M. Heistermann | S. Collis | M. J. Dixon | J. J. Helmus | A. Henja
D. B. Michelson | Thomas Pfaff

An Open Virtual Machine for Cross-Platform Weather Radar Science

Suggested citation referring to the original publication:
Bull. Amer. Meteor. Soc. 96 (2015), pp. 1641–1645
DOI http://dx.doi.org/10.1175/BAMS-D-14-00220.1

An Open Virtual Machine for Cross-Platform Weather Radar Science

BY M. HEISTERMANN, S. COLLIS, M. J. DIXON, J. J. HELMUS, A. HENJA, D. B. MICHELSON, AND THOMAS PFAFF

TOWARD OPEN AND INTEROPERABLE SOFTWARE IN WEATHER RADAR SCIENCE. Among those who work in the field of the atmospheric and related oceanic and hydrologic sciences, only a minority will consider themselves as professional software developers. Still, many will find themselves writing software in their daily work. The nature of that software might vary, from small scripts to large applications, from simple to complex, and from good to perhaps not so good. Commonly, software development is repeated over and over by different organizations, and these systems rarely interact well with each other.

These aspects certainly hold for the subdiscipline of radar meteorology, in which scientific software is developed for various purposes, such as receiver-level processing, the management of large amounts of data (often in a network of radars), analysis and quality control, visualization and display, or for application in related fields such as hydrology or atmospheric modeling.

In a recent *BAMS* article, Heistermann et al. (2014) argued that community-based Open Source Software (OSS) could foster scientific progress in weather radar research. More specifically, the authors claimed that community-based OSS could make weather radar software more affordable, more flexible, more transparent, more sustainable, and more interoperable. In particular, the vision of interoperability between different software tools was emphasized in that article:

> [. . .] we expect different [OSS] projects to coexist, and to interact in a dynamic collaboration. The guiding principle of such a cross-project collaboration will be interoperability, allowing not only for the platforms to exchange data, but also to exchange code (e.g., as shared libraries, code fragments, and Open Algorithms), and thus speed up technological and scientific progress through cross-fertilization [. . .]

A typical scenario illustrating the importance of interoperability would be the following: an agency uses software **A** for radar-based precipitation retrieval in real-time. However, the agency is not happy with the performance of the clutter detection. Some university has developed software **B** and claims it can do better in terms of clutter detection. Before expensively porting that algorithm to software **A**, though, the agency wants to verify that it actually adds value to their product. Interoperability of both **A** and **B** would allow the agency to run their particular radar processing chain **A**, but to insert the clutter detection algorithm from package **B** in order to evaluate its specific effect on product quality.

Good progress toward interoperability has already been achieved in radar-related research and applications. Most notably, the open data models ODIM_H5 (based on the HDF5 format) and CfRadial (based on the NetCDF format) facilitate data exchange via clearly defined interfaces and by using open libraries available for many programming languages (see "For Further Reading" section). And with the emergence of various OSS tools, developers have more flexibility in tailoring interfaces or chaining different tools together. However, both software developers and users may still find it difficult to get started: often, the available applications and libraries are cumbersome to install,

AFFILIATIONS: HEISTERMANN—University of Potsdam, Institute of Earth and Environmental Sciences, Potsdam, Germany; COLLIS AND HELMUS—Argonne National Laboratory, Argonne, Illinois; DIXON—National Center for Atmospheric Research (NCAR), Boulder, Colorado; HENJA—HENJAB, Växjö, Sweden; MICHELSON—Swedish Meteorological and Hydrological Institute, Norrköping, Sweden; PFAFF—Blue Yonder, Karlsruhe, Germany

CORRESPONDING AUTHOR: Maik Heistermann, Karl-Liebknecht-Str. 24-25, 14476 Potsdam, Germany
E-mail: maik.heistermann@uni-potsdam.de

DOI:10.1175/BAMS-D-14-00220.1

and for inexperienced users, it may be challenging to satisfy the countless dependencies; different operating systems might have particular issues, or not be supported at all; and most tools will have a steep learning curve, and entry points for a quick "look and feel" might be hard to find.

In this article, we present an open, community-based virtual machine designed to overcome some of these barriers.

