Managing Computational Gateway Resources with XDMoD

Jeanette M. Sperhac[1], Robert L. DeLeon, Thomas R. Furlani, Steven M. Gallo, Martins Innus, Matthew D. Jones, Jeffrey T. Palmer, Abani Patra, Benjamin D. Plessinger, Ryan Rathsam, Nikolay Simakov, Joseph P. White, Rudra Chakraborty, Gregary Dean

Center for Computational Research, University at Buffalo, State University of New York, Buffalo, NY, 14203, USA

Abstract

The U.S. National Science Foundation (NSF) has invested heavily in research computing, funding the XSEDE network of supercomputers and enabling their integration with science gateways. To ensure maximal return on this substantial investment, it is essential to monitor the use of these computing resources. XD Metrics on Demand (XDMoD) is an NSF-funded tool that was developed to help manage high performance computational resources. XDMoD metrics describe accounting and performance data for computational jobs, including resources consumed, wait times, and quality of service. XDMoD can provide information on individual jobs, or data aggregated over an ensemble of jobs. Its web interface offers centralized charting, exploration, and reporting of these metrics, for user-selected time ranges, and across resources.

XDMoD is directly relevant to the gateways community. In this paper, we introduce XDMoD, demonstrate its utility for gateways, and outline our plans to further enhance its capabilities for the gateways community. Furthermore, we demonstrate how XDMoD can help gateway users and gateway and resource managers answer many questions about utilization, performance, and availability. In doing so, we showcase the evolution of gateways in the XSEDE ecosystem.

Email address: jsperhac@buffalo.edu (Jeanette M. Sperhac)
[1]Corresponding author

Preprint submitted to Future Generation Computer Systems *March 18, 2019*

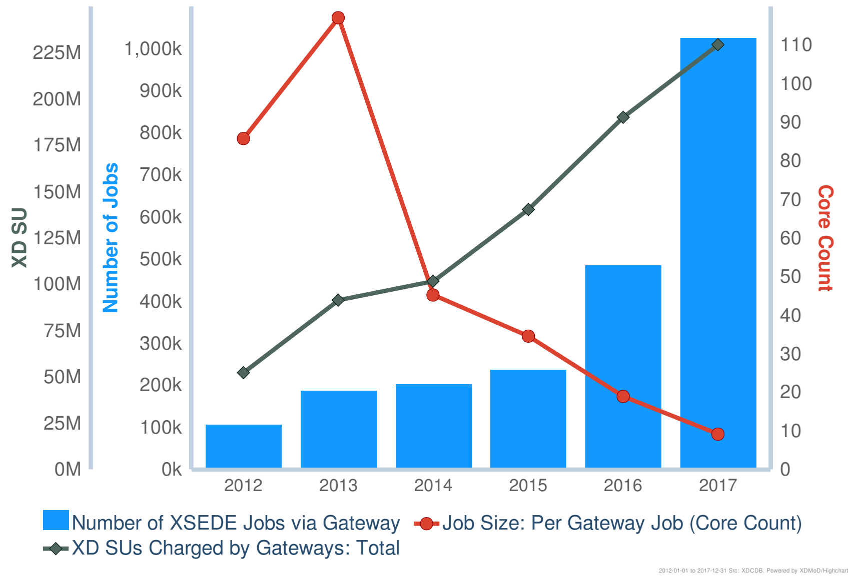


Figure 1: The XDMoD web interface enables users to plot and compare metrics. Shown is the tenfold growth in gateway job submissions to XSEDE resources, 2012-2017 (job count; blue bars), with a concurrent decrease in average job size (average core count; red circles). The two opposing trends result in linear growth of gateway resource consumption (XD SUs; gray squares).

Keywords: Science Gateways, High Performance Computing, monitoring

1. Introduction

The XD Metrics on Demand (XDMoD) tool provides web-browser access to data about utilization, performance, and quality of service for High Performance Computing (HPC) resources [1, 2, 3, 4]. This comprehensive tool was originally developed to support the National Science Foundation (NSF) XSEDE program; it was later open-sourced and made available to universities, government laboratories, and commercial entities [5]. XDMoD enables users, managers, and operations staff to monitor, assess, and maintain quality of service for a set of computational resources. To do this, XDMoD harvests data from the underlying resources, then displays the resulting job, usage, and accounting metrics, using the XDMoD web interface, which provides a rich array of visual analysis and charting tools.

Over the last five years, science gateways have come online in XSEDE, simplifying access to computing resources for a wide variety of new users. See Figure 1 for a clear illustration of the tenfold growth in gateway job submissions to XSEDE resources over this time. Gateway users and gateway and resource managers alike may have questions about the resulting utilization and how to manage it. The XDMoD team is designing new capabilities to better answer their questions. With these changes, we aim to improve our support for the gateways community, both inside and outside XSEDE.

The remainder of this paper is organized as follows: Section 1 introduces the capabilities of XDMoD; Section 2 demonstrates XDMoD's current support for gateways and their users; and Section 3 outlines future XDMoD enhancements planned for gateways. Finally, Section 4 provides a summary.

1.1. Objectives of XDMoD

XDMoD offers metrics and analysis tools to assist the end users, staff, developers, and managers of computing resources, gateways included. Its objectives are:

- Enable *users* to assess and optimize their use of resources;
- Assist resource *operations staff* as they tune resource and application performance;
- Provide *user support personnel* with job performance metrics;
- Provide *resource managers* with metrics for long-range analysis and planning.

1.2. Capabilities of XDMoD

XDMoD was designed to be modular. At base, XDMoD is composed of four main components: tools for collecting and processing raw data, a data warehouse, a role-based REST API[2], and a web-based front end. Optional *modules* can be installed to extend the base functionality of XDMoD. These modules provide quality of service monitoring and a job-level Performance realm. They are briefly described in Sections 1.2.1 and 1.2.2.

[2]An Application Program Interface (API) that conforms to the REpresentational State Transfer (REST) model.

Table 1: Example XDMoD Realms and Metrics. Realms collect metrics, or statistics about computing jobs, into related groups. See Appendix A for a complete list.

Realm	**Metric**
Accounts	Number of Accounts Created
Allocations	Allocation Usage Rate
	Number of Allocations
Jobs	CPU Hours
	Number of Jobs (via Gateway)
	Wall or Wait Hours
	Number of Users
Performance	Total Memory
	CPU User
	I/O Bandwidth
	Block Read or Block Write Rate

XDMoD mines log files from computational resources to provide a multitude of job metrics that describe resource usage and utilization. All XDMoD metrics are assembled into groups called *realms*, based on the type of information they provide. For example, the Jobs realm contains metrics gleaned largely from job accounting data, while Accounts and Allocations realms contain metrics describing user accounts and user allocations. Refer to Table 1 for examples of metrics found in these realms. Metrics can be filtered, grouped, and drilled down by *dimension*, where dimensions describe, for instance, the characteristics of a job, or of the user that ran the job. Table 2 provides examples of dimensions available in the Jobs and the optional Performance realms; lists of all XSEDE XDMoD metrics and dimensions are found in the Appendix.

The web-based XDMoD user interface offers plotting and data exploration functionality divided into *tabs*. The tool's front end enables users to plot metrics in either timeseries or aggregate view, and interact with metrics through filtering and drilldown. It provides reporting capabilities that include charting, data export, and custom report generation.

Table 2: Example XDMoD Dimensions. Dimensions describe a job, its performance, or the user that ran the job, and enable grouping, filtering, and drilldown on job statistics. See Appendix B for a complete list.

Realm	**Dimension**
Jobs	Gateway
	Resource
	Job Size
	Job Wall Time
Performance	Application
	CPI (Cycles per Instruction) Value
	Data Source
	Exit Status

1.2.1. Application Kernels module

The Application Kernels module enables quality-of-service monitoring for HPC resources [6, 7]. It periodically runs computationally lightweight benchmarks, establishing baselines for system performance. These benchmarks help center personnel pinpoint underlying problems when a resource performs poorly. Application Kernels are running on these XSEDE resources: TACC Stampede2 [8], SDSC Comet [9], PSC Bridges [10], and LSU SuperMIC [11]. They may optionally be used with Open XDMoD, and as such we run them at our own center. A detailed discussion of this module is outside of the scope of this paper.

1.2.2. Performance module

The Performance module[3] consists of components that collect data from system hardware counters. The resulting job-level data is presented for viewing and analysis in the XDMoD user interface [12]. Monitoring software runs on compute nodes, and collects information such as memory usage, interconnect fabric traffic, shared/exclusive status, and CPU and/or GPU performance [13, 14, 15]. These data are supplemented with job submission script information, then ingested into the XDMoD database. The web interface presents these metrics in the Performance realm. An analysis of gateway job performance data using this module is demonstrated in Section 2.2.

