

A Lidar and Radar Meteorology Science Gateway for Education and Research on the NSF Jetstream2 Cloud

1st Jennifer DeHart

*Department of Atmospheric Science
Colorado State University
Fort Collins, CO USA
<https://orcid.org/0000-0001-8536-4927>*

2nd Brenda Javornik

*Earth Observing Laboratory
NSF NCAR, UCAR
Boulder, CO USA
brenda@ucar.edu*

3rd Julien Chastang

*NSF Unidata Program Center
UCP, UCAR
Boulder, CO USA
<https://orcid.org/0000-0003-2482-3565>*

4th Ana Espinoza

*NSF Unidata Program Center
UCP, UCAR
Boulder, CO USA
<https://orcid.org/0000-0002-6292-073X>*

5th Michael Dixon

*Earth Observing Laboratory
NSF NCAR, UCAR
Boulder, CO USA
<https://orcid.org/0000-0002-9597-9113>*

6th Michael Bell

*Department of Atmospheric Science
Colorado State University
Fort Collins, CO USA
<https://orcid.org/0000-0002-0496-331X>*

Abstract—This paper introduces a lidar and radar meteorology science gateway deployed on the NSF Jetstream2 cloud, designed to enhance educational and research activities in atmospheric science. Utilizing the "Zero to JupyterHub with Kubernetes" workflow, we have created a science gateway that integrates lidar and radar meteorology software packages, notably the Lidar Radar Open Software Environment (LROSE). This integration allows users to execute applications directly from the JupyterLab terminal, streamlining the creation of datasets for further analysis and visualization within Jupyter notebooks. By combining traditional command-line operations with modern Python-based tools for data analysis and visualization, this gateway provides a robust end-to-end solution that caters to both educational and research needs. The gateway has already been vital in facilitating LROSE instructional workshops and will see future enhancements such as GPU acceleration to boost performance. Our work demonstrates the significant potential of merging established scientific computing techniques with advanced Python environments, opening new avenues for computational science education and research.

Index Terms—LROSE, Cloud-based Science Gateway, Radar Meteorology, Lidar Meteorology, Data Visualization, Educational Technology in Atmospheric Science, JupyterHub, NSF Jetstream2, Atmospheric Data Analysis

I. INTRODUCTION

In the rapidly evolving field of computational science, integrating traditional scientific tools with modern data analysis platforms is essential for advancing research and educational outcomes. The Lidar Radar Open Software Environment (LROSE; [1] [2] [3]) is an open-source project collaboratively developed by Colorado State University's Department of Atmospheric Science and the Earth Observing Laboratory at the National Center for Atmospheric Research. It provides modular tools for processing lidar and radar observations, including data conversion, derived quantities, quality control, gridding,

and visualization. The integration of Jupyter notebooks and JupyterLab with sophisticated scientific computing tools like LROSE facilitates access to advanced analytical capabilities, allowing users to manipulate software packages directly from a web browser. By deploying these tools on the NSF Jetstream2 cloud [4] [5] through a JupyterHub science gateway, we significantly lower barriers to entry and democratize access to compute resources for educational institutions and researchers, facilitating seamless transitions between executing complex command-line tools and visualizing outputs in pre-configured Jupyter notebooks. Additionally, it facilitates the hosting of LROSE instructional workshops, enhancing learning through hands-on experience. This integration, which includes tools like the Python ARM Radar Toolkit (Py-ART) [6], enhances educational and research activities by making advanced lidar and radar data analysis more accessible.

A. LROSE Background

The goal of the LROSE suite is to provide the lidar and radar community with well-tested applications for scientific research. The primary programming language of most LROSE applications is C++, providing the computational speed to work with large datasets. The LROSE team contributes to the Open Source Radar community [7]. LROSE applications can be categorized into the following: data conversion, display, gridding, quality control, and scientific analysis of echo and Doppler wind observations. Examples of the scientific analysis supported by LROSE include particle identification (e.g., distinguishing between rain and snow), quantitative precipitation estimation analysis, and wind retrievals from either a single or multiple Doppler instruments. Most applications, particularly those that perform scientific analysis, are based on algorithms

that are described in peer-reviewed literature, of which Radx-Pid [8] and RadxEvad [9] are examples.