AN OPEN VIRTUAL MACHINE AS AN INTEROPERABILITY SANDBOX. According to Wikipedia, a *virtual machine* (VM) is a software implementation of a machine (i.e., a computer) that executes programs just as a physical machine would. You can think of it as a fully configured computer system that you can run and access within your own physical computer (desktop, server, or in the cloud). A *virtual appliance* is a preconfigured virtual machine *image*, ready to run on any computer and any operating system that provides the required software (a so-called *hypervisor*). Virtual appliances are intended to eliminate the installation, configuration, and maintenance costs associated with running complex stacks of software, such as the OSS weather radar software we have been referring to above. A virtual appliance is portable by definition, and always behaves the same, irrespective of the physical environment in which it is run. This way, reproducibility and comparability are guaranteed for any results produced. Interestingly, the potential of virtual machines has already been extensively exploited in other research disciplines such as bioinformatics (see e.g., Angiuoli et al. 2011).

What is on our VM? Our VM is based on an Ubuntu Linux distribution. As stated before, though, it can be run on any common operating system (Linux, Windows, Mac OS) by using VirtualBox as a hypervisor (free installers can be downloaded from www.virtualbox.org). For example, this enables Windows users to enjoy software that is often supported for Linux only, without having to worry about compilation issues or cygwin (which is a UNIX-like environment and command-line interface for Microsoft Windows).

It is beyond the scope of this article to list all the features installed on the VM. Most importantly, it contains a suite of OSS tools that are related to weather radar processing:

- BALTRAD (http://git.baltrad.eu)
- Py-ART (http://arm-doe.github.io/pyart)
- wradlib (http://wradlib.bitbucket.org)
- RSL (http://trmm-fc.gsfc.nasa.gov/trmm_gv/software/rsl)
- RADX (www.eol.ucar.edu/software/radx)

For further reading, the article by Heistermann et al. (2014) provides brief profiles for these tools. Since BALTRAD, Py-ART, and wradlib have Python integration, the VM also comes with a scientific Python stack. In addition to the software, the machine contains a set of tutorials and sample data that serve as entry points for new users. The main platform for these tutorials is the IPython notebook (http://ipython.org/notebook.html)—a web-based interactive computational environment in which code execution, text, mathematics, plots, and rich media can be combined in a single document. Since our prime motivation was to foster the idea of interoperability, the machine contains specific recipes in which we demonstrate the combination of different radar software tools in one workflow.

What can you use the VM for? First of all, consider the VM as a virtual playground. You have instant access to all of the above software products without any nasty installation and configuration procedures. All the installed software products provide ample documentation on the web, but you can also get started using the tutorials and recipes included in the VM via the IPython notebook.

Not only can you use the VM to teach yourself, you can also use it to teach others. The IPython notebook is the perfect tool for interactive coding exercises, and for integrating other media and course materials. By building your courses on top of the VM, you will not have to worry that exercises will fail on some of the computers, or that your students will produce different results on different computers. And your students (or the administrator of your computer pool) will not have to worry about preparing the machines with the required course software.

You can go further by connecting any directory (or folder) on your host machine or in your network to the VM. In this way, you can use the VM in the same way you would use any other program for data processing. Should you need more performance, the VM is easily scalable: you decide yourself how much RAM and CPU you want to assign to it.

From a research perspective, the VM is an ideal environment to compare the performance of different algorithms from different software packages in benchmarking experiments. This is facilitated already by the existing level of interoperability between the included software packages, allowing you to chain algorithms from different tools into custom-tailored workflows.

The VM could contribute to the idea of reproducibility in scientific publishing, and to allow for a more rigorous peer review. According to the American Physical Society, "[. . .] the success and credibility of science are anchored in the willingness of scientists to expose their ideas and results to independent testing and replication by other scientists." But as the "Yale Law School Roundtable on Data and Code Sharing" put it in 2010: "Generating verifiable knowledge has long been scientific discovery's central goal, yet today it's impossible to verify most of the computational results that scientists present at conferences and in papers." The VM could be a useful component in achieving reproducible weather radar research, as it provides an open and fully reproducible environment across platforms. It should be noted, though, that achieving reproducible research not only requires open and reproducible software platforms, but also open access to raw data that were used in corresponding studies along with openly and completely documented algorithms. Using the VM, *Open Source, Open Data, and Open Algorithms* can actually become *Open Science*. While in the United States, archives of, for example, NEXRAD data are unconditionally open to the public, radar data (particularly raw data) are difficult to obtain in many other countries, including most EU member states.

