change mechanism for diploid genetic algorithms in dynamic environments. *Soft Computing*, page 803814, 2005.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1983.

[6] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[7] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, page 19421948. IEEE, 1995.

[8] E. L. Lawler, J. K. Lenstra, A. H. Kan, and D. B. Shmoys. *The Traveling Salesman Problem*. John Wiley & Sons, 1985.

[9] J. Lewis, E. Hart, and G. Ritchie. A comparison of dominance mechanisms and simple mutation on non-stationary problems. In *PPSN98*, page 139148, 1998.

[10] H. Richter and S. Yang. Memory based on abstraction for dynamic fitness functions. In *Proceedings of the 2008 conference on Applications of evolutionary computing, Evo08*, page 596605. Springer.

[11] H. Richter and S. Yang. Learning behavior in abstract memory schemes for dynamic optimization problems. *Soft Comuting. A Fusion of Foundations, Methodologies and Applications*, page 11631173, 2009.

[12] A. Simões and E. Costa. Variable-size memory evolutionary algorithm to deal with dynamic environments. In *Proceedings of the 2007 EvoWorkshops: Applications of Evolutionary Computing*, page 617626. Springer, 2007.

[13] P. Toth and D. Vigo, editors. *Vehicle Routing: Problems, Methods, and Applications. 2nd Edition*. SIAM - Society for Industrial and Applied Mathematics, 2014.

[14] S. Yang. Genetic algorithms with memory-and elitism-based immigrants in dynamic environments. *Evolutionary Computation*, 16:385–416, September 2008.

[15] S. Yang and X. Yao. Population-based incremental learning with associative memory for dynamic environments. *Evolutionary Computation*, 12(5):542 561, October 2008.