

# Using the Jetstream Research Cloud to provide Science Gateway resources

Richard Knepper, Eric Coulter, Marlon Pierce, Suresh Marru, Sudhakar Pamidighantam  
 Pervasive Technology Institute  
 Indiana University  
 Bloomington, Indiana, USA  
 Email: rich@iu.edu

**Abstract**—We describe the use of the Jetstream research-cloud, a purpose-built system with the goal of supporting “long-tail” research by providing a flexible, on-demand research infrastructure, to provide scalable back-end resources for science gateways. In addition to providing cloud-like resources for on-demand science, Jetstream offers the capability to instantiate long-running clusters which support science gateways. Science gateways are web-based systems built on computational infrastructure which provide commonly-used tools to a community of users. We created a persistent cluster on the Jetstream system which is connected to the SEAGrid science gateway and provides additional compute resources for a variety of quantum chemistry calculations. We discuss the further application of toolkits provided by the Extreme Science and Engineering Discovery Environment (XSEDE) to build general-purpose clusters on the research cloud.

**Index Terms**—cloud-based computation, science gateways, cluster toolkits.

## I. INTRODUCTION

The Jetstream system is designed to provide a production cloud resource in support of general science and engineering activities in the eXtreme Digital (XD) ecosystem. While the United States National Science Foundation (NSF) has funded a number of high performance computing (HPC) as well as high throughput computing (HTC) resources, but there remains a significant population of researchers who have computational and data analysis needs that fit neither HPC nor HTC resources[6]. The NSF has noted [7]the benefits of increasing diversity in the range of cyberinfrastructure (CI) resources available to researchers. Other efforts on the part of the NSF to provide more support for research in the “long tail of science” include the Comet[8] system and the Wrangler data storage and data analytics system[9]. Jetstream provides a set of virtual host systems with sufficient interconnect speed to allow real HPC jobs to be completed, with management of guests by Openstack and Atmosphere APIs.

The Jetstream resource and planned activities are well-described in [6]. The current system has been implemented at Indiana University (IU) and Texas Advanced Computing Center (TACC). Initial scientific work has begun and 13 scientific projects were allocated in the most recent allocation period for the system, with both elastic and cluster-based activities approved for use of the system. In this paper we review the Jetstream architecture and software environment

and detail the implementation of a persistent cluster in Jetstream in support of the SEAGrid science gateway [10]. We also discuss the general uptake of Jetstream as both gateway provider and as computational backend for gateway environments. We describe the SEAGrid gateway capabilities and software requirements for its backend computational resource. In order to provide scientific software, we made use of the XSEDE National Integration Toolkit (XNIT) [11] to provide a message-passing implementation and chemistry software. The XSEDE Community Resource Integration team developed the job management using TORQUE [5] and built a set of Ansible [3] scripts to configure the cluster. We detail the requirements for preparing cloud-based resources for a cluster installation using these tools. We conclude the paper with a discussion of future implementations of computational backends for Science Gateways.

## II. JETSTREAM OVERVIEW

The Jetstream system is designed to provide general purpose cloud resources for research in a configurable fashion. The system provides on-demand and persistent virtual systems that support a wide range of scientific software in the form of configurable environments.

### A. Service Functions

The Jetstream system as designed supports multiple modalities of use in support of scientific research that are currently not provided within the broader cyberinfrastructure ecosystem. These include: self-serve academic cloud services, based on virtual machines images provided by the user or selected from a library; persistent virtual machine systems which support the delivery of science gateways such as the Galaxy [12] life science research gateway; data movement via the Globus Connect software and authentication via Globus Auth [13]. Jetstream also provides facilities for publishing and sharing virtual machine images via IU’s persistent digital repository, IUScholarWorks, accessible via Digital Object Identifier (DOI)[14]. Access to interactive desktop sessions on Jetstream is available via VNC connections.