B. Why LROSE in a Science Gateway?

The LROSE team first explored the idea of an LROSE Science Gateway to improve accessibility. Educational exercises are often complicated by users having a variety of operating systems, potentially different versions of LROSE, and varying levels of experience with the applications. A cloud-based science gateway removes any installation barrier and ensures all participants are using the same version of the software. The use of JupyterHub as a science gateway was proposed by Zonca and Sinkovits [10]. Expanding upon the idea of using JupyterHub as a science gateway, since 2020 the NSF Unidata Program Center at the University Corporation for Atmospheric Research, has deployed numerous specialized PyAOS (Python for Atmospheric and Ocean Science) JupyterHubs tailored for semester-long academic courses and workshops [11] [12]. JupyterHub, and particularly Jupyter notebooks, is ideal as a Science Gateway for LROSE as notebooks mimic user workflows.

Previously, the LROSE team explored a specialized graphical user interface (GUI) as a gateway; however, any modifications to the workflow required development by software engineers. In general, the LROSE audience is familiar with programming in Python and wants to customize their interactions with LROSE. With Jupyter and Python, the interface is directly customized, without the need for additional software engineering. Specialized JupyterLabs can be deployed for specific workshops by the software engineers and radar scientists can focus their efforts on tutorial and Jupyter notebook materials. Finally, community members can develop their own workflows and submit them to the Gateway, thereby increasing engagement.

II. METHODOLOGY

In the spring of 2023, the LROSE team acquired an exploratory allocation on the NSF Jetstream2 cloud at Indiana University through ACCESS [5]. To develop the LROSE Science Gateway [13], we employed the "Zero to JupyterHub with Kubernetes" workflow ported to the NSF Jetstream2 cloud [10], enabling rapid and scalable deployment to accommodate a variable number of users. Authentication is managed via GitHub OAuth, reducing the maintenance burden. Since LROSE is a collection of C/C++ applications, we configured Docker containers based on the Jupyter Docker Stack to integrate the LROSE software, including command-line tools such as RadxConvert, FRACTL (Fast Reorder and CEDRIC Technique in LROSE), and SAMURAI (Spline Analysis at Mesoscale Utilizing Radar and Aircraft Instrumentation; [14]) available via the JupyterLab terminal. These containers also include Conda package manager environments equipped with Python packages like Py-ART [6] and Metpy [15] for further data analysis. A shared drive visible to all JupyterHub users contains instructional datasets for lidar and radar data analysis, which users can access and manipulate using LROSE

command-line tools. Additionally, we crafted custom Jupyter notebooks that can function as tutorials or operational tools, complete with pre-loaded examples to guide users from data analysis to visualization.

A significant amount of effort was made to optimize resource allocation and enable the execution of computationally intensive command-line applications like SAMURAI within a multi-user JupyterHub environment. SAMURAI's substantial resource demands often caused the single user server to stall, rendering it inaccessible. Additionally, heavy concurrent usage by multiple users risked overloading the entire Kubernetes cluster. To mitigate these issues, we increased the CPU and RAM allocation per user and adjusted Kubernetes pod configurations to ensure the JupyterHub "core" pods are isolated from the single user pods which utilize the majority of CPU cycles when running such intensive workflows [16]. These adjustments aimed to ensure cluster stability, despite occasional individual node disruptions. This remains an active area of research as we continue to refine our resource management strategies.

III. RESULTS AND DISCUSSION

A. Education Applications

These tutorials are designed to be used in workshop or classroom environments or by users who wish to learn more about specific LROSE tools. All the notebooks are also available on GitHub for offline usage on personal devices. The current tutorials offered on the LROSE Science Gateway highlight data quality control, particle identification (e.g., distinguishing between rain and snow, identifying ground clutter or birds), quantitative precipitation estimation, multi-Doppler analysis (e.g., SAMURAI), and the creation of mosaics from multiple radars. The user interface and the introductory portion of the quantitative precipitation estimation tutorial are shown in Fig. 1. Initial tutorials provided pre-made parameter files so a user could run all relevant applications without worrying about incorrect parameters that might produce errors. Based on user feedback, we have since created a "guided" multi-Doppler analysis tutorial that helps the user understand how to set each necessary parameter, inspired by instructional notebooks created for Metpy [15]. We plan to expand this type of tutorial to the existing notebooks.