[3]The XDMoD Performance module is also referred to as the SUPReMM module.

Job performance data are collected on these XSEDE resources: TACC Stampede2 [8], SDSC Comet [9], PSC Bridges [10], and LSU SuperMIC [11]. The performance module may optionally be installed with Open XDMoD, and as such we run it at our own center.

1.3. Open XDMoD

XDMoD was originally developed for the NSF XSEDE program. Open XDMoD, the widely-adopted open-source version of the tool, also enjoys regular releases and active support by the development team. We continually enhance it with new features, and we accept user-contributed enhancements that adhere to our code standards.

The source for Open XDMoD is available on github.com[4]; we also support installation by RPM package. Open XDMoD works with a variety of resource managers and architectures, and can be installed on computational resources of any size [17, 18]. Additionally, we offer tools to help configure Open XDMoD installations. Thus, settings describing resource information, user hierarchies, and user access controls can easily be customized to reflect any host institution and its computing resources. All optional modules developed to extend XDMoD are also supported in Open XDMoD; Sections 1.2.1 and 1.2.2 provide brief descriptions of these modules. Open XDMoD also provides support for gateways run by non-XSEDE computer centers.

The open-source version of XDMoD helps us to manage the Center for Computational Research (CCR) at the State University of New York at Buffalo. Other installations of Open XDMoD number in the hundreds, and include industrial and academic centers and government labs in Australia, Brazil, Canada, Ecuador, France, Germany, India, Norway, Poland, Saudi Arabia, Spain, Turkey, the United Kingdom, the U.S., and elsewhere.[5]

Aside from XSEDE-specific customizations, and configuration and setup features to support general use, the functionality of the XSEDE and Open XDMoD versions is similar. In this paper, both versions will be referred to as "XDMoD" unless the distinction is merited.

---

[4] The version described here is Open XDMoD 7.5.0 [16], `https://github.com/ubccr/xdmod/tree/xdmod7.5`.

[5] Institution and location are determined at download and upgrade time.

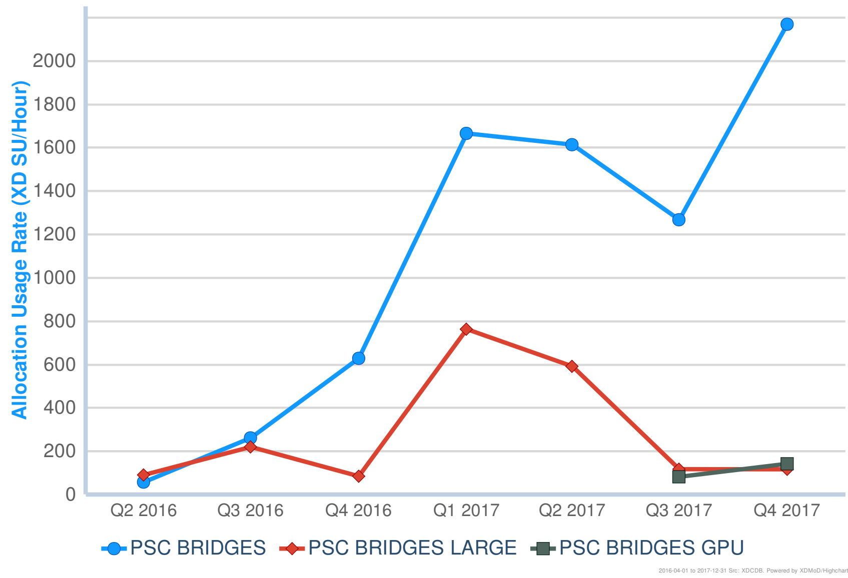


Figure 2: XDMoD plot of the allocation usage rate by gateways running jobs on PSC resources, Q2 2016 (the inception of the Bridges resource) to Q4 2017. Shown are Bridges (blue circles), Bridges Large (red diamonds), and Bridges GPU (gray squares) resources. Plots such as this could help a center manager quantify the use of their center's resources by gateways.

2. XDMoD and Gateways

Science Gateways are defined on the XSEDE Science Gateway page as "*portals to computational and data services and resources across a wide range of science domains.*" [19] Gateways bring resource and gateway managers together to provide end users with streamlined access to computation, including workflows, visualization tools, job submission tools, HPC, and so forth. Different members of the gateways community may have questions about performance and utilization of their gateways and the underlying resources. We undertake to answer these typical questions using the XDMoD tool.

For example, *center managers* may ask about the distribution of gateway activities across the resources they manage (Figure 2); while *resource managers* may ask how many jobs gateways are running on their resource (Figure 3), what proportion of all gateway jobs ran on their resource (Figure 4), or how many gateways are using their resource (Figure 5). In addition, *gateway managers* may ask about active allocations over the lifetime of their gateway, or the allocations' activity on different resources (Figure 6). *End users*,

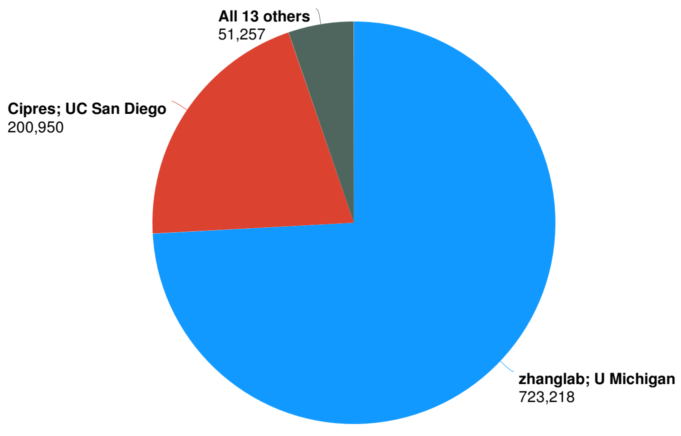


Figure 3: XDMoD plot of the number of gateway jobs started on SDSC's Comet supercomputer, 2017. The gateways shown are Zhanglab (or I-TASSER, blue), with 723,218 jobs started [20], and CIPRES (red), with 200,950 jobs started [21]. All other gateways using Comet in 2017 started 51,257 jobs (gray).

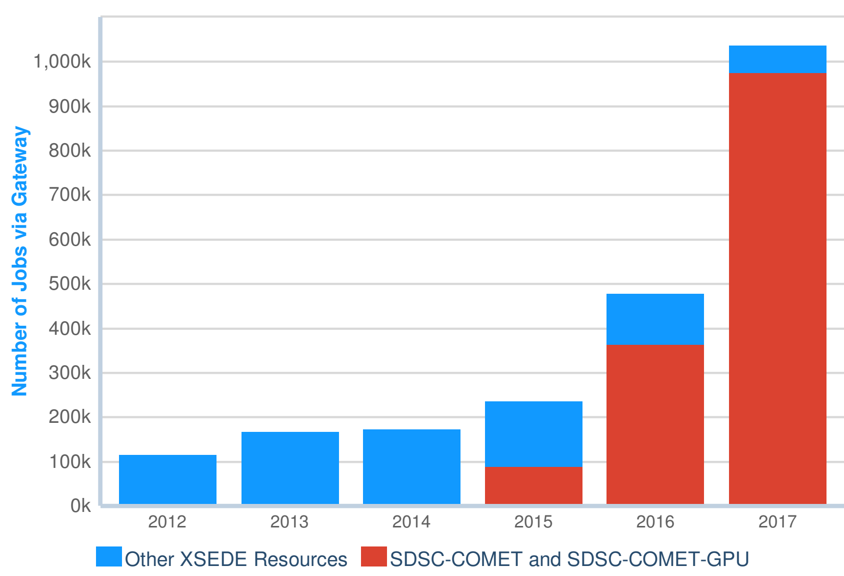


Figure 4: XDMoD plot of the growth in XSEDE gateway job count, 2012-2017. Jobs are divided into two groups: those run on Comet (red) and those run on all other XSEDE resources (blue). Comet is clearly the dominant gateways resource for XSEDE.