Surely, there is a range of use cases we have missed here. You might find, at a certain point, that you need closer integration with a specific software tool within your system. In that case you can go ahead and install it on your own.

What makes the VM "open"? The proposed VM is not intended to be a static product, but an open and dynamic environment that is extendable, scalable, and adjustable. What's the key in achieving this? The VM is built by a collection of scripts using a tool called Vagrant (www.vagrantup.com). These "build scripts" are hosted on a public repository (https://github.com/openradar/oss_weather_radar_vm). Users are invited to fork from that repository (i.e., to create their own working copy); to modify the scripts in order to equip the machine with additional software, data, or

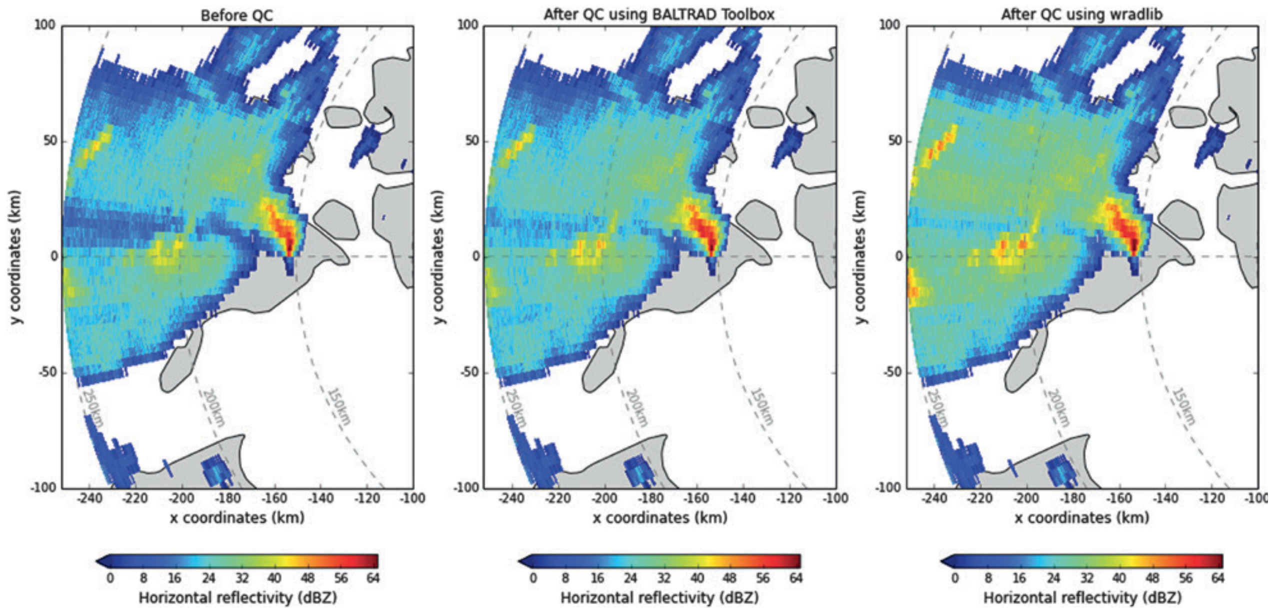
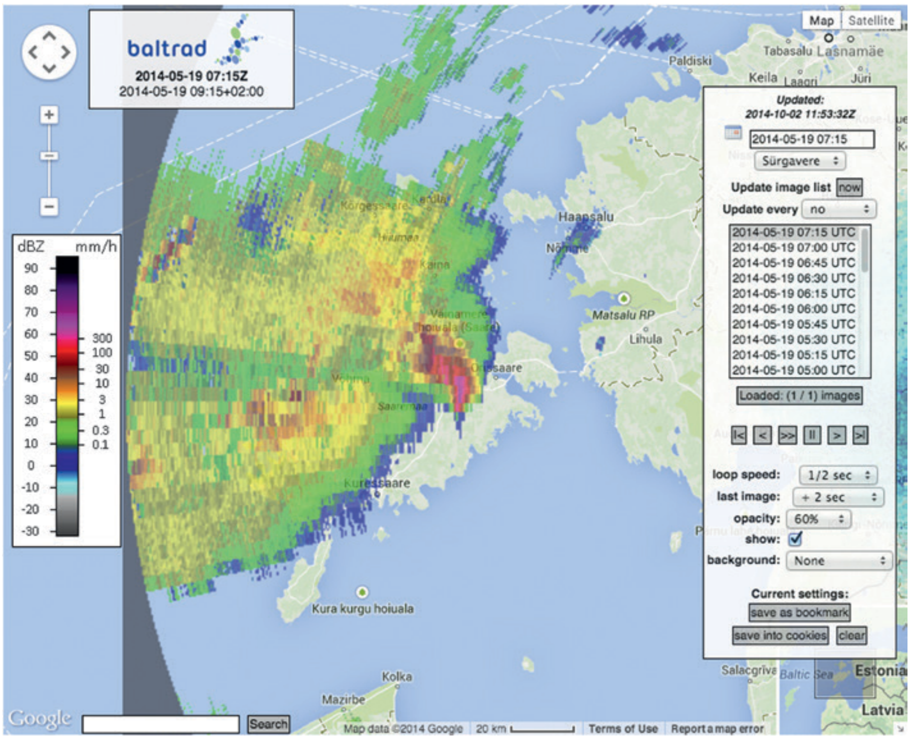


