




SPECIAL ISSUE PAPER

Jetstream—Early operations performance, adoption, and impacts

David Y. Hancock¹ | Craig A. Stewart¹ | Matthew Vaughn² | Jeremy Fischer¹ |
John Michael Lowe¹ | George Turner¹ | Tyson L. Swetnam³ | Tyler K. Chafin⁴ |
Enis Afgan⁵ | Marlon E. Pierce¹ | Winona Snapp-Childs¹ 

¹Pervasive Technology Institute, Indiana University, Bloomington, IN 47408, USA

²Texas Advanced Computing Center, University of Texas at Austin, Austin, TX 78758, USA

³BIO5 Institute, CyVerse, University of Arizona, Tucson, AZ 85721, USA

⁴Department of Biological Sciences, University of Arkansas, Fayetteville, AR 72704, USA

⁵Department of Biology, Johns Hopkins University, Baltimore, MD 21212, USA

Correspondence

David Y. Hancock, Pervasive Technology Institute, Indiana University, Bloomington, IN 47408, USA.

Email: dyhancoc@iu.edu

Craig A. Stewart, Pervasive Technology Institute, Indiana University, Bloomington, IN 47408, USA.

Email: stewart@indiana.edu

Funding information

National Science Foundation, Grant/Award Number: 1445604

Summary

Jetstream is a first of its kind system for the NSF — a distributed production cloud resource. We review the purpose for creating Jetstream, discuss Jetstream's key characteristics, describe our experiences from the first year of maintaining an OpenStack-based cloud environment, and share some of the early scientific impacts achieved by Jetstream users. Jetstream offers a unique capability within the XSEDE-supported US national cyberinfrastructure, delivering interactive virtual machines (VMs) via the Atmosphere interface. As a multi-region deployment that operates as an integrated system, Jetstream is proving effective in supporting modes and disciplines of research traditionally underrepresented on larger XSEDE-supported clusters and supercomputers. Already, Jetstream has been used to perform research and education in biology, biochemistry, atmospheric science, earth science, and computer science.

KEYWORDS

cloud computing, long tail of science, OpenStack, science impacts, user adoption, virtual machines

1 | INTRODUCTION

Jetstream is designed to deliver the services and programming models needed by researchers working in the “long tail of science.” It serves as the prime example of programmable cyberinfrastructure (CI) within the National Science Foundation (NSF), a flexible, highly customizable, highly configurable resource that is designed to evolve with researchers' needs. It is capable of delivering services in a way that is and is perceived to be easy, accessible, and valuable. In particular, Jetstream: 1) offers “self-serve” cloud services, enabling researchers (students, staff, and faculty) to select a preexisting VM image or to create a new environment for personalized research computing; 2) hosts persistent Science Gateways; 3) enables data movement, storage, and dissemination; 4) provides virtual desktop services to tablet devices, increasing CI access for users at resource-limited institutions. The vision for Jetstream¹ is straightforward: Jetstream is a managed science and engineering cloud — a cloud managed and operated in order to support open science and engineering research. Jetstream, as a system, and the Jetstream team, as a management and support group, complement existing NSF-funded CI resources supported by XSEDE (the eXtreme Science and Engineering Discovery Environment). In particular, Jetstream aims to provide resources for interactive use any time a handful of processor cores are needed and for programmatic large-scale computational use through science gateways or during “non-peak” hours.

Here, we present information on Jetstream's capabilities and its implementation process. We also discuss usage through Jetstream's first year of operations for the national research community as well as educational and early science impacts that have been achieved by end users and the Jetstream team. Our objective is that Jetstream be known for the distinctive research results and training outcomes it has enabled.

1.1 | Use cases

As a first step in planning to deliver “innovative computational resources to an increasingly diverse community,” we interviewed dozens of researchers, mostly working in areas underrepresented among existing users of resources in the NSF eXtreme Digital (XD) program. (We follow NSF usage in using “XD Program” to refer collectively to XSEDE and the NSF-funded resource providers that manage and deliver resources allocated and supported by XSEDE under NSF direction). On the basis of these interviews, we defined two canonical use cases for domain scientists and a number of use cases based on mode of use. All use-cases are designed to meet the needs of researchers who can be described generally as working in the “long tail of science” – a label that applies to many researchers working in areas supported by the NSF but not strongly represented in XD program usage.² These canonical use cases are described briefly below and in greater detail alongside the mode of use scenarios in a paper presented at XSEDE15.¹

The following describes the two generalizations of domain-based use cases derived from these interviews. The most common use case we heard was: A researcher wants straightforward access to specific, usually interactive, tools to analyze data delivered in a manner congruent with their normal operations and often driven by availability of new data. A related and also common use case we heard, generally from scientists who are developing new software was: A tool producer develops new analysis routines and methods to address research bottlenecks and needs to make said tool available to experimentalists without having them contend with technical complexities of operating system and software dependencies. Proposed solution: Develop an accessible platform where application creators can easily publish and share within a VM image and end users can easily invoke runnable instances of these applications via virtualization. The science impacts described in Section 5.4 on Watershed Management and Ecology both highlight the first of these generalized use cases. The impact on Genome Annotation also included in Section 5.4 is a good example of the latter use case on software improvement and distribution to a specific community. The science gateway descriptions, and specifically Galaxy,³ provide a prime example that demonstrates both generalized use cases in Section 5.5.

2 | SYSTEM IMPLEMENTATION

Jetstream consists of three components: Jetstream-IU, Jetstream-TACC, and Jetstream-AZ. We designed Jetstream to deliver availability and reliability in a way that would mimic commercial cloud experiences and be manageable on an acquisition and implementation budget of less than \$8M. Jetstream-AZ is housed at the University of Arizona and serves as a test and development environment. The two production components of Jetstream are located in two highly secure and robust data centers in two different geographical regions of the US. They are Jetstream-IU (operated by the Indiana University Pervasive Technology Institute and located on the IU-Bloomington campus) and Jetstream-TACC (operated by the Texas Advanced Computing Center on the campus of the University of Texas at Austin). These are identical and each consists of compute nodes (320 Dell M630 blades with 128-GB RAM), storage nodes (20 Dell R730 servers with 12 4-TB disks), and management nodes (7 Dell R630 servers). The Dell PowerEdge servers are interconnected with a 10/40 Gb/s Fat-Tree Ethernet network, oversubscribed 2:1 ratio (see the work of Stewart et al⁴ for details).