Our first class-based tutorial took place in the spring of 2024, where students in Prof. Michael Bell's Radar Meteorology class generated dual-Doppler analyses from two mobile radars that sampled a tornado in Goshen County, Wyoming observed during the Verification of the Origins of Rotation in Tornadoes Experiment (VORTEX2) field campaign in 2009. LROSE applications RadxPrint, FRACTL, and SAMURAI were used to generate gridded analyses of the two-dimensional wind field. RadxPrint provides the students with the radar file metadata, including variable names needed to run FRACTL and SAMURAI. FRACTL and SAMURAI use two different approaches to retrieve two-dimensional winds from radar data; FRACTL's point error statistics help identify regions where the SAMURAI analysis is of higher quality. Once the gridded

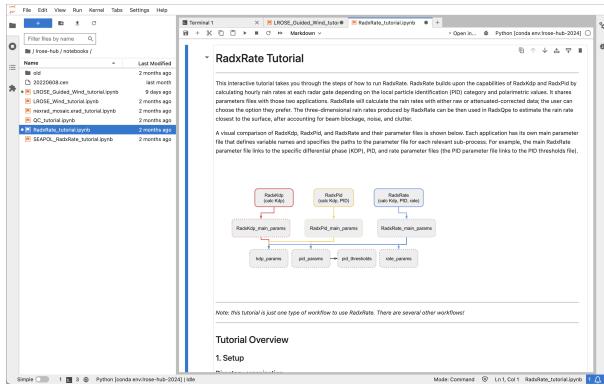


Fig. 1. A screenshot of the JupyterLab layout and the RadxRate tutorial that a user sees after logging into the LROSE Science Gateway.

analyses were produced, standard Python packages matplotlib [17], numpy [18], and xarray [19] were used in conjunction with cartopy [20] and Metpy [15] to visualize the wind field and storm structure; one of the images the students created in the tutorial is shown in Fig. 2, showing the horizontal tornado circulation in vectors and the radar reflectivity in color shading. Students provided positive feedback about the exercises in the "guided" tutorial and constructive feedback to improve tutorials for the future.

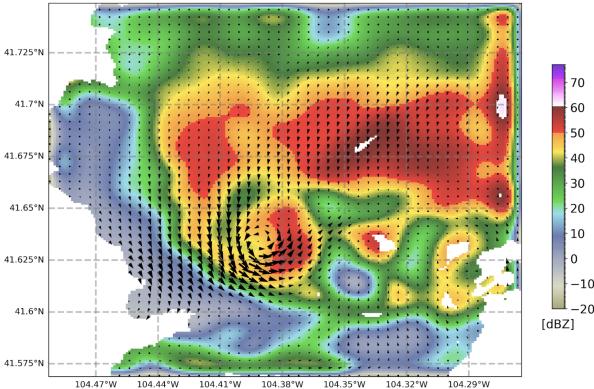


Fig. 2. Map of SAMURAI-analyzed 1.5-km radar reflectivity (colors) and wind vectors (arrows) of the VORTEX2 tornado case analyzed during the Spring 2024 classroom exercise.

B. Challenges

As previously discussed, computationally-intensive applications like SAMURAI are challenging to run, even when data sets are carefully chosen to minimize memory and computational usage for demonstrations and exercises. For example, SAMURAI will utilize as much CPU as is allocated to the environment—in this case, the Kubernetes pod—in which it is executed. When stress testing SAMURAI on the JupyterHub with two simultaneous JupyterLab instances on the same node, we noticed a slow down in the execution time as compared to the same workflow running on only one JupyterLab instance at a time. In addition, the step in the workflow where data is

read into the program also suffers a significant performance hit with two simultaneous workflow executions. While the latter is being investigated in the SAMURAI code, the former is still a subject of research and uncertainty. It is speculated that the multiple degrees of separation from the physical hardware (i.e., a hypervisor/VM, Kubernetes/containerization, JupyterHub/Lab) may affect how efficiently SAMURAI and other computationally intensive processes are executed. We find it necessary to close the knowledge gap using new or existing documentation and research in order to better determine the cause of the performance losses due to simultaneous execution. One way to mitigate them is to isolate single user pods to their own Jetstream2 virtual machine. This solution, however, is only realistically viable for certain scenarios, as described below.