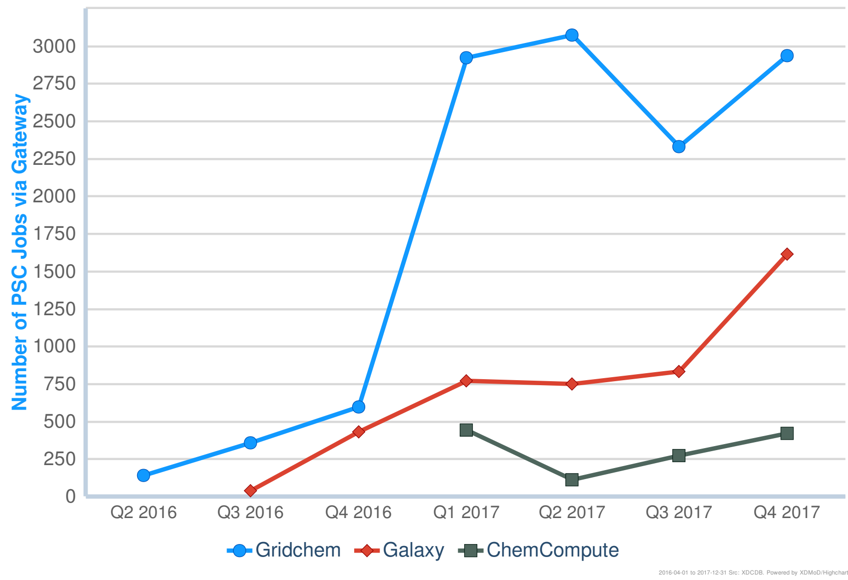


Figure 5: XDMoD plot of the top three gateways by the number of jobs submitted to PSC resources, Q2 2016 (the inception of the Bridges resource) to Q4 2017. Shown are Gridchem (blue circles) [22], Galaxy (red diamonds) [23], and ChemCompute (gray squares) [24]. This plot complements Figure 2's usage rate illustration by quantifying the top gateways responsible for using PSC resources over this time period.

finally, may ask how quickly they are consuming their allocations (Figure 7), and how efficiently their jobs ran (Figure 12). XDMoD can provide answers to these types of questions, as shown in the figures, using the plotting, drilldown, and analysis capabilities outlined in Section 1.2.

Gateway-specific information can be accessed in XDMoD by filtering any metric of interest on the *User* dimension, and specifying the gateway's Community User name, or by filtering on the *Gateways* dimension, and specifying the gateway name.[6] Also useful for gateways is the Jobs metric, *Number of Jobs by Gateway*, which is pre-filtered to include all Community Users. Charts in the XDMoD Custom Queries tab provide additional gateway information that is not yet available in the Jobs realm; Section 2.3 provides an introduction to this functionality.

[6]Each gateway runs its jobs on computing resources under a Community User account, with one such account per gateway. Therefore, the User dimension has a unique value corresponding to each gateway, making it effective for capturing gateway activity.

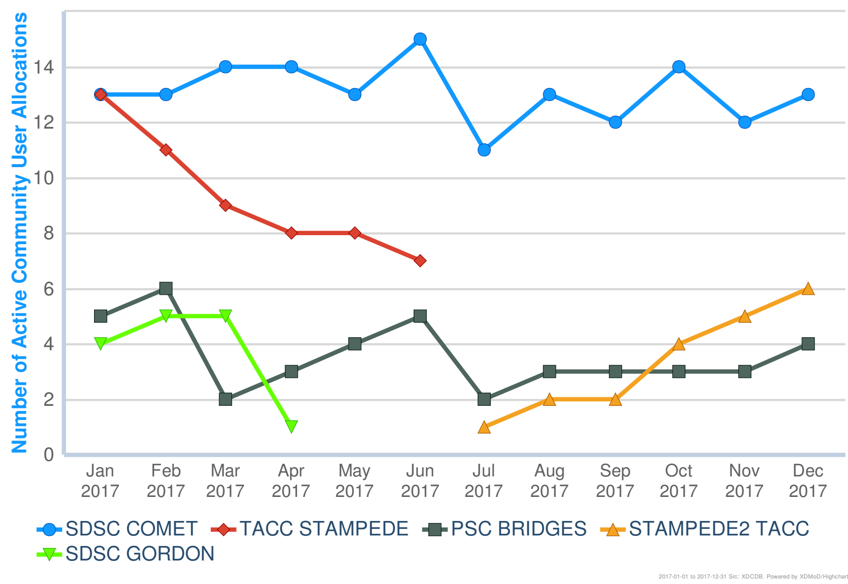


Figure 6: XDMoD plot of XSEDE gateway allocation counts, by resource, 2017. Shown are Comet (blue circles), Stampede (red diamonds), Bridges (gray squares), Stampede2 (yellow triangles–point up), Gordon (green triangles–point down). This plot shows Comet's consistent importance to XSEDE gateways, while illustrating Stampede2 allocations beginning to replace those on Stampede, and Gordon reaching retirement.

In rare cases, the *Allocation* dimension gives a more complete picture of gateway usage; see Figure 7, which charts allocations on different resources over the lifetime of the Computational Anatomy gateway [25], and Section 2.4.2, which discusses caveats in usage reporting for gateways that can lead to apparent gaps in the data. For further information on creating gateways plots in XDMoD, Appendix D describes how the XDMoD plots featured in this paper were prepared, including metrics, dimensions, grouping, filtering, and drilldown.

2.1. Jobs realm metrics and dimensions

The metrics available in XDMoD's Jobs realm measure attributes of the computational jobs. These metrics entail such figures as counts of jobs run and measures of the resources consumed by those jobs. A complete list of Jobs metrics for XSEDE XDMoD is provided in Table A.4 of the Appendix. To demonstrate some of the possibilities for gateways analysis, gateway-specific plots of Jobs realm metrics are illustrated in Figures 1 through 10.

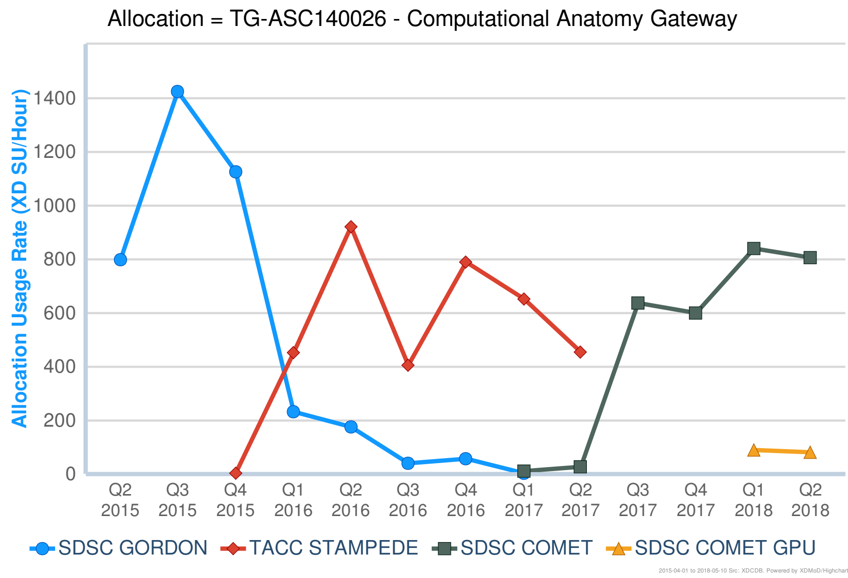


Figure 7: XDMoD plot of the Computational Anatomy Gateway's quarterly allocation usage rate on four XSEDE resources from Q2 2015 to Q2 2018. [25] Shown are Gordon (blue circles), Stampede (red diamonds), Comet (gray squares), Comet GPU (yellow triangles). The plot enumerates the resources used by this gateway over several years of its operational lifetime, as new resources came online and others retired. Note that the gateway's allocation usage rate does not regain its peak of Q3 2015.