The software environment is built on OpenStack.⁵ OpenStack is a sophisticated infrastructure delivery platform capable of provisioning many different types of infrastructure with RESTful microservice APIs. For each of the two production environments, a minimum of six databases are used to persist state for the OpenStack services. Each production site has at least 620 services or agents running on 288 hosts configured with 1138 unique configuration files. This degree of complexity across two administrative domains required adopting a DevOps style of deployment where the OpenStack infrastructure is deployed continuously in a prescriptive fashion. IU used SaltStack⁶ in previous projects, while TACC had used Puppet.⁷ Each site used the configuration management tool that was most familiar to administrators and aligned with existing practices. System administrators transcribe configuration changes between the two configuration management systems as needed. OpenStack has a major release every six months and the end of life for a major release is typically eighteen months after the initial release. It is not possible to skip a major release during an upgrade, so it is essential, for continuity of service, to follow the major release cycle. As a consequence, the system is effectively redeployed every six months with significant configuration changes required between OpenStack releases.

The acceptance review report originally submitted to the NSF is available online.⁸ After Jetstream was accepted in May 2016, the Jetstream team performed additional network and storage testing with nuttcp and IOR as requested by an NSF review panel.⁹ The original and updated data were previously discussed in detail.⁴ The NSF¹⁰ specified several criteria that relate to basic integration of a CI resource with XSEDE, all of which were fulfilled while the hardware performance tests were being completed. Jetstream became available to the national research community in “Early Operations” mode on February 10, 2016. On May 27, 2016, the NSF granted Indiana University authority to pay our vendor partner, Dell Inc., for the Jetstream system hardware, confirmation that the system was implemented as proposed in the terms of our cooperative agreement, scope of work, and project execution plan.

3 | JETSTREAM: SIMULTANEOUSLY FIRST OF ITS KIND PILOT SYSTEM AND PRODUCTION SYSTEM

The NSF has funded three previous experimental grids and cloud systems—FutureGrid, CloudLab, and Chameleon.¹¹⁻¹³ As previously stated, Jetstream is a first of its kind acquisition and implementation for the NSF within the NSF-funded national CI. Because of that and the modest

implementation budget, the system is, in a sense, a pilot experiment of a production cloud system. Even post-acceptance, Jetstream continues to evolve as a first of its kind pilot environment. Merely demonstrating integrated cloud functions is not fully satisfactory in an ever-evolving cloud ecosystem. Throughout each year of production operations, Jetstream will continue to pilot new features available through the Atmosphere-based Jetstream portal and native-OpenStack application programming interfaces (APIs).¹⁴

Cloud Functionality Tests. Working with the NSF and developing a program execution plan (PEP) that was peer reviewed by outside experts, we arrived at the following tests of integrated cloud functionality of the system. As described in the PEP, Jetstream would be a “configurable large-scale computing resource that leverages both on-demand and persistent virtual machine technology to support a much wider array of software environments and services than current NSF resources can accommodate.” The cloud functionality tests had to support the four key functions described in Section 1. The first of these functions we committed to deliver was to provide academic self-serve cloud services that enable users to select a VM image from a published library or to create or customize their own virtual environment for personalized research computing with authentication via Globus.¹⁵ As agreed by NSF and the Jetstream team, proof of functionality was demonstrated by successfully accomplishing a series of specific tasks described in the acceptance report.⁸ An updated version of that report was also produced that includes additional tests requested by the NSF review panel.⁹

Gateway Functionality Tests. The Jetstream PEP also included functionality tests for supporting science gateways, developed by agreement between the NSF and the Jetstream team. We committed to delivering support for persistent science gateways, including the capability of hosting persistent science gateways within a VM when the nature of the gateway is consistent with VM operation. Galaxy was one of the initial science gateways supported. Proof of functionality is included in the published acceptance report.⁸








Data Movement, Storage, and Dissemination. The Jetstream PEP also included functionality tests for supporting data movement, storage, and dissemination. The function we committed to deliver was data movement, storage, and dissemination using Globus Connect for the data movement and IUScholarWorks for dissemination to obtain a Digital Object Identifier (DOI) associated with the VM stored.¹⁶ Proof of functionality tests were demonstrated live for the NSF review panel and documented via screenshots.⁸

4 | OPERATIONAL METRICS

In April 2016, we collected the metrics specified by the NSF for all production CI systems in the XD program and additional metrics agreed upon between the NSF and the Jetstream team. We now have the opportunity to examine those results as of the end of PY1, shown in Table 1. The Jetstream team and the NSF had to determine how to interpret and apply some of the metrics to a cloud system. A full discussion of the metric definitions and data from early operations was presented at XSEDE16.⁴

CPU Utilization. Perhaps the most challenging metric to transpose into a cloud setting – and the one metric that remains open as a topic of discussion between the Jetstream team and the NSF – is the concept of the busyness of the system. For a cloud environment, one must consider some of the core functions, the ability for on-demand self-service, as well as rapid elasticity.¹⁷ Additionally, with a compute-focused cloud environment such as Jetstream, the system must be responsive to high I/O utilization via the network, bursts of intensive CPU utilization, as well as a small number of VMs occupying the majority of a physical host. Traditional cloud environments typically gain efficiency through over-subscription, which would negatively impact the user experience. It is worth noting that even with significant over-subscription, a commercial cloud environment spends the bulk of time in the 10-50% CPU utilization range.¹⁸ Our approach to date has been to optimize for the productivity of individual Jetstream users knowing that the system is funded under the “High Performance Computing System Acquisition” track of the NSF and that many users will exhaust the per node memory prior to saturating the processors.

TABLE 1 Operational metrics for Jetstream PY1

Metric	Goal	Achieved	Outcome
System availability (uptime of the production hardware as % of wall clock time)	95%	100%	
Capacity availability (% of total capacity of Jetstream available for use over time)	95%	99.4%	
Job completion success—featured VM launches that reported status to Atmosphere as active	96%	97.7%	
Total number of distinct users	1000	1921	
Use—mean number of VMs active per day	320	mean:512 peak:790	
CPU % utilization	6%	mean:4.2% peak:21.6%	
VM images published with a DOI via IUScholarWorks	10	11	

Note: G indicates a metric is being met. Y indicates 60% of target, a metric for focused attention and effort.

