2018 IEEE International Conference on Cluster Computing

Federating XDMoD to Monitor Affiliated Computing Resources

Workshop paper: HPCMASPA 2018

Jeanette M. Sperhac*, Benjamin D. Plessinger, Jeffrey T. Palmer, Rudra Chakraborty, Gregary Dean, Martins Innus, Ryan Rathsam, Nikolay Simakov, Joseph P. White, Thomas R. Furlani, Steven M. Gallo, Robert L. DeLeon, Matthew D. Jones, Cynthia Cornelius, Abani Patra
Center for Computational Research, University at Buffalo, State University of New York, Buffalo, NY, 14203, USA
*Corresponding author email: jsperhac@buffalo.edu

***Abstract*—The XD Metrics on Demand (XDMoD) tool was designed to collect and present detailed utilization data from the XSEDE network of supercomputers. XDMoD displays a wealth of information on computational resources. Its metrics report accounting and performance figures for computational jobs, including resources consumed, wait times, and quality of service. XDMoD can aggregate this information over an ensemble of jobs, or report it for individual jobs. Its web-based interface supports charting, exploration, and reporting for any time range, across all computing resources. Over the last eight years, we have open-sourced XDMoD, generalizing and packaging it to produce Open XDMoD. Open XDMoD may be installed, configured, and run on any computing cluster, and we have extended and refined it to include support for storage, cloud resources, science gateways, and more.**

In this paper, we describe the federation of XDMoD instances. This new functionality makes it possible to associate and monitor networks of related resources, using the full power of XDMoD to aggregate and display data from disparate XDMoD installations. Charting and reports from XDMoD federations can encompass anything ranging from a single resource to the entire network. The federation capability is flexible; federated resources can be coupled loosely or tightly, and a variety of different information sharing and authentication paradigms are supported. We also highlight new functionality that incorporates storage and cloud facilities into XDMoD federations. Federation will enable XDMoD users to monitor and manage ever more diverse computing resources.

I. Introduction

The XD Metrics on Demand (XDMoD) tool provides ready access to data about utilization, performance, and quality of service for High Performance Computing (HPC) and cloud and other resources. [1]–[5] This comprehensive tool was originally developed to support the U.S. National Science Foundation (NSF) XSEDE program; [6] it was later open-sourced and made available to universities, government laboratories, and commercial entities. [7] XDMoD enables operations staff, end users, resource managers, and other stakeholders to monitor and optimize system and job level performance. To do this, XDMoD harvests and aggregates job, usage, and accounting data from computing resources, then enables users to chart the resulting metrics over any desired timeframe, using a rich array of visual analysis and charting tools. See Figure 1 for an XDMoD plot showing the top three XSEDE resources in 2017, by total SUs charged.

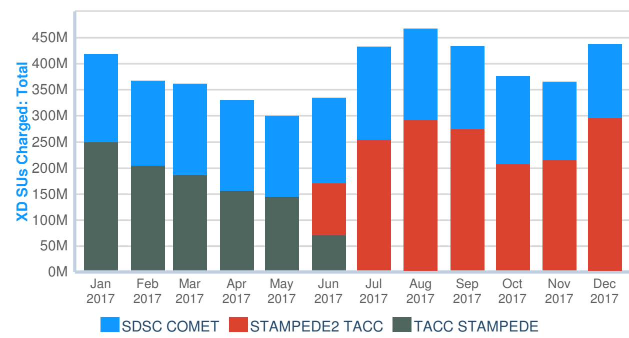

Fig. 1. The XDMoD web interface enables users to chart and explore usage data. Shown are the top three XSEDE resources in 2017, by total SUs charged: Comet [8] (blue); Stampede2 [9] (red); and Stampede [10] (gray).

While XDMoD already supports job, allocation, and performance data and metrics, the XDMoD team is actively enhancing the tool with data federation capabilities and with specialized metrics that describe cloud and storage resources. With these new offerings, we aim to further improve our support for diverse resources and organizations.

The remainder of this paper is organized as follows: Section I briefly introduces XDMoD; Section II details the new XDMoD Federation capability; Section III describes bringing Storage and Cloud facilities into XDMoD federations; and Section IV concludes the paper.

A. XDMoD objectives

XDMoD offers metrics and analysis tools intended to assist the varied stakeholders of high-performance computing resources. It enables users to plot, explore, and interact with metrics, export their data, and automate reports. XDMoD can:

- Enable the *user community* to assess and optimize their use of resources;

2168-9253/18/$31.00 ©2018 IEEE
DOI 10.1109/CLUSTER.2018.00074

- Assist resource *operations staff* as they diagnose and tune resource, job, and application performance;
- Provide *user support personnel* and *end users* with detailed information on job performance;
- Allow *software developers* to perform detailed analysis of job and application performance for code optimization and improvement;
- Provide other *stakeholders* (funding agencies, institutional administration, computing center management) with metrics for long-range analysis and planning.

The objective of our current work on XDMoD federation, as we articulate in this paper, is to further enhance XDMoD for use across distributed high-performance computing environments, larger organizations, and diverse resource types.

B. Related work

A number of monitoring tools can track HPC performance; some open-source examples include Performance Co-Pilot (PCP) [11], Ganglia [12], [13], Lightweight Distributed Metric Service (LDMS) [14], [15], XALT [16], [17], Zabbix [18], and TACC_Stats [19], [20]. Others are discussed elsewhere. [21] Most of these tools track hardware and OS performance counters, to enumerate the performance of the computational nodes. Indeed, several of these tools (PCP, TACC_Stats, Ganglia) form important parts of the data pipeline for XDMoD by providing raw system-level performance data.