In the Jobs realm, XDMoD users can investigate different utilization measures, such as Node Hours, CPU Hours, and XSEDE Service Units (XD SUs), selecting the metric that best reflects the resources under consideration. For example, in Figures 8 and 9, XD SUs are plotted, rather than CPU Hours, to allow a more meaningful comparison of usage across resources with different configurations.[7]

Furthermore, numerous Jobs realm metrics are measured in multiple ways. For example, Number of Jobs is determined for running, ended, started, and submitted jobs; utilization figures such as Node Hours, CPU Hours, and XD Service Units (XD SUs) are also reported both as totals and per job. To illustrate, Figure 8 displays total XD SUs, or the *total use* by

[7]XD SUs are computational resources consumed, in units standardized across XSEDE; each resource defines a conversion factor derived from High-Performance LINPACK (HPL) benchmarks. An XD SU is defined as one CPU-hour on a Phase-1 DTF cluster; a Phase-1 DTF SU is equal to 21.576 Normalized Units (NUs). [26]

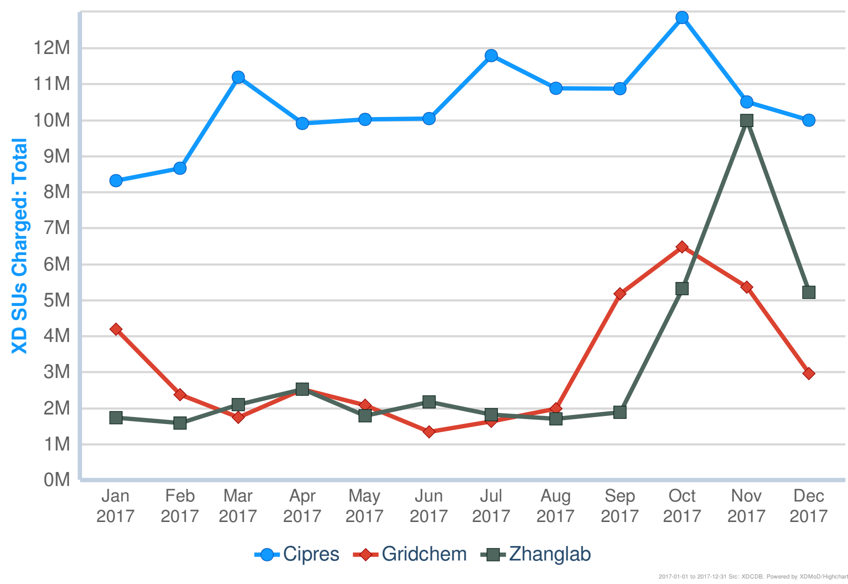


Figure 8: XDMoD plot of the top three XSEDE gateways by *total* XD SUs charged, 2017: CIPRES (blue circles) [21], Gridchem (red diamonds) [22], Zhanglab (gray squares) [20]. CIPRES is the clear leader in resources consumed among XSEDE gateways.

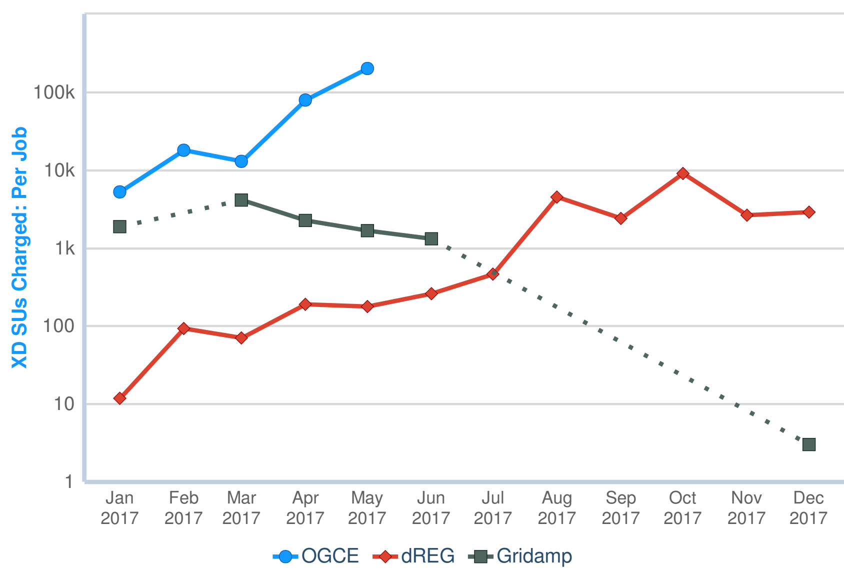


Figure 9: XDMoD plot of the top three XSEDE gateways by average XD SUs charged *per job*, 2017: OGCE (blue circles) [27], dREG (red diamonds) [28], Gridamp (gray squares) [29]. Note that the top gateways for average resource consumption per job, shown here, differ from the top gateways for total resource consumption, shown in Figure 8.

gateways during 2017, whereas Figure 9 shows average XD SUs *per gateway job* in the same year. Offering these related metrics enables XDMoD users to contrast, in this example, the top gateway in total XD SUs consumed (CIPRES, whose users consumed approximately ten million XD SUs per month in 2017) with the top gateway in average XD SUs consumed per job (OGCE, whose users consumed on average 50,000 XD SUs per job in each month from January to May 2017).

Other Jobs realm metrics describe XSEDE allocations and accounting as they pertain to computing jobs. These metrics enable resource managers to answer questions such as: how many gateways allocations are active on their resources; and the rate at which those allocations are being charged. For example, Figure 2 displays the gateways allocation usage rate on Pittsburgh Supercomputing Center's (PSC) Bridges resource, since its inception; usage rates for Bridges, Bridges–Large, and Bridges–GPU are shown. Meanwhile, Figure 6 illustrates counts of gateway (Community User) allocations on XSEDE resources Comet, Stampede, Bridges, Stampede2, and Gordon, by month of 2017. The introduction and retirement of various resources can be seen in the plotted allocations.

Jobs metrics can be filtered or grouped by different dimensions for plotting and further analysis. These dimensions include Allocation, Gateway, Job Size, Resource, User, and others, as shown in Table B.7 of the Appendix. For example, Figure 10 groups two Jobs metrics by the NSF Directorate dimension to show 2017 gateways job counts and the resulting resource consumption by those directorates. The Grant Type dimension, though it contains a *Gateway* value, incorporates types of usage that are not typically associated with gateways. Therefore, for analyzing gateway metrics, we employ the User or Gateway dimension instead of Grant Type.

2.2. Performance realm metrics and dimensions

The Performance realm's metrics provide information such as shared/exclusive status, requested and allocated resources, parallel file system usage, and memory usage. Lists of all performance metrics and dimensions are provided in Tables A.5 and B.8 of the Appendix. These metrics are made available so long as the gateway's underlying resources collect and report job performance data. (Note that the optional performance module must be installed for these metrics to be collected under an Open XDMoD installation.) The XDMoD web interface enables users to aggregate performance metrics for an ensemble of jobs, or to probe these metrics for a single job.

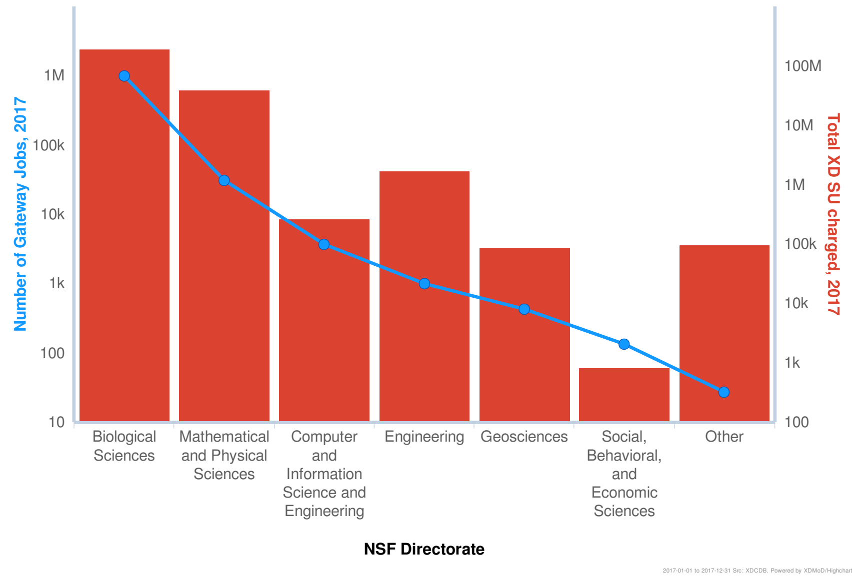


Figure 10: XSEDE XDMoD plot of gateway job counts (blue circles), and total XD SUs charged by gateways (red bars) in 2017, grouped by the top seven U.S. NSF directorates served (by gateway job count). Note that NSF Directorate "Other" is one of the 158 Directorates originating in the XSEDE Central Database. The NSF directorates vary widely in both job counts and usage.

Demonstrating a Performance realm plot in aggregate, Figure 11 compares three CPU% metrics for CIPRES gateway jobs run on XSEDE resources, in 2016. This plot displays the average percent of processor time spent idle, devoted to job code, and spent running system calls, all weighted by core hours, for exclusive (non-shared) CIPRES jobs. The results are provided as a function of job size (core count). To interpret this plot, we consider that approximately 60% of 2016 CIPRES jobs ran on the Comet resource, which has 24-core nodes; the remaining 40% ran on Gordon, which had 16-core nodes.[8] Noting that, we see that the jobs with size 1, 2-4, and 5-8 cores (those with average CPU% User near 50%) are all running on single nodes. Meanwhile, the 9-64 core bin includes one- to two- node Comet jobs and

[8]This resource distribution was determined using XDMoD. In 2016, 124,403 CIPRES jobs ran on SDSC Comet (of these, 153 were shared jobs, which are not included in this analysis), and the remaining 81,653 on SDSC Gordon (node sharing was not supported on Gordon).