We commonly hear discussion of cluster and supercomputer operations at more than 80% or 90% utilization. These statistics are really measures of the extent to which a resource is occupied with jobs, how much of the available wall-clock time is utilized. They do not reflect raw CPU utilization, and those data are typically not published. Through an examination of a local shared general-purpose cluster resource at IU over a period from December 2014 through May 2017, we found that real CPU utilization averaged 50.08% of the wall-time utilization reported by the resource manager. In other words, if your cluster is 80% utilized in the traditional sense, the processors are likely averaging 40% utilization. This real utilization may be even lower on systems with accelerators or co-processors where a concerted effort is required to effectively utilize the hardware. Ultimately, we will continue to report these data as required, but we believe that the focus of a system such as Jetstream should be on enabling use-cases that are not possible or not practical on traditional HPC environments.

5 | DEMONSTRATIONS OF UTILITY

5.1 | Use and allocations

By the end of April 2016, just before the initial NSF review, 327 different people had used Jetstream. Of those initial 327 people, 159 were “end-user researchers or students” and 168 were staff. At the end of Project Year 1 (PY1) operations, which covered the period from June 2016 through May 2017, that number had grown to 1921. At that time, the user base consisted of 464 center staff/Campus Champions, 574 students, and 861 researchers, as well as a handful of users of undisclosed status. The larger demographic picture at that time shows 54 fields of science were represented through 298 active XSEDE projects with 165 distinct institutions using the system. Jetstream SUs (XSEDE Service Units) are based on virtual CPU hours with varying VM sizes. One SU is one wall clock hour for one vCPU and associated RAM and storage. A vCPU on Jetstream is a single hyper-threaded Intel Xeon Haswell processor core, meaning they are much more powerful than the vCPU on a standard VM designed for traditional web services. Table 2 shows a breakdown of allocations and SUs by area of science as of the end of PY1. These data illuminate the continued progress toward increasing the diversity of users and uses of resources of the XD program through Jetstream. Relative to typical allocations on large clusters supported by XSEDE, Jetstream allocations show much more interest on the part of biologists (working in areas other than molecular biosciences) than XSEDE as a whole. Principal Investigators with allocations represent 42 states, Puerto Rico, and the District of Columbia. There are allocations to PIs in 19 EPSCoR jurisdictions (including Puerto Rico). There are also allocations to PIs at four Minority Serving Institutions. As of May 2017, 86% of Jetstream users had never executed a job on another XSEDE resource, a higher percentage of new users than other XSEDE systems. Taken as a whole, these data demonstrate success in attracting unreached communities and clear progress toward our stated vision.

5.2 | Technology Adoption, Community Support

Through the early operations phase and PY1, virtual organizations (VOs) and XSEDE Campus Champions have assisted with disseminating information about Jetstream. VOs and Campus Champions have been key to the educational and scientific impacts of Jetstream to date. Included in the operations budget for Jetstream are the creation of Cornell Virtual Workshops because XSEDE training capabilities for specific resources are limited. The first workshop, released in PY1, focused on the application process for XSEDE resources, specifically Jetstream.¹⁹ The workshop

TABLE 2 Distribution of allocations by discipline for Jetstream PY1 and for all other systems supported and allocated via XSEDE

Discipline or Area of Interest	# of Jetstream Allocations	SUs allocated on Jetstream	% of SUs Allocated on Jetstream	% of all SUs Allocated on Other XSEDE-Supported Systems
Astronomy	2	1,108,096	3.04%	8.61%
Atmospheric Sciences	4	2,752,400	7.55%	3.73%
Biological Sciences	57	5,199,000	14.27%	4.95%
Campus/Domain Champions	123	6,105,500	16.76%	0.09%
Computational Science	11	1,150,000	3.16%	0.92%
Computer Science	15	4,944,302	13.57%	1.8%
Education Allocations	24	2,847,600	7.82%	0.01%
Engineering	1	100,000	0.27%	3.81%
Geosciences	10	1,978,400	5.43%	2.87%
Humanities/Social Sciences	10	560,000	1.54%	0.45%
Molecular Biosciences	8	4,647,520	12.75%	17.65%
Network Science	3	200,000	0.55%	0.06%
Ocean Science	3	230,000	0.63%	1.30%
Physics	4	2,252,400	6.18%	16.43%
Training & Development	11	2,362,000	6.48%	0.16%

complements the getting started material that introduces the environment and the startup allocation process for Jetstream.²⁰ Over time, we expect assistance from XSEDE Extended Collaborative Support Services with in-depth engagements and science gateway development. However, XSEDE support is not funded to assist in resolving day-to-day resource-specific issues for users. Toward the end of PY1, the Jetstream team received a supplement to accelerate the adoption of the resource and added a team member that will be focused on education, outreach, and training for the next two years; this position complements IU-funded support efforts and fills a gap between XSEDE and the Jetstream team.

To on-board users, Jetstream uses the XSEDE allocations process exclusively. Users must be on a valid XSEDE allocation to gain access. The Jetstream team has made adoption of the resource for new users a priority and has made it easier for those users through the introduction of trial allocations that allow users to “test drive” Jetstream before deciding whether to put forth the effort to apply for a startup, educational, or research allocation. Trial allocations began in beta at the end of PY1, and in the first 78 days, 71 users have taken advantage of this option. The Jetstream team would like to see the account creation and propagation processes within XSEDE reduced from a few hours to a few minutes to further improve the user experience and make trial allocations viable for workshop use. The expectation is that, over time, users on trial allocations will begin to request startup or educational allocations and eventually research allocations if their computational needs increase in size or duration. Trial allocations are also bolstered by the ease at which users can authenticate to Jetstream through integration with Globus Online.

Once users are on a valid allocation, they access Jetstream via the Atmosphere web portal or optionally via the Jetstream API after requesting access. The Jetstream team elected to partner with the Globus team for authentication and authorization features available within Globus Online. Authentication for Atmosphere is done via Globus Auth, so no user credentials are actually stored on Jetstream itself. At the start of our early operations phase, users were required to authenticate to the Jetstream portal with their XSEDE portal identity. An enhancement during PY1 now allows users to associate their XSEDE identity with campus identity providers through use of CILogon or to leverage Google or ORCID identities if their campus is not one of the more than 150 participating organizations.^{21,22} After users have logged in to the Atmosphere portal, researchers may select from a library of “Featured Images,” images maintained by the Jetstream team. These images are mostly images with common development tools that users may build on for their customized workflows. There are also images with specialized software such as the Intel compilers and profiling tools, R, R Studio, and Matlab. The Intel compilers and Matlab are commercially licensed software that are available using licenses from Indiana University and TACC with permission from Intel and Mathworks.