Within the cyberinfrastructure ecosystem, the Open Science Grid, the Extreme Science and Engineering Discovery Environment (XSEDE), and other projects provide a broad

range of resources for computational support of research: high-performance computing at large scale, high-memory resources, big data, and visualization systems, but to date no system provides a highly configurable virtualized environment with the capabilities described above. The Jetstream system is designed to provide resources for the “long tail” of science users [15], who frequently need more access to interactive computational resources than they have locally available. Jetstream provides resources for those that need computational resources for a short time, which can then be released to serve other users, while retaining data for future activities. For longer-running requirements, users can make use of the Jetstream allocation systems to provide an ongoing resource over time.

### B. Hardware Configuration

The Jetstream system hardware architecture follows commercial cloud offerings in terms of uptime and availability. Two production systems, one at IU and one at TACC, provide a 100Gbps-linked, distributed cluster infrastructure. A third development and testing system resides at the University of Arizona. This configuration offers “zoning” between the two production centers, similar to that offered in Amazon EC2 resources and elsewhere. A schematic of the Jetstream system is presented in figure 1.

Jetstream provides multiple VM configurations, ranging from “Tiny”: 1 CPU, 2GB of memory, 20GB of storage, allowing as many as 46 concurrent virtual machine instances per node, up to “XXL”: 44 CPUs, 120GB memory, 480GB storage, with one virtual instance per node.[6] “Small”, “Medium”, “Large”, and “X-Large” configurations are also offered, with according sizes. Virtual machines on Jetstream burn XSEDE ‘Service Units’ (SUs) at a rate of  $1SU / (\text{number\_of\_CPU} \times \text{Hour})^{-1}$  [21].

### C. Software Architecture

Jetstream utilizes the Atmosphere[16] software stack for presenting a user interface, managing images, provisioning, monitoring and managing cloud infrastructure. Openstack provides host and virtual machine management, as well as virtual machine filesystem storage, with iRODS software providing replication between sites. Authentication is provided by Globus Auth, and user data transfer between the user’s desktop and virtual machine filesystems is provided by Globus Transfer. A diagram of the Atmosphere implementation on Jetstream is shown in figure 2.

Atmosphere offers a number of features essential to research computing in a cloud context. Identity management services, networking configuration, and security policies are integrated with the software stack. The software provides complete functions for managing virtual machine instances throughout their lifecycle. Finally, data lifecycle management is simplified via the Atmosphere web portal and API. The web portal displays images that can be launched and worked with by the user, shared between users, and new images can be developed and uploaded. Once an image is launched it can be used interactively. A diagram of the Jetstream user interface is presented in figure 3.