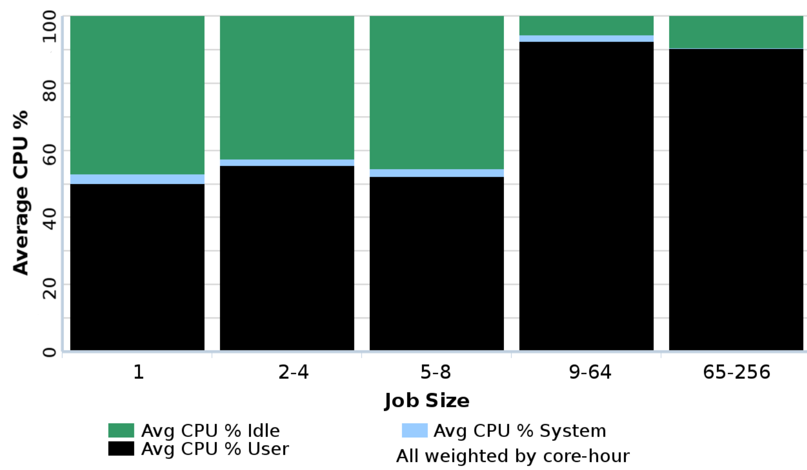


Figure 11: XDMoD plot of CPU% for all non-shared CIPRES gateway jobs run on XSEDE resources in 2016. This displays the percent of processor time spent idle (green), spent running job code (black), and spent running system code on behalf of a job (blue), as a function of Job Size (Core Count). Job Size bins have 1, 2-4, 5-8, 9-64, and 65-256 cores. Jobs that used more of the available cores per node dedicated more CPU cycles to job processes.

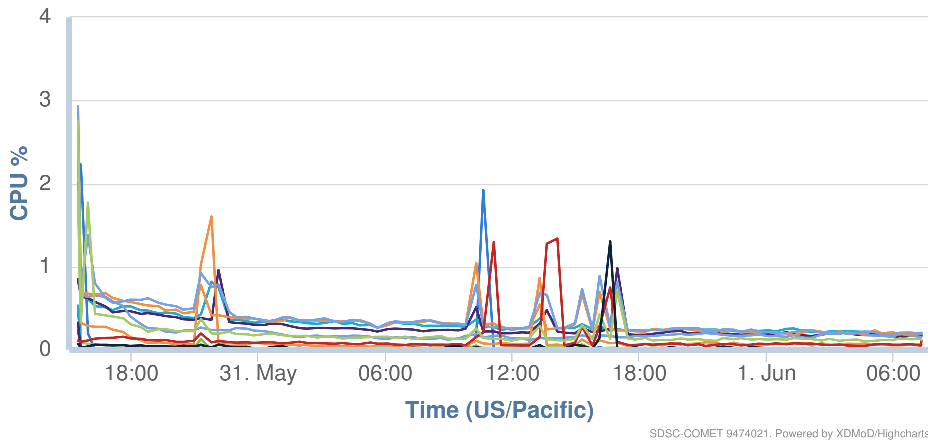
one- to four-node Gordon jobs, and the 65-256 core bin includes jobs using 3 to 10 nodes on Comet, and 5 to 16 nodes on Gordon. The latter two bins show average CPU User surpassing 90%. We conclude that CIPRES gateway jobs that used more of the available cores per node also used the CPU most efficiently, dedicating more CPU cycles to user (in this case, job) processes. Customizable bin sizes (not presently available in XDMoD) would further clarify this figure.

2.2.1. Job Viewer

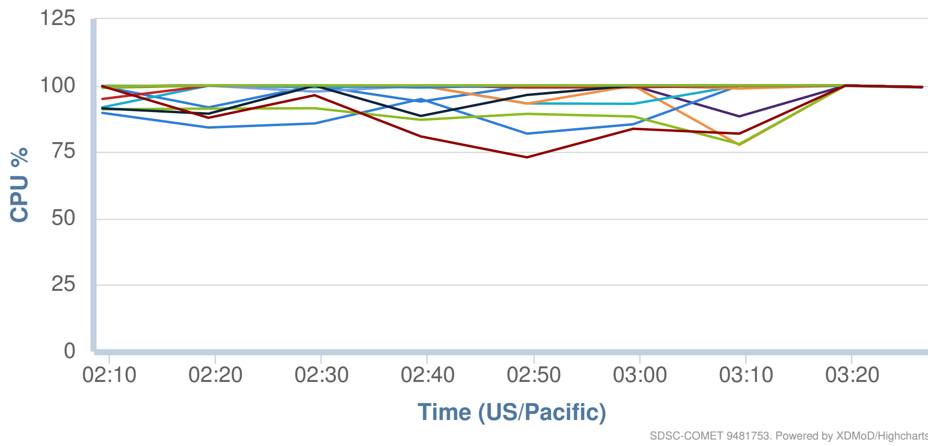
The XDMoD user interface's Job Viewer presents details about a job's executable, requested and allocated resources, wall and wait time, shared/exclusive status, and timeseries plots of job metrics such as CPU user, flops, parallel file system usage, and memory usage.[9]

For example, in Figure 12, we contrast two jobs running the same ex-

[9]In XSEDE XDMoD, individual job performance details are accessible only to the user that submitted the job, and to resource support personnel and principal investigators, who are considered to be managers of the job. In Open XDMoD, the optional performance module must first be installed.



(a) Inefficient resource utilization: sub 3% CPU User for 24 cores.



(b) Efficient resource utilization: near 100% CPU User for 24 cores.

Figure 12: XDMoD Job Viewer timeseries plots compare CPU User % (the average percentage of time spent in CPU user mode) for two CIPRES jobs on Comet. Both jobs ran the same executable. Note the y-axis scale difference. Each line represents one of 24 cores.

ecutable, each in exclusive mode. The figure clearly displays a substantial difference in resource utilization between the two related jobs.[10] Given reasonable assumptions about the executable, namely, that it is floating point intensive, and uses low memory bandwidth, we conclude that Figure 12a shows inefficient use of the requested resources, with CPU User less than 3%

[10]Note that we lack information about the input data.

for 24 cores; Figure 12b, meanwhile, shows CPU User near 100% for 24 cores.

In summary, the Job Viewer lets support staff and end users examine job performance data in detail, enabling diagnosis of inefficient jobs, and fostering better use of computing resources.

2.3. Custom Queries: Gateways Plots

XSEDE XDMoD release 7.1 included a new set of aggregate gateway plots, available to all users[11] under the Custom Queries tab. This series of pie-chart plots shows counts of jobs or users across gateways and resources, using data attributes not yet fully exploited by XDMoD. Several plots of-

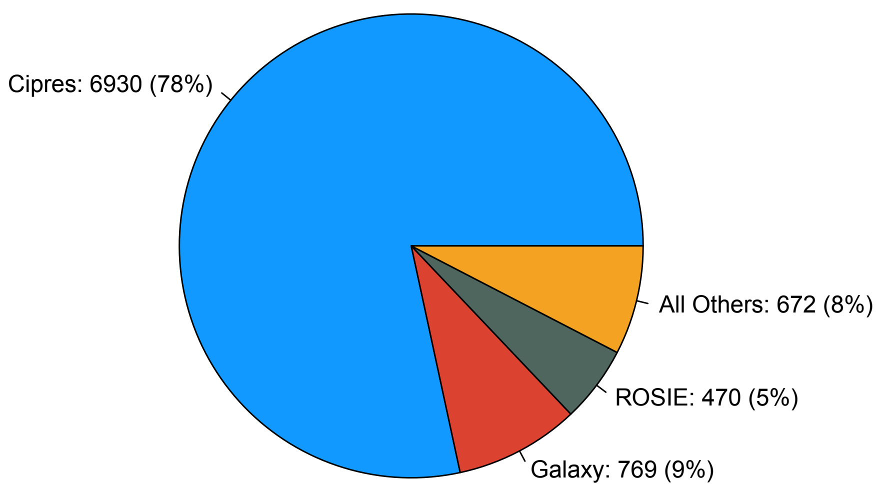


Figure 13: Plot of the number of unique gateway users who ran jobs on XSEDE resources during 2016, grouped by gateway. Gateways shown: CIPRES (blue, 78.4% of unique XSEDE gateway users) [21], Galaxy (red, 8.7%) [23], ROSIE (gray, 5.3%) [30].