5.3 | Education Usage, Survey, and Impacts

The Jetstream team has also made a concerted effort to engage educators to use Jetstream for their workshops, tutorials, and courses. XSEDE offers an education allocation that is intended for this use as a means to differentiate educational use from research and development use. As Table 2 shows, for PY1, almost 8% of Jetstream's allocated SUs were for educational purposes as compared to 0.01% over all XSEDE allocations. This demonstrates the positive results from the efforts put into showing Jetstream as a useful tool for educational purposes.

Tools for Growth: Jupyter Notebooks. One key method of engaging educational use of Jetstream is working with those helping to create the tools educators use. A very popular tool in recent times is Jupyter Notebooks. The Jetstream team has been actively working with a team from Berkeley that supports the Jupyter development team as well as developing a means to package it with a containerized auto-scaling environment (Kubernetes) for using it efficiently.^{23,24} These informal partnerships help the development team extend the reach of their product and provide a means of obtaining feedback directly from users – they also benefit the educators that use these tools.

Jetstream Education Survey. In March 2017, a survey was conducted among the PIs, Co-PIs, and key personnel of the education allocations that had been used between March 2016 and March 2017. This group included 23 allocations. Out of those queried, nine individuals agreed to participate and did so with informed consent. In-depth course and workshop reviews are covered in a paper presented at SIGUCCS17.²⁵ The primary goal in pursuing this survey was to explore “how Jetstream was used, what went well, what could be improved, and if there were ways that educators felt cloud services in general could be made more attractive for classroom use.” While the sample size itself is not large, the results show a broad user base and diverse topics. Ultimately, 100% of the respondents felt that their educational use of Jetstream was a success. Constructive criticism was given on ways to improve Jetstream for use as an educational platform, and Jetstream team will use that feedback to help expand the usability and reach of Jetstream for STEM education, training, and course work. As of July 31, 2017, there were 29 active education allocations on Jetstream. While a growth of five allocations in the 60 day span from the end of PY1 seems modest, it is a 21% growth. As the Jetstream Education, Outreach, and Training (EOT) staff continue to present at conferences, participate in workshop events, and engage educators directly, we expect to see this trend of using Jetstream to train current researchers as well as the next generation of researchers grow.

5.4 | Science Impacts

Perhaps the strongest sign of Jetstream's ability is the multitude of analyses performed that will accelerate the submission of scientific technical reports to peer-reviewed journals. Early results involve several priority research areas from our initial proposal – genomics and field biology, psychology, computer and computational science. Here, we discuss a subset of the science impacts to date.

Watershed Management. Mountains are vital to ecosystems given their influence on global carbon and water cycles; however, the extent to which topography regulates mountain forest carbon uptake and storage is not well understood. In Colorado, Swetnam et al²⁶ explored topographic versus climatic variation in forest ecosystems using aerial lidar (a surveying method) to make 3D-representations of the landscape and forest. This work

required significant computational power. Analyses in a GIS as well as R and R-Studio²⁷ were powered by Jetstream cloud computing cyberinfrastructure. R notebooks are available in their supporting materials; project data, additional code, and a wiki are maintained using CyVerse.²⁸ These results have improved researchers' understanding of forest drought stress, mortality, and overall carbon sequestration in complex terrain.

Additional advanced research is being carried out in the century-old Santa Rita Experimental Range (SRER) [Est. 1902], in southern Arizona. SRER is a National Ecological Observatory Network (NEON) Core site with an anticipated program period of 30 years. Data will be collected continuously by NEON at monitoring locations across the SRER, and annual overflights will be conducted by the Airborne Observatory Platform (AOP) across the entire SRER (~21,000 hectares). The AOP collects (1) lidar, (2) Red-Green-Blue (RGB) photogrammetry, and (3) hyperspectral data cubes measuring reflected light energy in 428 narrow spectral bands extending from 380 to 2510 nm. In addition, data are being collected at a much higher temporal resolution using small unmanned aircraft systems (sUAS) flown over monitoring plots for RGB, hyperspectral, and structure from motion (Sfm) photogrammetry.²⁹ These data complement the ongoing NEON effort by providing phenological information and structural change detection at two to three orders of magnitude higher detail than the AOP. Given the large amounts of data collected by the sUAS, access to computational resources, which are flexible, scalable, and provide a space for remote collaboration such as Jetstream, will be crucial.

Conservation and Molecular Ecology. Biodiversity loss is a fundamental issue of the Anthropocene (defined as current time period, with human activities dominating ecosystem processes³⁰), with direct potentially irreversible consequences for the wellbeing of humankind. The Endangered Species Act (ESA) of 1973 established powerful legislation for the mitigation of biodiversity loss yet carries with it the prerequisite that candidates for conservation be recognized as species, subspecies, or "distinct population segments" therein. Implicit in this language is the need to understand how biological diversity within species is partitioned across space (eg, within landscapes/riverscapes) to quantify divergence among those groups and, from this pattern, extrapolate recommendations for effective and adaptive management. Although establishing universal criteria for delineating such units remains contentious (eg, see the works of McElroy et al³¹ and Sullivan et al³²), recent molecular advances offer unprecedented resolution.³³ The field of conservation genetics, borne of these molecular developments over the last several decades, is increasingly empowered by access to large-scale data through advances in DNA sequencing technology.^{34,35} Molecular approaches can answer many questions, which would be intractable for "boots-on-the-ground" conservation methods such as: defining conservation units as a function of both historical geomorphic processes³⁶ and contemporary climate change³⁷; how human habitat modification affects integrity of biodiversity "units"³⁸; and evaluating the success of prior management action.³⁹

The emergence of conservation genetics into the era of "biological big data" exposes a critical limitation in that our ability to generate data often out-paces our bioinformatic and computational capabilities. Conservation geneticists (and indeed most biologists using genomic data) face unique constraints such as dataset size (sometimes at the terabyte scale) and limited scalability of analytical methods, restricting applicability of most XSEDE systems. The novelty of genomic data for non-model organisms also leaves the field in many ways naïve in terms of computational expertise, with GUI software executed on personal computers rapidly being replaced, out of necessity, with massively parallel pipelines deployed on high-performance computer clusters.