FIG. 1. Figure from the BALTRAD/wradlib interoperability demonstration at the Open Source Radar Short Course in the context of ERAD 2014, based on data from lowest sweep angle of the Estonian C-band radar near Sürgavere on 19 May 2014 at 0715 UTC. This case presents a zoom into a convective event with strong attenuation; (left) horizontal reflectivity before quality control (QC); (center) after QC using the BALTRAD Toolbox, including attenuation correction via RADVOL-QC with default settings; (right) after attenuation correction using the wradlib.atten module. BALTRAD output (ODIM_H5 compliant) was georeferenced and visualized using wradlib.



Updated: 2014-10-02 11:53:32Z

2014-05-19 07:15

Sürgavere

Update image list: now

Update every: no

2014-05-19 07:15 UTC
2014-05-19 07:00 UTC
2014-05-19 06:45 UTC
2014-05-19 06:30 UTC
2014-05-19 06:15 UTC
2014-05-19 06:00 UTC
2014-05-19 05:45 UTC
2014-05-19 05:30 UTC
2014-05-19 05:15 UTC
2014-05-19 05:00 UTC

Loaded: (1 / 1) images

loop speed: 1/2 sec
last image: + 2 sec
opacity: 60%
show: ✓
background: None

Current settings:
save as bookmark
save into cookies
clear

Fig. 2. Output of the BALTRAD/Py-ART interoperability exercise, showing data read by the BALTRAD Toolbox, interfaced in memory to Py-ART where dual-polarization moments were used to perform attenuation correction on horizontal radar reflectivity factor using default settings (see Giangrande et al. 2014 for details). The corrected reflectivity were then interfaced from Py-ART back to BALTRAD in memory and mapped to a regular grid, where they were then visualized using BALTRAD's GoogleMapsPlugin functionality.

examples; and also to request that those modifications be pulled back into the main repository. And yet, anyone who does not wish to build the VM using Vagrant can download and deploy the latest image of the VM.

What is the fastest way to get started? Find all required guidance on http://openradar.github.io: download the latest VM image (about 1 GB) and launch it using VirtualBox (which needs to be installed on your machine). Once your VM is deployed, you can get started with examples provided on the web page.

REALITY CHECK: THE OPEN SOURCE RADAR SHORT COURSE AT ERAD 2014. On 1–5 September 2014, the 8th European Conference on Radar in Meteorology and Hydrology (ERAD) took place in Garmisch-Partenkirchen, Germany. On the Sunday before the conference, the author team of this article organized a one-day short course in order to make users familiar with the concepts behind some Open Source weather radar software tools and to equip them with hands-on examples and tutorials. The tools presented in that course were BALTRAD, Py-ART, and wradlib. Another aim of the course was to demonstrate interoperability between these three software tools (see Fig. 1 and Fig. 2 for interoperability examples from the course). The VM seemed to be the ideal vehicle to achieve these objectives.