[11] All registered XSEDE users may request access; public users are excluded.

Figure 13 shows the number of unique gateway users who ran jobs on XSEDE resources during 2016, grouped by the top three such gateways, CIPRES, Galaxy, and ROSIE. We see from this figure that CIPRES has more than three times as many unique, active XSEDE users than all other XSEDE gateways combined. Such insights into gateway data on the XSEDE systems provide a forerunner of the full-blown gateways realm we are planning for XDMoD; refer to Section 3 for further information.

2.4. Gateways data caveats

XDMoD was designed to provide resource managers, end users, and others with access to job, performance, and utilization data. The generalization of data access to the gateways community, however, is not complete. In this section we discuss some caveats of XDMoD data access for this community, before proposing solutions in Section 3.

2.4.1. Data access limitations

Although XDMoD presents useful aggregated data relevant for gateways users and managers, several data access shortcomings are evident; as a result, some analyses that are possible for resource managers are simply unavailable to gateway managers. For example, gateway managers or users might want to use the XDMoD Job Viewer to search for, investigate, and troubleshoot gateways jobs. There are two reasons this is not yet possible. First, access to non-aggregated job performance data is currently unavailable to individual gateway users. Second, *gateway* user IDs cannot yet be directly searched or filtered in the XDMoD web interface. This analysis is therefore not available to the gateways community at this time.

Some other types of analyses are not yet possible for the gateways community. For example, a gateway manager might want to evaluate how rapidly their gateway users are consuming computing resources. However, as mentioned, individual gateway usernames are not yet tracked in XDMoD; only the Community User under which the jobs run is known to the resource. Furthermore, gateway managers typically lack full access to the resources where their gateways submit jobs, and as a result, they cannot normally access those individual job performance data. Therefore, this usage analysis cannot yet be performed by the gateways community in XDMoD. A proposed enhancement to XDMoD directly addresses these data access challenges that gateways face; this work is discussed in Section 3.

2.4.2. Data reporting limitations

In XSEDE XDMoD, the XSEDE Central Database is a critical source of data which receives job data directly from the resources. Gateway usage is typically identified with the *community user* account under which gateway jobs run on the associated resources. The community user data returned from the XSEDE Central Database thus provide a summary of gateway usage on the resource, as in the *Number of Jobs via Gateway* metric on XDMoD.

However, variation may arise between counts of jobs submitted by the gateway community user account and the related allocation. This can occur during gateway or resource development, when a resource user is mistakenly tagged as a community user, when gateways routinely access a resource under a non-community user account, or under other circumstances (For example, extremely short job wall times, or jobs run by guest users). These variations at the dimension level can add up to mismatched counts when gateways metrics are filtered or grouped. Therefore, we urge gateways and resources to work with us to ensure accurate reporting to the XSEDE Central Database and reliable gateways metrics on XDMoD.

3. Future work: Gateways realm

Although XDMoD offers access to aggregated data for gateways users and managers, it does not satisfy all of the gateways community's data access needs. (Current limitations of gateways data access are discussed in Section 2.4.1). To address these gaps, the XDMoD team plans to develop a dedicated gateways data realm that will treat gateways as their own entities, much as service providers and computing resources are treated now in XDMoD. In this proposed realm, individual gateway users will be directly associated with the jobs they run on a resource, regardless of whether the users have accounts on the resource. This substantial enhancement will enable drilldown, filtering, and searching by individual gateway user. In addition, this new realm will collect new gateways job metrics and attributes to simplify and extend XDMoD use by the gateways community.

These enhancements will also permit gateway managers to access individual job data pertinent to their own gateways. Gateway managers will now be able to evaluate, for example, how efficiently individual gateway jobs use computing resources. Overall, this proposed enhancement to XDMoD will treat gateways as first-class citizens and facilitate truly granular access to gateway user behavior and job management.

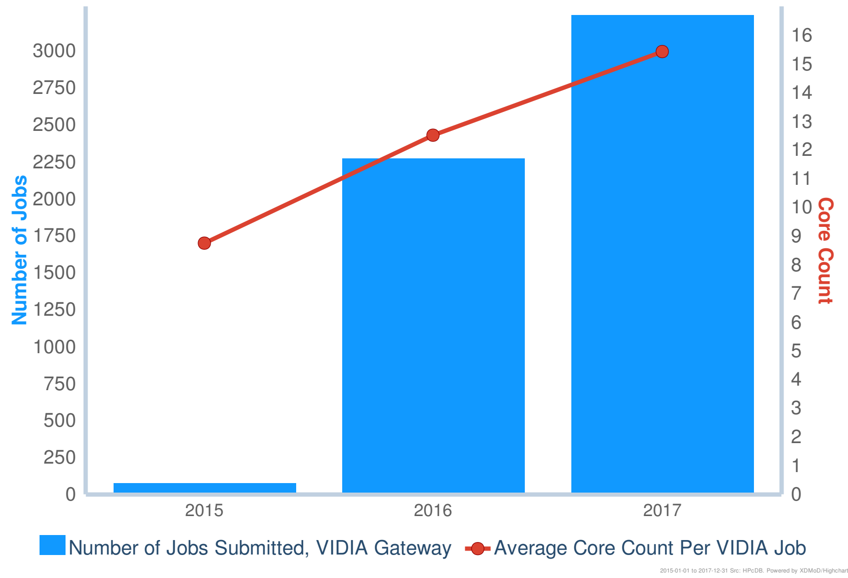


Figure 14: Open XDMoD plot of the count of jobs submitted from the VIDIA gateway (blue bars), and average core count per job (red circles), 2015-2017 [31]. All jobs ran on CCR's academic cluster.

Offering new gateways metrics starts with collecting the data. Gateways managers have requested collection of job attributes that include: Gateway name/user; Resource/Software used; Job size/Walltime; Field of science; and Principal Investigator (PI). Gateways and providers must log these job attributes so they can be incorporated into XDMoD's database, and so that the corresponding dimensions can be offered for analysis, searching, filtering, and drilldown.

The new gateways realm will be prototyped in-house at CCR with usage data from VIDIA [31], a gateway that supports data-intensive computation for university campuses that lack access to HPC resources. VIDIA enables access to CCR's HPC cluster, where its users ran more than 3200 jobs in 2017. See Figure 14 for an XDMoD plot of the growth in VIDIA-submitted cluster jobs from 2015 to 2017. The Open XDMoD instance running at CCR will thus serve as a testbed for collecting, ingesting, and aggregating data on gateway resource access and user jobs. We are confident that better integration of gateways data in XDMoD will help us manage our own gateway, and will likewise assist the XSEDE and open-source gateways communities.

4. Conclusion

XDMoD, a comprehensive tool for managing HPC resources, collects and presents numerous metrics that describe computational jobs, resources consumed (computation, memory, disk, network, etc.), wait times, and quality of service. The XDMoD web interface offers tools for charting and visualizing these metrics, enabling interactive drilldown to expose and plot related information. The information provided by XDMoD allows for a comprehensive view of gateway usage in XSEDE, as we have demonstrated in this paper. Gateways users and administrators, whether they use XSEDE or Open XDMoD, can benefit from the data and metrics offered in XDMoD. We intend to further enhance the tool for the gateways community.

Acknowledgments

This work is supported by the National Science Foundation under grants OCI 1025159 (Technology Audit Service), OCI 1203560 (SUPReMM), and ACI 1445806 (XD Net Metrics Service for High Performance Computing). The authors would like to thank the reviewers for their detailed comments.

References

[1] Center for Computational Research, University at Buffalo, XD Metrics on Demand (XDMoD), `https://xdmod.ccr.buffalo.edu` (2018).

[2] J. C. Browne, R. L. DeLeon, A. K. Patra, W. L. Barth, J. Hammond, M. D. Jones, T. R. Furlani, B. I. Schneider, S. M. Gallo, A. Ghadersohi, R. J. Gentner, J. T. Palmer, N. Simakov, M. Innus, A. E. Bruno, J. P. White, C. D. Cornelius, T. Yearke, K. Marcus, G. von Laszewski, F. Wang, Comprehensive, open-source resource usage measurement and analysis for HPC systems, Concurrency and Computation: Practice and Experience 26 (13) (2014) 2191–2209, cPE-14-0027.R1. doi:10.1002/cpe.3245.
URL `https://dx.doi.org/10.1002/cpe.3245`

[3] T. R. Furlani, B. I. Schneider, M. D. Jones, J. Towns, D. L. Hart, S. M. Gallo, R. L. DeLeon, C. Lu, A. Ghadersohi, R. J. Gentner, A. K. Patra, G. Laszewski, F. Wang, J. T. Palmer, N. Simakov, Using XDMoD

to facilitate XSEDE operations, planning and analysis, in: Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery (XSEDE '13), ACM, 2013, p. 8. doi:10.1145/2484762.2484763.