In navigating these new waters and following much tinkering with less-than-optimal solutions, the Conservation and Molecular Ecology Lab at the University of Arkansas (PI: Marlis Douglas and Michael Douglas) has shifted much of its computational work to Jetstream due primarily to its: 1) Flexibility in workspace environment; 2) Flexibility in workspace size (eg, CPUs and memory), and job walltime (single jobs, for example, may run for weeks to months); and 3) User-friendly management and deployment of resources through Atmosphere. To date, they have used nearly 900,000 Service Units across numerous projects, encompassing 5 in-progress and 1 completed dissertation.³⁸ Due to cumulative limitations (eg, I/O speed, memory, walltime, software environment), there are no other practical and publicly-accessible alternative computational resources available which satisfy all criteria.

Genome annotation. Recent advances in genome sequencing and assembly technologies have significantly accelerated the numbers of organisms whose genomes have been sequenced and assembled. However, this is not the end of a genome project, it is of vital importance to gain insight into functional loci within a genome. Genome annotation is the process of attaching biological information to the assembled sequences. Despite considerable attempts to structurally annotate genes in newly sequenced genomes where preexisting gene models are mostly lacking, the annotation of those genomes still remains a computationally intensive process. One of the most popular tools for genome structural annotation is MAKER, a flexible and scalable genome annotation pipeline that is used for *de novo* annotation of newly sequenced genomes and for updating existing genome annotations. It is widely used as a preliminary step in the analysis of population characterization, gene function and understanding the relationship between species at different levels. MAKER can be deployed as a serial multi-core or MPI application depending on the resources available. Although the scalability of MAKER makes it appropriate for projects of any size, deploying MAKER on traditional HPC clusters is widely seen as a strenuous task mainly because it has a large number of software dependencies that must be installed. MAKER itself is an amalgamation of many external bioinformatics tools, each with their own dependencies and hard-coded location for configuration files, making it difficult to configure it for a multiuser setup typically found in HPC or any shared computational system. In addition, MAKER runs are not time efficient because they are performed sequentially from the input file (one sequence at a time).

To overcome scalability and dependency problems with installing and managing MAKER, Thrasher and collaborators developed a version of MAKER with Workqueue (WQ-MAKER) designed to run on multiple virtual machines in the cloud.⁴⁰ Workqueue is a master-worker framework that provides a means of creating tasks and submitting them to a heterogeneous pool of workers in the cloud.⁴¹ In order to integrate the master-worker, MPI capabilities, and resolve the complex software dependencies for communities interested in genome annotation, Jetstream partners at the University of Arizona collaborated with Douglas Thain's group at the University of Notre Dame to install WQ-MAKER application as a Jetstream image

that uses coarse-grained parallelism over the network and fine-grained parallelism (MPI or multi-core) within a single VM. Because WQ-MAKER does not require any shared file system, this further reduces the file system load and allows for good scalability across commodity networks. A benefit of installing WQ-MAKER on Jetstream compared to traditional High Performance Computers (HPC) is the ability to modify and improve the underlying core platform and network configuration per version, without impacting users of the prior version. Research teams are making effective use of cloud capabilities offered through Jetstream by using the master-worker capabilities and launching multiple workers to scale performance. Likewise, the ability to support multiple versions of WQ-MAKER provides continuity and reproducibility for users relying on a specific version.

5.5 | Science Gateways

The previous decade has seen unprecedented progress in biomedical, computational, and information technology domains. Areas of life sciences research that were previously distant from each other in ideology, analysis practices, and toolkits such as microbial ecology and personalized medicine have all embraced techniques that rely on big data, but the capacity to generate big data greatly outpaced our ability to analyze it. Existing data generation technologies are more mature and accessible than the methodologies that are available for individual researchers to move, store, analyze, and present data in a fashion that is transparent, valid, and reproducible. One tool, which is at the forefront of enabling reproducible research at a large scale, is the science gateway. Science gateways provide Web and desktop-enabled user interfaces and middleware to cyberinfrastructure.⁴² They have dramatically broadened the scope of access to high-performance computing; the XSEDE science gateway program, for example, reports that more unique users per quarter access XSEDE resources through a science gateway to execute scientific applications than traditional command line users. For the June-August 2017 time period, gateway-based users outnumbered traditional users by more than three to one. XSEDE has also supported gateways by providing virtual machines (VMs) that can host gateway services.

5.5.1 | How Jetstream generally enables science gateways

Jetstream extends the reach of science gateways by solving important problems not addressed by previous systems. Jetstream allows gateway operators to dynamically control their VM environment through the OpenStack API and through OpenStack ecosystem projects such as Heat.⁴³ This is essential to modern science gateways, which are adopting design and development practices used by private sector, cloud-based companies. These include the use of microservices, in which the gateway is decoupled into a set of separate, semi-independent processes that typically run inside containers such as Docker. Jetstream VMs are complementary to containers, as the VM itself provides isolation from other users of the system, while containers allow multiple services to run within a single VM by containerizing dependencies. The physical separation of Jetstream's hardware to two geographically distant sites further allows gateways to take advantage of modern cloud approaches to provide redundancy.

Second, Jetstream solves some of the friction in XSEDE and other HPC providers that simultaneously serve traditional and gateway users by providing "virtual clusters" directly to the gateway provider. In mixed traditional and gateway user environments, a common practice is to provide community accounts.⁴⁴ Unfortunately for gateway users, the community user (which may represent thousands of users) may suffer fair-share, queue limits, and other scheduling policies that are designed assuming single users. Jetstream Virtual Clusters⁴⁵ provide computing resources that can be dedicated to the gateway. When equipped with a queuing system such as SLURM, these Virtual Clusters can allow the gateway to set its own usage policies.⁴⁶ As container orchestration engines such as Apache Mesos, Docker Swarm, and Kubernetes^{24,47,48} gain in popularity, gateways can also use these systems in a self-service manner, without relying upon service provider system administrators to deploy resources configured to use these alternative engines for workload management. Finally, Jetstream is co-located at both IU and TACC sites with a complementary system, Wrangler, which provides a 10 petabyte file system space at each site. When coupled with Jetstream (using high speed physical networking and NFS mounts), the two systems combined support data-centric gateways that were not ideal fits for XSEDE in the past. The abstract use case (common in the geosciences in practice) is to use Jetstream VMs to run services and Web interfaces, while serving 10's or 100's of TB of data from Wrangler through network file system mounts.