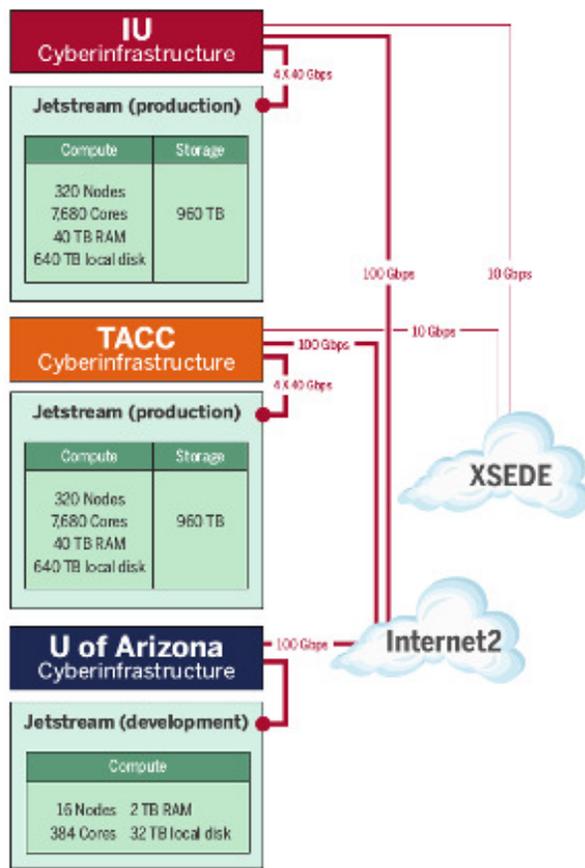


Fig. 1: Jetstream physical sites and network connectivity

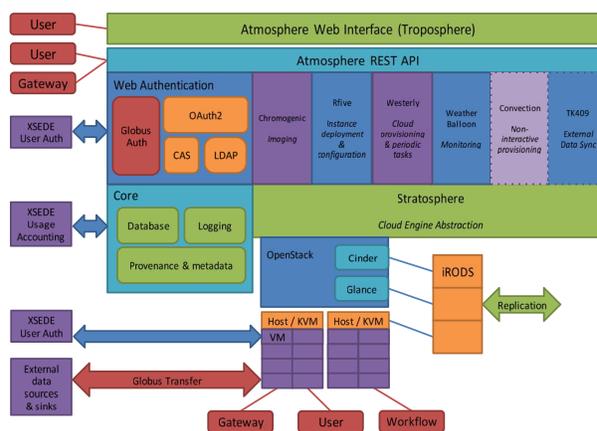


Fig. 2: Atmosphere Architecture in Jetstream

The Openstack API and Horizon Dashboard are also optionally exposed to some users who require greater customization of their virtual environment than is available through Atmosphere. This allows for direct interaction with the Openstack core services such as keystone, nova, neutron, and cinder, for scripted or command-line configuration of identity management, compute resources, networking, and storage volumes. The Heat service allows for simple YAML configuration files to be used in specifying an entire “stack” of components.

#### D. Uptake of Jetstream as a Science Gateway hosting environment

The Jetstream project description when originally proposed hinged on the implementation of Jetstream VM’s as back-end computational infrastructure for science gateways. The intent of providing these resources was to leverage already-existing demand in order to supply a community group with little to no transition costs for switching to Jetstream from their previous back-end systems.

The Jetstream environment has proven to be fertile ground for hosting Science Gateways from a variety of disciplines. The Chemcompute project augments local resources at Sonoma State University to allow users to run jobs with the GAMESS and TINKER packages on Jetstream, using a single node[18], as well as resources on SDSC’s Comet and PSC’s Bridges systems. The SEAGrid and SciGap projects already make use of Jetstream for hosting the front-end web-server portions of their science gateway projects. The ATLAS project uses Jetstream to extend their available compute resources for particle physics simulations and data analysis, through a gateway-like workflow. The CIPRES Gateway for systematic and population biology currently allows users to run a variety of computationally intense codes on XSEDE resources, including Jetstream, through a point-and-click web interface that provides access to phylogenetic reconstruction

Project	SUs Spent
GenApp	4,383,727
ATLAS	1,615,493
Galaxy	1,578,590
Galaxy Gateway Infrastructure Development	533,577
SciGap	253,329
SEAGrid	132,362
<b>Total Gateways</b>	<b>8,497,078</b>
<b>Total Jetstream</b>	<b>13,067,258</b>

TABLE I: Major gateway projects utilizing Jetstream resources, both for computational and infrastructure purposes. SUs on Jetstream are used up at the rate of  $1 \text{ SU} \times (\text{number\_of\_CPUs} \times \text{Hour})^{-1}$ . This data was taken on Feb 20th, 2017 via the XDMoD utility[20], and spans the total lifetime of the Jetstream project.

software [19]. The Neuroscience Gateway project, which has been using XSEDE resources for 5 years, is expanding to use Jetstream as a computational resource in the second round of XSEDE to conduct computational neuroscience simulations. Emre Brookes at University of Texas Health Science Center at San Antonio has expanded the Genapp project, which leverages the Airavata framework to allow users to submit jobs either from a web portal or GUI, to use Jetstream as an extensible computational resource. This project has become one of the largest (by SUs spent) projects currently running on Jetstream I.

One of the largest groups of gateway users on Jetstream comes from the Galaxy Project. The Galaxy project is a workflow-based science gateway for biomedical research. Through Jetstream, users can create their own multi-user Galaxy instances, or use Jetstream machines as computational resource through the main Galaxy page. A particularly interesting application, enabled by the Galaxy Gateway Infrastructure Development project, is the ability for a user to spin up a small Galaxy master, configure it to support their particular workflows (using Jetstream or other resources for computational power), and share/replicate as needed amongst their colleagues. The Galaxy Gateway Infrastructure Development project is led by Enis Afghan at Johns Hopkins University. With the addition of DOI-tagging on Jetstream VMs, this becomes a very powerful scientific tool in the biomedical research world.