[4] J. C. Browne, R. L. DeLeon, C. Lu, M. D. Jones, S. M. Gallo, A. Ghadersohi, A. K. Patra, W. L. Barth, J. Hammond, T. R. Furlani, R. T. McLay, Enabling comprehensive data-driven system management for large computational facilities, in: Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis (SC '13), ACM, 2013, p. 11. doi:10.1145/2503210.2503230.

[5] J. T. Palmer, S. M. Gallo, T. R. Furlani, M. D. Jones, R. L. DeLeon, J. P. White, N. Simakov, A. K. Patra, J. M. Sperhac, T. Yearke, R. Rathsam, M. Innus, C. D. Cornelius, J. C. Browne, W. L. Barth, R. T. Evans, Open XDMoD: A tool for the comprehensive management of high-performance computing resources, Computing in Science and Engineering 17 (4) (2015) 52–62. doi:10.1109/MCSE.2015.68.
URL https://dx.doi.org/10.1109/MCSE.2015.68

[6] N. A. Simakov, J. P. White, R. L. DeLeon, A. Ghadersohi, T. R. Furlani, M. D. Jones, S. M. Gallo, A. K. Patra, Application kernels: HPC resources performance monitoring and variance analysis, Concurrency and Computation: Practice and Experience 27 (17) (2015) 5238–5260, cPE-14-0402.R1. doi:10.1002/cpe.3564.
URL https://dx.doi.org/10.1002/cpe.3564

[7] T. R. Furlani, M. D. Jones, S. M. Gallo, A. E. Bruno, C.-D. Lu, A. Ghadersohi, R. J. Gentner, A. K. Patra, R. L. DeLeon, G. von Laszewski, L. Wang, A. Zimmerman, Performance Metrics and Auditing Framework Using Applications Kernels for High Performance Computer Systems, Concurrency and Computation: Practice and Experience 25 (7) (2013) 918–931. doi:10.1002/cpe.2871.
URL https://dx.doi.org/10.1002/cpe.2871

[8] TACC Stampede 2 Supercomputer, https://www.tacc.utexas.edu/systems/stampede2, Texas Advanced Computing Center, Austin.

[9] SDSC Comet Computing Cluster, http://www.sdsc.edu/support/

user_guides/comet.html, San Diego Supercomputer Center at UC San Diego.

[10] PSC Bridges Supercomputer, https://www.psc.edu/bridges, Pittsburgh Supercomputing Center.

[11] LSU SuperMIC Supercomputer, http://www.hpc.lsu.edu/resources/hpc/system.php?system=SuperMIC, Louisiana State University High Performance Computing.

[12] Center for Computational Research, University at Buffalo, XDMoD Performance (SUPReMM) Module, http://supremm.xdmod.org/ (2018).

[13] J. P. White, R. L. DeLeon, T. R. Furlani, S. M. Gallo, M. D. Jones, A. Ghadersohi, C. D. Cornelius, A. K. Patra, J. C. Browne, W. L. Barth, J. Hammond, An analysis of node sharing on HPC clusters using XDMoD/TACC_Stats, in: Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment, XSEDE '14, ACM, New York, NY, USA, 2014, pp. 31:1–31:8. doi:10.1145/2616498.2616533.
URL https://dx.doi.org/10.1145/2616498.2616533

[14] T. Evans, W. L. Barth, J. C. Browne, R. L. DeLeon, T. R. Furlani, S. M. Gallo, M. D. Jones, A. K. Patra, Comprehensive resource use monitoring for HPC systems with TACC_Stats, in: Proceedings of the First International Workshop on HPC User Support Tools, HUST '14, IEEE Press, Piscataway, NJ, USA, 2014, pp. 13–21. doi:10.1109/HUST.2014.7.
URL https://dx.doi.org/10.1109/HUST.2014.7

[15] R. L. DeLeon, T. R. Furlani, S. M. Gallo, J. P. White, M. D. Jones, A. K. Patra, M. Innus, T. Yearke, J. T. Palmer, J. M. Sperhac, R. Rathsam, N. A. Simakov, G. von Laszewski, F. Wang, TAS view of XSEDE users and usage, in: Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure, XSEDE '15, ACM, New York, NY, USA, 2015, pp. 21:1—-21:8. doi:10.1145/2792745.2792766.
URL https://dx.doi.org/10.1145/2792745.2792766

[16] Center for Computational Research, University at Buffalo, Open XDMoD, Version 7.5.0, DOI: 10.5281/zenodo.1469590 (2018).

[17] J. P. White, M. Innus, M. D. Jones, R. L. DeLeon, N. Simakov, J. T. Palmer, S. M. Gallo, T. R. Furlani, M. Showerman, R. Brunner, A. Kot, G. Bauer, B. Bode, J. Enos, W. Kramer, Challenges of Workload Analysis on Large HPC Systems: A Case Study on NCSA Blue Waters, in: Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact, PEARC17, ACM, New York, NY, USA, 2017, pp. 6:1–6:8. doi:10.1145/3093338.3093348.
URL http://doi.acm.org/10.1145/3093338.3093348

[18] N. A. Simakov, J. P. White, R. L. DeLeon, S. M. Gallo, M. D. Jones, J. T. Palmer, B. D. Plessinger, T. R. Furlani, A Workload Analysis of NSF's Innovative HPC Resources Using XDMoD, CoRR abs/1801.04306. arXiv:1801.04306.
URL http://arxiv.org/abs/1801.04306

[19] XSEDE Science Gateway page, https://www.xsede.org/ecosystem/science-gateways.

[20] Zhanglab Gateway, or I-TASSER (Iterative Threading ASSEmbly Refinement), https://zhanglab.ccmb.med.umich.edu/I-TASSER/, Protein Structure and Function Prediction Gateway, University of Michigan.

[21] CIPRES Gateway, https://www.phylo.org/, cyberinfrastructure for Phylogenetic Research, University of California, San Diego.

[22] SEAGrid Gateway (also known as Gridchem Gateway), https://seagrid.org/, Scientific gateway for Computational Chemistry, Molecular Dynamics, Structural Mechanics, and Fluid Dynamics, Indiana University.

[23] Galaxy Gateway, https://galaxyproject.org/, Platform for data-intensive biomedical research.

[24] ChemCompute Gateway, https://chemcompute.org.

[25] Computational Anatomy Gateway, https://mricloud.org/, a gateway for the quantitative analysis of variability of biological shape.

[26] XSEDE (Extreme Science and Engineering Discovery Environment), XSEDE SU Converter, https://www.xsede.org/su-converter (2017).

[27] OGCE/SciGaP Gateway, https://scigap.org/.

[28] dREG Gateway, https://dreg.dnasequence.org/.

[29] GridAMP, the Asteroseismic Modeling Portal, https://amp.phys.au.dk/, a gateway for Stellar Astronomy and Astrophysics.

[30] ROSIE Gateway, http://rosie.rosettacommons.org/, Gateway for the Rosetta software suite enables 3-D structure prediction for biological macromolecules.

[31] J. M. Sperhac, S. M. Gallo, VIDIA: A HUBzero Gateway for Data Analytics Education, Future Generation Computer Systems-doi:https://doi.org/10.1016/j.future.2018.02.004.
URL http://www.sciencedirect.com/science/article/pii/S0167739X17308531

Appendix A. Realms and Metrics

Realms in XDMoD collect metrics, or statistics about computing jobs, into related groups. Tables A.3, A.4, and A.5 list all metrics supported in the Accounts, Allocations, Requests, Jobs, and Performance realms for XSEDE XDMoD version 7.5.

Table A.3: XSEDE XDMoD Metrics for Accounts, Allocations, and Requests Realms, as of Release 7.5.

Realm	Metric
Accounts	Number of Unique Accounts Created
	Number of Unique Accounts Created, with Jobs
Allocations	Allocation Usage Rate
	CPU Core Hours Allocated
	Normalized Units (NUs) Allocated
	Number of Allocations
	XSEDE Standardized Units (XD SUs) Allocated [26]
	XD SUs Used
Requests	Number of Projects
	Number of Proposals

Table A.4: XSEDE XDMoD Metrics for Jobs Realm, as of Release 7.5.