We see future opportunities for Jetstream to serve science gateways. As a dynamic cloud environment supporting gateway middleware, it would be useful to provide simpler "out of the box" support for features such as load-balancing and failover, and we should consider how to support continuous integration and delivery as a service. The Virtual Cluster capabilities need to become more dynamic (with VMs created and destroyed as needed) to avoid excessive resource usage. Finally, storage access performance is critical for data-based gateways, so we need to identify ways to improve integration between Jetstream and Wrangler. One gateway that's already taking advantage of both the bursting capability and redundancy Jetstream provides as well as standalone interactive environments is Galaxy.

5.5.2 | How Jetstream enables an exemplar science gateway: Galaxy

Galaxy is a software framework for making computational tools available for users without informatics expertise. It provides a user-friendly web-based interface for scientists to use tools individually or connect them into complex workflows, while all aspects of infrastructure management are automatically handled for them. Developers can easily integrate their tools into Galaxy by providing an abstract description of the behavior of a tool. Given this tool metadata, Galaxy can then automatically generate an intuitive user interface, provide automated provenance, integrate with other tools, and more. One key aspect of how Galaxy accelerates research is accessibility – by providing an easy-to-access toolkit, big data analysis

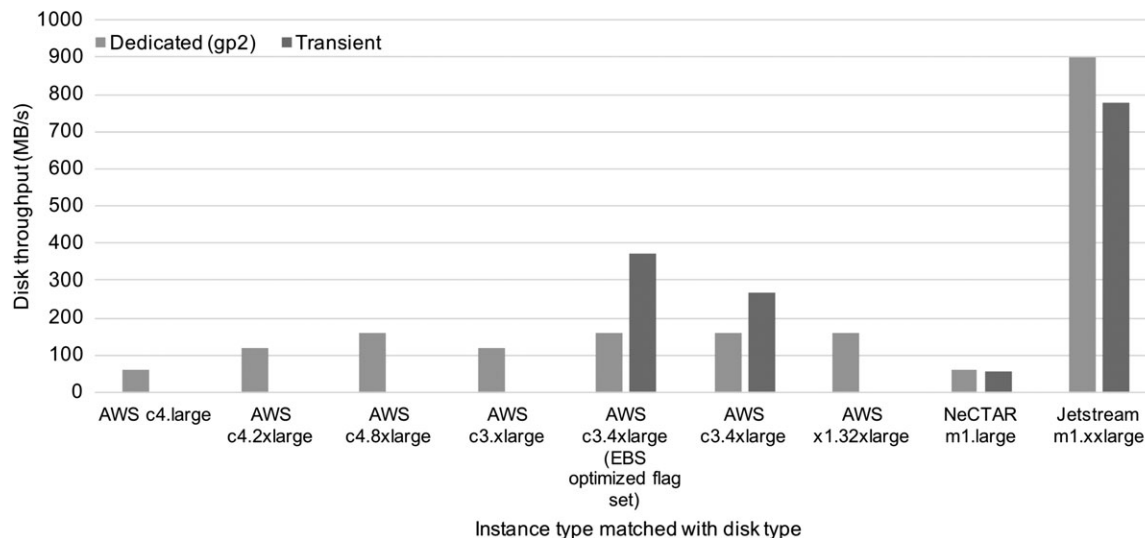


FIGURE 1 Disk throughput performance using the fio command for a variety of disks and instance types from multiple cloud providers and shows the significant performance benefit of Jetstream resources

is democratized. In this regard, and in combination with the Jetstream cloud, the Galaxy project offers two modes of access to Jetstream: via the public Galaxy Main server, accessible to anyone at usegalaxy.org, and via self-serve standalone Galaxy instances.

Galaxy Main on Jetstream. Galaxy Main has been integrated into the Jetstream cloud where analysis jobs are routed to Jetstream while the data transfer and job management are handled by the Galaxy framework. This provides an excellent user experience because analyses run faster using extended compute capacity from Jetstream and users need not be concerned with infrastructure management. Galaxy Main has over 100,000 registered users that run about 250,000 jobs each month. In the past 15 months, Galaxy Main has automatically routed approximately 115,000 jobs to Jetstream from more than 10,000 distinct users. Because of the performance resource characteristics of Jetstream instances (see Figure 1), only computationally demanding multi-CPU jobs are routed to Jetstream. Overall, Jetstream has improved two key aspects of Galaxy Main: (1) it is now possible to run more jobs, more quickly due to the extended resource capacity; and (2) it is possible to run new types of jobs. Jobs can now be executed with complete isolation via virtualization, and through the use of containerization technologies, Galaxy Main now offers interactive environments (IEs). IEs allow users to create extensible environments such as Jupyter or RStudio and programmatically operate on data available in their analysis histories – something previously not possible on the public Galaxy server.

Self-serve Galaxy on Jetstream. For the scenarios where users need to install tools not available on the public server or have resource requirements that exceed quotas, it is also possible to deploy a private production-quality Galaxy server on Jetstream within minutes using just a web browser. The server comes preconfigured with hundreds of tools and access to terabytes of reference genomic data. Once launched, the server is entirely managed by an individual scientist with the ability to install additional tools from the Tool Shed or attach external data storage. In addition to being used by individuals as a research workspace, standalone instances are a great tool for hands-on workshops and tutorials to deliver an environment that is standardized yet customizable by participants. Over the past year, over 650 standalone Galaxy instances were launched on the Jetstream cloud.

Benefits of Jetstream and cloud models. The described solutions contribute toward the accessibility goal of Galaxy. In particular, the ability to have complete control over resources allows for native integration with the traditional compute infrastructure Galaxy applications expect without requiring extensive modifications. This further facilitates a great degree of flexibility in how resources are configured, allowing the execution of interactive environments or matching of resource capabilities to job requirements. Galaxy jobs are often I/O intensive, and experiments on Jetstream have demonstrated the underlying infrastructure outperforms a popular commercial cloud provider by nearly an order of magnitude (see Figure 1). Cumulatively, Galaxy integration with Jetstream has delivered a robust computational platform for biomedical research that can be leveraged by users and service providers outside the Galaxy project.