Usage of Jetstream via Gateway environments has been quite strong, as can be seen in Table I. Gateway usage to date has been about 65% of total Jetstream usage. Some of the projects mentioned here have only begun to utilize the power of this fully operational cloud environment, and several new gateway-oriented allocations have been granted as well, such as the extension to the SCEC Petascale Research project led by Thomas Jordan at the University of Southern California,

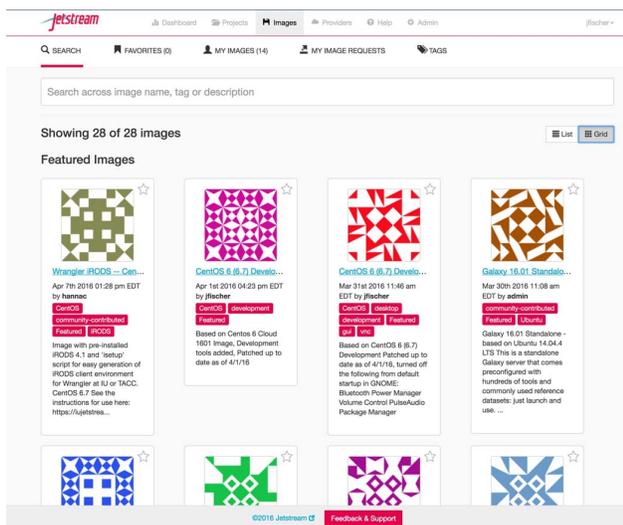


Fig. 3: Jetstream User Interface

which develops physics-based predictive models of earthquake rupture and ground motion processes and improves existing seismic hazard computational methods.

### III. SEAGRID ARCHITECTURE

SEAGRID[10] is a science gateway for chemistry and molecular dynamics research, which provides a single point of access to resources via a desktop client, which communicates with a middleware system. The middleware system, Apache Airavata [17], manages job submissions on batch scheduled HPC resources. SEAGrid also provides a browser-based client at SEAGrid.org which accesses the same middleware. The client and browser application both allow access to resources allocated via XSEDE or via other local allocations. SEAGrid leverages XSEDE computing resources, and has enabled over 120 publications and hundreds of projects.

The architecture of Apache Airavata manages worker services that allow users to compose, launch, and monitor workflows on a variety of computing infrastructure. It has been developed with a particular eye towards enabling science gateways, which allow users to focus on performing calculations without managing the peculiarities of different computing resources. Airavata requires nothing more than an authorized user on a compute resource, and can submit jobs to a variety of schedulers. It does require a static address to access for each resource, and luckily Jetstream was designed with long-term persistent VMs in mind. While it would be possible to provision dynamic computing resources behind a persistent headnode, for the initial project we made the entire cluster static.

### IV. IMPLEMENTING PERSISTENT CLUSTERS IN JETSTREAM

In implementing persistent clusters on Jetstream, a few main goals were identified as requirements that served both the needs of the SEAGrid community as well as providing simple management on the part of the XSEDE Community Resource Integration team without undue development requirements:

- Easily created/destroyed with minimal hands-on work to the VMs
- Easily configured for different gateways
  - Simple scheduler configuration
  - Simple configuration of users
  - Easy to add/change needed scientific software
- Allow for longer execution times than available on other resources

To these ends, we chose to use the configuration management tool Ansible for the final configuration of a live set of VM instances. The initial VMs are created through the implementation of a small number of Heat templates. While these allow for a great deal of customization, it was necessary to handle additional steps such as scheduler configuration and building some scientific software after the VMs are started. A series of Ansible “roles” [2] were developed to mimic the basic structure of a standard MPI-oriented cluster, handling separate tasks such as defining users, sharing nfs mounts, installing and configuring the scheduler, and building scientific software. The

cluster has a basic architecture of a single head-node, based on a ‘Small’ VM, controlling a number of compute nodes, on ‘Large’ VMs, on a private network. For most scientific software, we leverage the existing resources available in the XNIT when possible, as well as using the pre-built versions of MPI. In the case of SEAGrid, we added the step of building Quantum Espresso [4] from source, as it was not yet available in the XNIT.