Realm	**Metric**
Jobs	Allocation Usage Rate
	CPU Hours (per Job; or Total)
	Job Size (Min; or Max)
	Normalized Job Size (% of total Cores)
	Job Size per Job
	Job Size, weighted by XSEDE Standardized Units (XD SUs) [26]
	Normalized Units (NUs) Charged (per Job; or Total)
	Node Hours (per Job; or Total)
	Number of Active Allocations
	Number of Active Institutions
	Number of Jobs (Ended; Running; Started; Submitted; or via Gateway)
	Number of Active Principal Investigators (PIs); or Resources; or Users
	User Expansion Factor
	Wait Hours (per Job; or Total)
	Wall Hours (per Job; or Total)
	XD SUs Charged (per Job; or Total)
	XSEDE Utilization (%)

Table A.5: XSEDE XDMoD Metrics for Performance Realm, as of Release 7.5.

Realm	**Metric**
Performance	Average CPU % (Idle; System; or User)[12]
	Average (/home; /scratch; or /work) write rate[13]
	Average Cycles Per Instruction (CPI)[14]
	Average Ratio of Clock Ticks to L1D Cache Loads (CPLD)[14]
	Average CPU User Coefficient of Variation (CV)[12]
	Average CPU User Imbalance[12]
	Average Floating Point Operations per Second (FLOPS)[14]
	Average InfiniBand Rate[13]
	Average Memory Bandwidth[14]
	Average Memory[14]
	Total Memory[14]
	Block Read Rate[13]
	Block Write Rate[13]
	Average eth0 transmit rate[13]
	Average eth0 receive rate[13]
	Average ib0 transmit rate[13]
	Average ib0 receive rate[13]
	Average Lustre transmit rate[13]
	Average Lustre receive rate[13]
	Average mic0 transmit rate[13]
	Average mic0 receive rate[13]
	Average mic1 transmit rate[13]
	Average mic1 receive rate[13]
	Total CPU Hours (Idle; System; or User)
	Total CPU Hours
	Number of Jobs (Ended; Running; Started; or Submitted)
	Wait Hours (per Job; or Total)
	Wall Hours (per Job; Total; Requested per Job; or Requested Total)

[12]Weighted by core-hour
[13]Per node, weighted by node-hour
[14]Per core, weighted by core-hour

Appendix B. Dimensions

Dimensions in XDMoD describe a job, its performance, or the user that ran the job, and enable grouping, filtering, and drilldown on job statistics. Tables B.6, B.7, and B.8 list all dimensions associated with the Accounts, Allocations, Requests, Jobs, and Performance realms for XSEDE XDMoD version 7.5. Open XDMoD will have slightly different dimensions, depending on how the installation is configured.

Table B.6: XSEDE XDMoD Accounts, Allocations, and Requests Dimensions, as of Release 7.5.

Realm	Dimension
Accounts	Resource
	Resource Type
Allocations	Allocation
	Allocation Type
	Board Type
	Field of Science
	National Science Foundation (NSF) Directorate
	Parent Science
	Principal Investigator (PI)
	Resource
	Resource Type
Requests	Field of Science
	NSF Directorate
	Parent Science

Table B.7: XSEDE XDMoD Jobs Dimensions, as of Release 7.5.

Realm	**Dimension**
Jobs	Allocation
	Field of Science
	Gateway
	Grant Type
	Job Size
	Job Wall Time
	Node Count
	National Science Foundation (NSF) Directorate
	Parent Science
	Principal Investigator (PI)
	PI Institution
	Queue
	Resource
	Resource Type
	Service Provider
	System Username
	User
	User Institution
	User NSF Status

Table B.8: XSEDE XDMoD Performance Dimensions, as of Release 7.5.

Realm	**Dimension**
Performance	Application
	Application Class: Method
	Catastrophe Rank
	Cycles per Instruction (CPI) Value
	CPU User Coefficient of Variation (CV)
	CPU User Value
	Datasource
	Exit Status
	Field of Science
	Grant Type
	Granted Processing Element
	InfiniBand Receive Rate
	Job Size
	Job Wall Time
	Lustre Bytes Received
	Node Count
	National Science Foundation (NSF) Directorate
	Parent Science
	Peak Memory Usage
	Principal Investigator (PI)
	PI Institution
	Queue
	Resource
	Service Provider
	Share Mode
	System Username
	User
	User Institution

Appendix C. Gateways Custom Queries

The following table lists the Gateways Custom Queries developed for XSEDE XDMoD.

Table C.9: XSEDE XDMoD Gateways Metrics available as Custom Queries, as of Release 7.5.

Gateways Metric
Unique Gateway Users by Resource and Gateway
Unique Gateway Users by Resource
Unique Gateway Users by Gateway
Unique NEW Gateway Users by Resource and Gateway
Unique NEW Gateway Users by Resource
Unique NEW Gateway Users by Gateway
Number of Gateway Jobs by Resource
Number of Gateway XD SUs by Resource [26]
Number of Gateway Jobs by Gateway
Number of Gateway XD SUs by Gateway
Number of Gateway Jobs from Unknown User
Number of Gateway XD SUs from Unknown User
Gateway XDSU vs Total XSEDE XD SUs

Appendix D. Preparation of figures

In this section we describe how to prepare the figures shown in this paper, using the XDMoD tool. Figures 1 through 13 use XSEDE XDMoD, and appropriate permissions are required to access Job Viewer and Custom Queries data.

Figure 1 plots three metrics together: the Jobs via Gateway, Job Size per Job, and XD SUs (total) metrics. Filter on Community User to exclude non-Gateway jobs. The TAS Community User is also filtered out (Note that the TAS Community User is used to collect data for XDMoD and is not, strictly speaking, a gateway).

Figure 2 uses the Allocation Usage Rate metric, filtered by the User and the Resource dimensions, to show only gateways (Community User) usage on PSC resources Bridges, Bridges Large, and Bridges GPU. The TAS Community User is also filtered out.

Figure 3 uses the Number of Gateway Jobs Started metric, selects the 2017 time duration in aggregate, and filters on Resource equal to Comet. The plot then drills down by Gateway, and displays the top 2 data series, with value labels selected, as a pie chart.

Figure 4 shows a five-year bar chart of Number of Jobs via Gateway (filter by Resource, on Comet and Comet GPU). Select time aggregation: by Year. Add a second data series, again with the Number of Jobs via Gateway metric, filtered by Resource, excluding Comet and Comet GPU. Stack the bars and edit the legend items for brevity.

Figure 5 uses the Jobs via Gateway metric, groups by User, filters on all Community Users, filters on all PSC resources, shows top 3 Users, and suppresses the remainder (displayed as Other). Legends have been edited for brevity. The TAS Community User is also filtered out.

Figure 6 filters the User dimension by Community User so that only gateway allocations are selected. The plot uses the Number of Allocations: Active metric, filtered by Community User and grouped by Resource. The TAS Community User is also filtered out.

Figure 7 uses the Allocation Usage Rate metric, filtered by the Computational Anatomy Gateway allocation and grouped by the Resource dimension. The *lddmmproc* Community User under which the majority of these jobs are run does not account for all of this Gateway's jobs.

Figure 8 uses the total XD SUs metric, groups by User, filters on all Community Users, shows top 3, and suppresses the remainder (displayed as

Other). Legends have been edited for brevity. The TAS Community User is also filtered out.

Figure 9 uses the XD SUs per job metric, groups by User, filters on all Community Users, shows top 3, and suppresses the remainder (displayed as Other). Legends have been edited for brevity. The TAS Community User is also filtered out.

Figure 10 is a 2017 aggregate plot count of gateway jobs grouped by NSF directorate. Add the metric Total XD SU charged, grouped by NSF directorate, plotted as bars. Filter Total XD SUs to show gateways only. Log scale both y axes; edit legend for brevity. Finally, display only the top 7 directorates.

Figure 11 is an aggregate plot of the three Performance metrics, CPU % Idle, User, and System, for the year 2017. The three metrics are filtered to display only exclusive (jobs that did not share nodes) CIPRES gateway jobs, and are grouped by Job Size (Core Count).

Figure 12 displays CPU % User Timeseries metrics, available in the Job Viewer.

Figure 13 uses the Unique XSEDE Users by Gateway metric, in the Custom Queries tab, selecting the top four entries (one of which aggregates the user counts for all gateways beyond the top three). The resulting data were exported from XDMoD as a CSV and the plot shown was prepared in R. Custom Queries plots are not as customizable as the remainder of XDMoD, but all data are exported easily.

Figure 14 is a timeseries plot of two Jobs metrics, namely Number of Jobs and Core Count, for the years 2015 to 2017. The metrics are filtered on user to display only VIDIA community user data, and are aggregated by year. The data series legends are edited for brevity. These data are available only on the Open XDMoD instance housed at CCR (`metrics.ccr.buffalo.edu`).