With open-ended assignments, in contrast to time-limited workshops, preallocating virtual machines can unnecessarily drain ACCESS resources, particularly when usage is inconsistent (e.g., overnight when students are not working). In the cases where gateway usage is better known, we use the soft scale technique to quickly bring nodes up and down as needed [21]. This technique reduces the allocation usage rate by allowing for worker nodes to be pre-configured as part of the Kubernetes cluster and then "shelved," a VM state that does not exhaust Service Units. However, this technique requires developer intervention. Automated scaling of virtual machines on Jetstream2 would be a useful functionality in the future. The authors are open to feedback or suggestions on how the hurdles described in this section can be overcome.

IV. CONCLUSION AND FUTURE WORK

In conclusion, the LROSE Science Gateway enhances the accessibility and usability of advanced atmospheric analysis tools in education and research. By leveraging the NSF Jetstream2 cloud infrastructure and integrating it with JupyterHub, we have effectively reduced the entry barrier for students and researchers to utilize sophisticated lidar and radar data analysis tools. This gateway facilitates seamless transitions between command-line operations and interactive data visualization, making it easier to enter this specialized field.

Forthcoming development efforts will focus on improving the gateway's performance and exploring GPU acceleration. Future educational plans include partnering with more university classrooms, running an intermediate SAMURAI tutorial to help users understand how the advanced parameters affect scientific results, and developing more tutorials to encompass more aspects of lidar and radar meteorology. Based on the positive feedback to our "guided" multi-Doppler analysis tutorial, we plan to upgrade our remaining notebooks to teach users to create their own parameter files independently. New tutorials are in the preliminary stages of development, which would highlight other LROSE applications and explain how to debug common error messages. Finally, we intend to showcase how users can integrate LROSE applications with other open-source packages (e.g., Py-ART, CSU RadarTools [22]).

V. ACKNOWLEDGMENTS

LROSE development is supported by NSF grants #2103776 and #2103785. This material is based upon work supported by the NSF National Center for Atmospheric Research, which is a major facility sponsored by the U.S. National Science Foundation under Cooperative Agreement No. 1852977. NSF Unidata Program Center funding is supported by NSF grant #1901712. This work used NSF Jetstream2 at Indiana University through allocation #EES200002 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

VI. REFERENCES

[1] M. M. Bell, “nsf-lrose/lrose-topaz: lrose-topaz-20220222.” Zenodo, 2022. doi: 10.5281/zenodo.6909479.

[2] “LROSE: The Lidar Radar Open Software Environment.” <http://lrose.net>.

[3] “Main Page.” http://wiki.lrose.net/index.php/Main_Page.

[4] D. Y. Hancock *et al.*, “Jetstream2: Accelerating Cloud Computing via Jetstream,” in *Practice and Experience in Advanced Research Computing (PEARC '21)*, New York, NY, USA: Association for Computing Machinery, 2021, pp. 1–8. doi: 10.1145/3437359.3465565.

[5] T. J. Boerner, S. Deems, T. R. Furlani, S. L. Knuth, and J. Towns, “ACCESS: Advancing Innovation: NSF’s Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support,” in *In Practice and Experience in Advanced Research Computing (PEARC '23)*, Portland, OR, USA: Association for Computing Machinery, 2023, pp. 173–176. doi: 10.1145/3569951.3597559.

[6] J. Helmus and S. Collis, “The Python ARM Radar Toolkit (Py-ART), a Library for Working with Weather Radar Data in the Python Programming Language,” *Journal of Open Research Software*, vol. 4, no. 1, p. e25, 2016, doi: 10.5334/jors.119.