Before a new “virtual cluster” (VC) is built, a few things are specified in configuration files, primarily to allow for greater flexibility when new users or groups ask for a new VC. The main details have to do with specifying the allocation under which the VC is created, and which users need to be added to the allocation. Public SSH keys for those users must also be added to the Ansible role in order to allow access, and a list of scientific applications can be edited in the Ansible script itself. The current setup uses the Torque resource manager with a simple “first-in, first-out” (FIFO) scheduler, although a basic Slurm [1] setup is also an option.

Currently, it is possible to spin up a working VC for a new group in a matter of minutes, provided that no additional software outside of the XNIT must be included, which already includes a wide range of software. A detailed list of XNIT software is available at the XSEDE Community Software Repository The Ansible roles, heat templates, and example deployment scripts are available in a public github repository.

### V. CONCLUSION

We have demonstrated the capability to setup a persistent system for SEAGrid, which is currently in production, and refined the process to the point that setting up additional resources is a matter of gathering requirements from the user, and running a handful of commands. This substantially contributes to the availability of resources for users of science gateways, in the case that there are not sufficient computational resources available for the community. This also makes a new class of compute resource available through Jetstream, rather than a single instance model. For resource-constrained groups needing long-term HPC-style compute resources, the Jetstream-based VC is an effectively free resource that can be used for both research and education, and also offers a new solution a problem often faced by quantum chemistry researchers in particular, where jobs often need more runtime than is allowed on many resources. This project also substantially developed the XSEDE Community Resource Integration toolkit, helping the team adapt it for research cloud deployment, and allows campus cyberinfrastructure professionals the option of trying the XNIT stack before they implement locally, increasing adoption and familiarity with the toolkit. In the future, this resource will be further developed to allow for dynamic allocation of compute nodes based on job size, since a persistent cluster will more rapidly burn through a Jetstream project allocation, as Jetstream projects are based on the total time VMs are running. The XSEDE Community Resource Integration team will work closely with the Science Gateways Research Center in order to extend this capability and provide more flexible resources for new potential science gateway communities.