6 | CONCLUSIONS

Our key conclusion is that the system functions as proposed and has thus far proved valuable to an ever-growing community of users, whether those be individuals performing research or education in scientific and engineering disciplines or entire communities of users brought to NSF CI, and specifically to Jetstream, through a science gateway. A panel of experts appointed by the NSF reviewed the Jetstream implementation in April 2016, resulting in system acceptance upon their recommendation and formal NSF approval. The system then transitioned into full production status, and after another NSF panel review in July 2017, our continued operations have been approved. Jetstream has succeeded in providing high-end resources to a diverse community. It provides valuable software tools for the communities identified as intended users of Jetstream, communities

that are not necessarily large users on traditional XD program resources. It supports multidisciplinary computational science and engineering disciplines, and early experiences suggest it will further the progress of the US open science and engineering community.

The NSF and the Jetstream team have worked together to set new precedents for measuring the effectiveness and operation of government-funded cloud systems. The extensive scope of the acceptance review – spanning basic benchmarks to analysis of user experiences to operational metrics during early operations – helped ensure that the system implemented and presented to the NSF for acceptance is indeed the system that we proposed in the introduction of our initial proposal (corrected for budget changes between proposal and award). The metrics continue to show their relevance at the end of project year 1, and coupled with a more results oriented view of the educational and scientific impacts of Jetstream, we anticipate that they will inform future cloud-based approaches. We endorse this approach as one that will hold the cyberinfrastructure community highly accountable to itself, funding agencies, and our users—and as an approach that in the long run will increase the conformity of systems as delivered to systems as described in grant proposals to funding agencies.

One of the critical challenges going forward will be for the Jetstream team to demonstrate value in the form of return on investment for the NSF. For that reason, we ask that Jetstream users acknowledge Jetstream use by citing it¹ in all publications and products created via some use of Jetstream or Jetstream-related products (including use of VMs accessed from IUScholarWorks); for more information, see our previous work.⁴⁹ This will allow the Jetstream team to easily discover products created with some contribution from Jetstream so that we can document the value this system.

ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation under Award 1445604. We thank all of the staff of the Jetstream partner organizations for making Jetstream such a success so far (especially Therese Miller, C. Bret Hammond, Mike Packard, Edwin Skidmore, Brian Beck, and Ashley Bucholz). The Indiana University Pervasive Technology Institute and our partners have also supported Jetstream implementation. Any opinions expressed here are those of the authors and do not necessarily represent the opinions of any funding agencies.

ORCID

Winona Snapp-Childs  <http://orcid.org/0000-0003-4354-7092>

REFERENCES

1. Stewart CA, Cockerill TM, Foster I, et al. Jetstream - A self-provisioned, scalable science and engineering cloud environment. Paper presented at: XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure; 2015; St. Louis, MO.
2. Heidorn PB. Shedding light on the dark data in the long tail of science. *Libr Trends*. 2008;57(2):280-299. http://muse.jhu.edu/journals/library_trends/v057/57.2.heidorn.html
3. Goecks J, Nekrutenko A, Taylor J, The Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol*. 2010;11(8). <http://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-8-r86>
4. Stewart CA, Hancock DY, Vaughn M, et al. Jetstream - Performance, early experiences, and early results. In: Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale; 2016; Miami, FL.
5. OpenStack Foundation. OpenStack. 2010. <https://www.openstack.org/>
6. SaltStack Inc. SaltStack. 2017. <https://docs.saltstack.com/en/latest/topics/cloud/openstack.html>
7. OpenStack Foundation. Puppet. 2017. <https://wiki.openstack.org/wiki/Puppet>
8. Stewart CA, Hancock DY, Vaughn M, et al. System Acceptance Report for NSF award 1445604 "High Performance Computing System Acquisition: Jetstream - A Self-Provisioned, Scalable Science and Engineering Cloud Environment". Bloomington, IN: Indiana University; 2016. Technical Report.
9. Hancock DY, Packard M, Turner G, Stewart CA. Updated acceptance test results for the Jetstream production environment. Bloomington, IN: Indiana University; 2016. Technical Report. <http://hdl.handle.net/2022/20958>
10. National Science Foundation. NSF Solicitation. 2014. <http://www.nsf.gov/pubs/2014/nsf14536/nsf14536.htm>
11. Fox GC, Fortes J, Grimshaw AS, Keahey K, Smith W. FutureGrid: An experimental, high-performance grid test-bed. Bloomington, IN: Indiana University: National Science Foundation; 2009. <http://nsf.gov/awardsearch/showAward.do?AwardNumber=0910812>
12. Ricci R, Akella A, Wang K, Elliott C, Zink M, Ricart G. Cloudlab. 2014. <http://www.cloudlab.us/>
13. Keahey K, Mambretti J, Panda DK, Rad P, Stanzione D, Riteau P. Chameleon: A configurable experimental environment for large-scale cloud research. 2014. <https://www.chameleoncloud.org/>
14. Merchant N, Lyons E, Goff S, et al. The iPlant collaborative: Cyberinfrastructure for enabling data to discovery for the life sciences. *PLoS Biol*. 2016;14(1):e1002342. <http://dx.plos.org/10.1371/journal.pbio.1002342>
15. Globus. Globus. 2010. <https://www.globus.org/>
16. Indiana University. IUScholarWorks Repository. 2018. <https://scholarworks.iu.edu>, <http://scholarworks.iu.edu/dspace>
17. Mell P, Grance T. The NIST definition of cloud computing recommendations of the National Institute of Standards and Technology. *NIST Special Publ*. 2011;145:7.
18. Barroso L, Hölzle U. *The Datacenter as a Computer*. San Rafael, CA: Morgan & Claypool Publishers. 2013.
19. Stewart CA, Hancock DY, Fischer J. Jetstream overview - what it is, how to apply for use. Bloomington, IN: Indiana University; 2017. <https://cwv.cac.cornell.edu/JetstreamReq/>