[7] M. Heistermann *et al.*, “The Emergence of Open-Source Software for the Weather Radar Community,” *Bulletin of the American Meteorological Society*, vol. 96, no. 1, pp. 117–128, 2015, doi: 10.1175/BAMS-D-13-00240.1.

[8] J. Vivekanandan, D. S. Zrnic, S. M. Ellis, R. Oye, A. V. Ryzhkov, and J. Straka, “Cloud Microphysics Retrieval Using S-Band Dual-Polarization Radar Measurements,” *Bulletin of the American Meteorological Society*, vol. 80, no. 3, pp. 381–388, 1999, doi: 10.1175/1520-0477(1999)080<0381:CMRUSB>2.0.CO;2.

[9] T. Matejka and R. C. Srivastava, “An improved version of the extended velocity-azimuth display analysis of single-doppler radar data,” *Journal of Atmospheric and Oceanic Technology*, vol. 8, no. 4, pp. 453–466, 1991, doi: 10.1175/1520-0426(1991)008<0453:AVOTE>2.0.CO;2.

[10] A. Zonca and R. S. Sinkovits, “Deploying Jupyter Notebooks at scale on XSEDE resources for Science Gateways and workshops,” in *Proceedings of the Practice and Experience on Advanced Research Computing (PEARC '18)*, New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–7. doi: 10.1145/3219104.3219122.

[11] J. Chastang and A. Espinoza, “Unidata Science Gateway.” GitHub, 2017. doi: 10.5065/688s-2w73.

[12] J. Chastang and A. Espinoza, “Advancing Atmospheric Science Education: Customized PyAOS JupyterHubs via the Unidata Science Gateway,” in *Proceedings, 40th Conference on Environmental Information Processing Technologies, 104th AMS Annual Meeting*, Baltimore, Maryland, USA, 2024. doi: 10.6084/m9.figshare.25251655.v1.

[13] J. DeHart, T. Barbero, T.-Y. Cha, M. Bell, M. Dixon, and B. Dolan, “LROSE Science Gateway.” <https://github.com/nsf-lrose/lrose-hub>, 2024.

[14] M. M. Bell, M. T. Montgomery, and K. A. Emanuel, “Air-sea Enthalpy and Momentum Exchange at Major Hurricane Wind Speeds Observed during CBLAST,” *Journal of the Atmospheric Sciences*, vol. 69, no. 11, pp. 3197–3222, 2012, doi: 10.1175/JAS-D-11-0276.1.

[15] R. M. May *et al.*, “MetPy: A Meteorological Python Library for Data Analysis and Visualization,” *Bulletin of the American Meteorological Society*, vol. 103, no. 10, pp. E2273–E2284, 2022, doi: 10.1175/BAMS-D-21-0125.1.

[16] A. Espinoza, J. Chastang, and W. G. Blumberg, “Deploying an Educational JupyterHub for Exploratory Data Analysis, Visualization, and Running Idealized Weather Models on the Jetstream2 Cloud,” in *Gateways 2023*, Pittsburgh, PA, USA, 2023. doi: 10.5281/zenodo.10034606.

[17] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[18] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[19] S. Hoyer and J. Hamman, “Xarray: N-D labeled arrays and datasets in Python,” *Journal of Open Research Software*, vol. 5, no. 1, 2017, doi: 10.5334/jors.148.

[20] Met Office, *Cartopy: a cartographic python library with a Matplotlib interface*. Exeter, Devon, 2010. Available: <https://scitools.org.uk/cartopy>

[21] A. Zonca, “Soft Scaling Kubernetes on Jetstream.” <https://www.zonca.dev/posts/2024-02-02-soft-scaling-kubernetes-jetstream>, Feb. 2024.

[22] T. Lang, B. Dolan, N. Guy, C. Gerlach, and J. Hardin, “CSU-Radarmet/CSU_RadarTools: CSU_RadarTools v1.3.” Zenodo, Feb. 2019. doi: 10.5281/zenodo.2562063.