## REFERENCES

- [1] "Slurm workload manager," 2016. [Online]. Available: <https://slurm.schedmd.com/>. Accessed: Jan. 9, 2017.
- [2] "Playbook roles and include statements Ansible documentation," 2016. [Online]. Available: [http://docs.ansible.com/ansible/playbooks\\_roles.html](http://docs.ansible.com/ansible/playbooks_roles.html). Accessed: Jan. 9, 2017.
- [3] Red Hat, "Ansible is simple IT automation," 2016. [Online]. Available: <https://www.ansible.com/>. Accessed: Jan. 9, 2017.
- [4] Giannozzi, P. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials *J. Phys.: Condens. Matter* 2009, 21, 395502, DOI: 10.1088/0953-8984/21/39/395502
- [5] Adaptive Computing, "TORQUE resource manager," 2017. [Online]. Available: <http://www.adaptivecomputing.com/products/open-source/torque/>. Accessed: Jan. 9, 2017.
- [6] Stewart, Craig A., et al. "Jetstream: A self-provisioned, scalable science and engineering cloud environment." Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure. 2015. ACM: St. Louis, MO, USA. p. 29-37.
- [7] Committee on Future Directions for NSF Advanced Computing Infrastructure to Support U.S. Science in 2017-2020. 2014. Future Directions for NSF Advanced Computing Infrastructure to Support U.S. Science and Engineering in 2017- 2020: Interim Report. Washington, DC. The National Academies Press. 48 pp. <http://www.nap.edu/catalog/18972/future-directions-for-nsf-advanced-computing-infrastructure-to-support-us-science-and-engineering-in-2017-2020>
- [8] Moore, R.L., C. Baru, D. Baxter, G. Fox, A. Majumdar, P. Papadopoulos, W. Pfeiffer, R.S. Sinkovits, S. Strande, M. Tatineni, R.P. Wagner, N. Wilkins-Diehr, M.L. Norman. Gateways to Discovery: Cyberinfrastructure for the Long Tail of Science. In Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment. 2014. ACM: Atlanta, GA, USA. p. 1-8.
- [9] Texas Advanced Computing Center. Wrangler. <https://www.tacc.utexas.edu/~wrangler-data-intensive-system-opens-to-scientists>
- [10] S. Pamidighantama, S. Nakandala, E. Abeysinghe, C. Wimalasena, S. Rathnayakae, S. Marru, M. Pierce, Community Science Exemplars in SEAGrid Science Gateway: Apache Airavata Based Implementation of Advanced Infrastructure, *Procedia Computer Science* Volume 80, 2016, Pages 1927-1939
- [11] Fischer, Jeremy, Richard Knepper, Matthew Standish, Craig A. Stewart, Resa Alvord, David Lifka, Barbara Hallock, and Victor Hazlewood. "Methods For Creating XSEDE Compatible Clusters." In Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment, p. 74. ACM, 2014.
- [12] Goecks, J. A. Nekrutenko, J. Taylor, and The Galaxy Team. 2010. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* 11(8):R86. <http://genomebiology.com/2010/11/8/R86>
- [13] Foster I, Kesselman C. Globus: A metacomputing infrastructure toolkit. *International Journal of High Performance Computing Applications.* 1997 Jun 1;11(2):115-28.
- [14] Bobay J. Institutional Repositories: Why Go There?. *Indiana Libraries.* 2008 Jan 1;27(1):7-9
- [15] Heidorn, P.B. Shedding Light on the Dark Data in the Long Tail of Science. *Library Trends.* 2008. 57(2), p. 280-299. [http://muse.jhu.edu/journals/library\\_trends/v057/57.2.heidorn.html](http://muse.jhu.edu/journals/library_trends/v057/57.2.heidorn.html)
- [16] Skidmore, E., S.-j. Kim, S. Kuchimanchi, S. Singaram, N. Merchant, and D. Stanzione. iPlant Atmosphere: A Gateway to Cloud Infrastructure for the Plant Sciences. In Proceedings of the 2011 ACM workshop on Gateway computing environments. 2011, ACM: Seattle, Washington, USA. p. 59-64. <http://dl.acm.org/citation.cfm?id=2110495>
- [17] Marru S, Gunathilake L, Herath C, Tangchaisin P, Pierce M, Mattmann C, Singh R, Gunarathne T, Chinthaka E, Gardler R, Slominski A. Apache airavata: a framework for distributed applications and computational workflows. In Proceedings of the 2011 ACM workshop on Gateway computing environments 2011 Nov 18 (pp. 21-28). ACM.
- [18] Perri, M. J., and S. H. Weber. "Web-Based Job Submission Interface for the GAMESS Computational Chemistry Program." *Journal of Chemical Education* 91.12 (2014): 2206-2208.
- [19] Miller, M.A., Pfeiffer, W., and Schwartz, T. (2010) "Creating the CIPRES Science Gateway for inference of large phylogenetic trees" in Proceedings of the Gateway Computing Environments Workshop (GCE), 14 Nov. 2010, New Orleans, LA pp 1 - 8.
- [20] Jeffrey T. Palmer, Steven M. Gallo, Thomas R. Furlani, Matthew D. Jones, Robert L. DeLeon, Joseph P. White, Nikolay Simakov, Abani K. Patra, Jeanette Spherhac, Thomas Yearke, Ryan Rathsam, Martins Innus, Cynthia D. Cornelius, James C. Browne, William L. Barth, Richard T. Evans, "Open XDMoD: A Tool for the Comprehensive Management of High-Performance Computing Resources", *Computing in Science & Engineering* 17.4 (2015):52-62, 2015 10.1109/MCSE.2015.68
- [21] "Jetstream User Guide," 2016. [Online]. Available: <https://portal.tacc.utexas.edu/user-guides/jetstream>. Accessed: Feb. 20, 2017.