20. Stewart CA, Fischer J, Hancock DY. Getting Started With Jetstream. 2015. <https://jetstream-cloud.org/support/index.php>
21. Basney J, Fleury T, Gaynor J. CILogon: A federated X.509 certification authority for cyberInfrastructure Logon. <http://www.ncsa.illinois.edu/People/jbasney/cilogon-xsede13.pdf>
22. ORCID Inc. ORCID. 2012. <https://orcid.org>
23. Project Jupyter team. Zero to JupyterHub With Kubernetes. 2017. <https://zero-to-jupyterhub-with-kubernetes.readthedocs.io/en/latest/>
24. The Kubernetes Authors. Kubernetes. 2017.
25. Jeremy F, Hancock DY, Lowe JM, Turner G, Snapp-Childs W, Stewart CA. Jetstream: A cloud system enabling learning in higher education communities. In: Proceedings of SIGUCCS '17; 2017; Seattle, WA.
26. Swetnam TL, Brooks PD, Barnard HR, Harpold AA, Gallo EL. Topographically driven differences in energy and water constrain climatic control on forest carbon sequestration. *Ecosphere*. 2017;8(4):e01797. <http://doi.wiley.com/10.1002/ecs2.1797>
27. Rstudio Team. RStudio: Integrated Development for R. 2015. <http://www.rstudio.com/>
28. iPlant Collaborative. iPlant Home. 2011. <http://iplantcollaborative.org/>
29. Sankey TT, McVay J, Swetnam TL, McClaran MP, Heilman P, Nichols M. UAV hyperspectral and lidar data and their fusion for arid and semi-arid land vegetation monitoring. *Remote Sens Ecol Conserv*. 2017;4(1):20-33. <http://doi.wiley.com/10.1002/rse2.44>
30. Lewis S, Maslin MA. Defining the Anthropocene. *Nature*. 2015;519:171-180. <https://dx.doi.org/10.1038/nature14258>
31. McElroy DM, Shoemaker JA, Douglas ME. Discriminating *Gila robusta* and *Gila cypha*: Risk assessment and the Endangered Species Act. *Ecol Appl*. 1997;7(3):958-967. [papers2://publication/uuid/9D213FD2-5D16-45CE-B41B-E41AEC5997B0](https://pubs2://publication/uuid/9D213FD2-5D16-45CE-B41B-E41AEC5997B0)
32. Sullivan BK, Douglas MR, Walker JM, et al. Conservation and management of polytypic species: The little striped whiptail complex (*Aspidoscelis inornata*) as a case study. *Copeia*. 2014;2014(3):519-529. <http://www.bioone.org/doi/abs/10.1643/CG-13-140>
33. Funk WC, McKay JK, Hohenlohe PA, Allendorf FW. Harnessing genomics for delineating conservation units. *Trends Ecol Evol*. 2012;27(9):489-496. <http://linkinghub.elsevier.com/retrieve/pii/S0169534712001279>
34. Allendorf FW, Hohenlohe PA, Luikart G. Genomics and the future of conservation genetics. *Nat Rev Genet*. 2010;11(10):697-709. <http://www.nature.com/doi/10.1038/nrg2844>
35. Douglas MR, Douglas ME. Molecular approaches to stream fish ecology. *Community Ecology of Stream Fishes: Concepts, Approaches, and Techniques*; 2010:157-195.
36. Hopken MW, Douglas MR, Douglas ME. Stream hierarchy defines riverscape genetics of a North American desert fish. *Mol Ecol*. 2013;22:956-971.
37. Douglas MR, Davis MA, Amarello M, et al. Anthropogenic impacts drive niche and conservation metrics of a cryptic rattlesnake on the Colorado Plateau of western North America. *Royal Soc Open Sci*. 2016;3(4):160047. <http://rsos.royalsocietypublishing.org/lookup/doi/10.1098/rsos.160047>
38. Bangs MR. Fishes as a Template for Reticulate Evolution: A Case Study Involving *Catostomus* in the Colorado River Basin of Western North America [PhD thesis]. Fayetteville, AR: University of Arkansas; 2016.
39. Mussmann SM, Douglas MR, Anthonysamy WJB, et al. Genetic rescue, the greater prairie chicken and the problem of conservation reliance in the Anthropocene. *Royal Soc Open Sci*. 2017;4(2). <https://dx.doi.org/10.1098/rsos.160736>
40. Thrasher A, Musgrave Z, Kachmarck B, Thain D, Emrich S. Scaling up genome annotation using MAKER and work queue. *Int J Bioinform Res Appl*. 2014;10(4/5):447. <http://www.inderscience.com/link.php?id=62994>
41. Bui P, Rajan D, Abdul-Wahid B, Izaguirre J, Thain D. Work queue+ python: A framework for scalable scientific ensemble applications. Paper presented at: Workshop on Python for High Performance and Scientific Computing at SC11; 2011; Seattle, WA.
42. Lawrence KA, Zentner M, Wilkins-Diehr N, et al. Science gateways today and tomorrow: positive perspectives of nearly 5000 members of the research community. *Concurr Comput Pract Exp*. 2015;27:4252-4268.
43. OpenStack Foundation. Heat. 2017. <https://wiki.openstack.org/wiki/Heat>
44. Welch V, Barlow J, Basney J, Marcusiu D, Wilkins-Diehr N. A AAAA model to support science gateways with community accounts. *Concurr Comput Pract Exp*. 2007;19(6):893-904. <http://doi.wiley.com/10.1002/cpe.1081>
45. Knepper R, Coulter E, Pierce M, Marru S, Pamidighantam S. Using the jetstream research cloud to provide science gateway resources. Paper presented at: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID); 2017; Madrid, Spain
46. SchedMD LLC. SchedMD | Slurm Support and Development. 2017. <https://www.schedmd.com/>
47. Apache Software Foundation. Apache Mesos. 2017. <http://mesos.apache.org/>
48. Docker. Swarm mode overview. 2017. <https://docs.docker.com/engine/swarm/>
49. Stewart CA, Hancock DY, Fischer J. Citing Jetstream. <https://jetstream-cloud.org/research/citing-jetstream.php>

How to cite this article: Hancock DY, Stewart CA, Vaughn M, et al. Jetstream—Early operations performance, adoption, and impacts. *Concurrency Computat Pract Exper*. 2018:e4683. <https://doi.org/10.1002/cpe.4683>