To get an idea about the background and motivations of the course participants, we carried out a brief survey. Twenty-five participants from different institutional backgrounds attended the course; 60% were from academia, 28% from government agencies, and 12% from private companies. As professional priorities, most of them picked "operational radar data processing," "interactive radar data analysis," and "development of algorithms." Somewhat less emphasis was put on the option "system development." Concerning their motivation to attend the course, 84% of the attendees endorsed the statement "I want to use Open Source Radar Software for my work," 71% agreed that "I want to use Python for radar data processing," and 67% chose "I want to collaborate with other developers."

The only requirement for the participants was to install VirtualBox before the course, while prebuilt VM images for 32-bit and 64-bit machines were handed to the participants on a USB stick. Of the 25 participants, two failed to run the VM during the course due to yet-unresolved issues on old Windows XP machines. The rest of the participants (Linux, WindowsXP/7, and Mac OS X) were able to run the VM and the IPython notebooks without problems.

THE FUTURE. We believe that the VM presents an important step toward Open (Weather Radar) Science. It may become a platform that allows combining various software products, but also compares different tools against each other. It could catalyze progress toward interoperability, and it will definitely lower the barrier for new users and developers, thus extending the weather radar community and user base.

ACKNOWLEDGMENTS. BALTRAD software has been developed as part of the BALTRAD and BALTRAD+ projects that have been partly financed by the European Union (European Regional Development Fund and European Neighbourhood and Partnership Instrument). Argonne National Laboratory's work was supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, under Contract DE-AC02-06CH11357. The development of wradlib was partly funded by the German Federal Ministry for Research and Education within the PROGRESS project. We thank Remko Uijlenhoet and an anonymous referee for helpful comments on the original manuscript.

FOR FURTHER READING

Angiuoli, S. V., and Coauthors, 2011: CloVR: A virtual machine for automated and portable sequence analysis from the desktop using cloud computing. *BMC Bioinformatics,* **12,** 356, doi:10.1186/1471-2105-12-356.

Dixon, M., W.-C. Lee, B. Rilling, and C. Burghart, 2013: CfRadial Data File Format: Proposed CF-compliant netCDF Format for Moments Data for RADAR and LIDAR in Radial Coordinates, 66 pp. [Available online at www.eol.ucar.edu/system/files/CfRadialDoc.v1.3.20130701.pdf.]

Giangrande, S. E., S. Collis, A. K. Theisen, and A. Tokay, 2014: Precipitation estimation from the ARM distributed radar network during the MC3E campaign. *J. Appl. Meteor. Climatol.,* **53,** 2130–2147, doi:10.1175/JAMC-D-13-0321.1.

Heistermann, M., S. Jacobi, and T. Pfaff, 2013: Technical note: An open source library for processing weather radar data (wradlib). *Hydrol. Earth Syst. Sci.,* **17,** 863–871, doi:10.5194/hess-17-863-2013.

——, and Coauthors, 2014: The emergence of open source software for the weather radar community. *Bull. Amer. Meteor. Soc.,* **96,** 117–128, doi:10.1175/BAMS-D-13-00240.1.

Lin, J. W.-B., 2012: Why Python is the next wave in Earth sciences computing. *Bull. Amer. Meteor. Soc.,* **93,** 1823–1824, doi:10.1175/BAMS-D-12-00148.1.

Michelson, D. B., J. Koistinen, T. Peltonen, J. Szturc, and M. R. Rasmussen, 2012: Advanced weather radar networking with BALTRAD+. ERAD 2012, *The Seventh European Conf. On Radar In Meteorology And Hydrology,* Toulouse, France, Meteo France. [Available online at www.meteo.fr/cic/meetings/2012/ERAD/extended_abs/NET_073_ext_abs.pdf.]

Michelson, D. B., R. Lewandowski, M. Szewczykowski, H. Beekhuis, and G. Haase, 2014: EUMETNET OPERA weather radar information model for implementation with the HDF5 file format. Version 2.2. EUMETNET OPERA Deliverable. [Available online at www.eumetnet.eu/sites/default/files/OPERA2014_O4_ODIM_H5-v2.2.pdf.]

The Yale Law School Roundtable on Data and Code Sharing, 2010: Reproducible research. *Comput. Sci. Eng.,* **12,** 8–12.