Perhaps the most directly comparable open-source tool is HOPSA. [22] HOPSA collects performance information to assist in the tuning of both HPC systems and the applications that run on them. To do so, HOPSA also relies on performance counter data and a number of performance optimization tools. [23]–[26] However, development has halted on this tool.

One of XDMoD's strengths lies in its modular design, which allows future adoption of additional data sources, enables flexible configuration for use on different systems, and provides extensible views suited to a range of users from managers to application developers. As architectures and practices evolve, we anticipate XDMoD keeping pace, as exemplified by the work described in this paper.

C. Open XDMoD

XDMoD itself was developed for the NSF XSEDE program; however, Open XDMoD, the tool's open-source version, also enjoys active support and regular new releases. We continually enhance it with features requested by users, and we accept user-contributed enhancements that adhere to code standards.

Open XDMoD can be installed from an RPM package or from source, which is available on github.com [27]. It can be installed on computational resources of any size; it can be configured to collect and represent data in academic or industrial settings; and it accepts data from a variety of resource managers and architectures. [28], [29] Additionally, we have developed tools to assist academic or industrial centers in XDMoD's configuration, so that departmental hierarchy, resource information, user types and access, and other settings reflect the host institution and its computing resources. Open XDMoD also supports the optional modules developed to extend XDMoD. (Refer to Section I-E).

As a result, installations of Open XDMoD number in the hundreds, running at industrial and academic computing 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.[1] The tool also helps us manage our own computing center, the Center for Computational Research (CCR) at the State University of New York at Buffalo.

Although XDMoD contains some XSEDE-specific customizations, and Open XDMoD features adaptations for general usage, the user interface and functionality of the XSEDE and Open XDMoD versions is similar. Therefore, in this paper, when the distinction is unnecessary, the two will simply be referred to as "XDMoD".

D. XDMoD capabilities

XDMoD mines log files from resource managers such as SLURM and output from monitoring systems such as Performance Co-Pilot (PCP) to provide a wide array of job metrics. Metrics collected describe usage and utilization of the computing resources, including counts of jobs, CPU hours, wall times, wait times, and job sizes. These metrics may be examined for a single job or aggregated over an ensemble of jobs within any desired time frame, and most metrics can be plotted in either timeseries or aggregate view. XDMoD supports data-analytic functions such as filtering, grouping and drill-down. It also provides reporting capabilities that include data export and custom report generation. See Figure 1 for a plot showing the total SUs charged in 2017 on the leading XSEDE resources (ranked by total SUs charged).

The metrics collected by XDMoD are assembled into groups called *realms*, based on the type of information they measure. For example, the HPC Jobs realm metrics, describing aggregate usage, consist of measures that are gleaned largely from job accounting data. The SUPReMM realm, meanwhile, contributes metrics describing individual job-level performance data, such as total memory, CPU usage, memory bandwidth, I/O bandwidth, block read and block write rates. These performance data are collected from system hardware counters, then aggregated by XDMoD. The performance of individual jobs can be assessed in the XDMoD Job Viewer. XDMoD can also display aggregated performance data with metrics that 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. Refer to Section III for an introduction to two new data realms we are developing for XDMoD: Storage and Cloud Metrics.

E. Additional modules

Several optional modules extend XDMoD's base capabilities with quality of service and job level performance metrics.

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

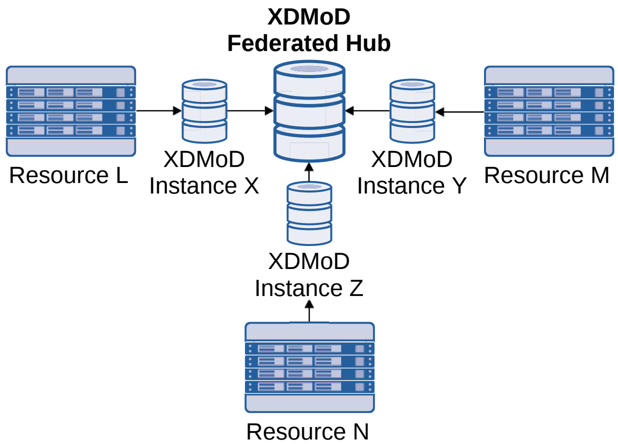

Fig. 2. Simple XDMoD federation example, in which each independent computing resource *L, M, N* is monitored by a satellite XDMoD instance *X, Y, Z*. Each satellite instance is a member of a central XDMoD federation, and its data is replicated to the federated hub's database.

Namely, the Application Kernel module enables quality-of-service monitoring for HPC resources; the SUPReMM (performance) module collects data from system hardware counters to offer viewing and analysis of both aggregate and individual job-level data. [21], [30], [31]

The new XDMoD Federation module further extends the application, providing the ability for multiple disparate XDMoD installations to replicate their raw data to a central, federated hub server. Through its XDMoD user interface, this federated hub then offers an overview of all systems that are included in the federation. Refer to Section II for a thorough treatment of this new, optional module.

II. Federated XDMoD

This section introduces the new Federation module. It discusses the following:

1) What is Federation?
2) Federation example
3) Elements of federation
4) Authenticating for federation
5) Federation use cases

A. *What is Federation?*

Imagine a collection of independent XDMoD instances, each monitoring its own set of computing resources. Each instance may comprise very different underlying resources, configurations, and aggregation levels. Reporting on resource consumption on any one of these XDMoD instances could be done easily through the powerful user interface. Reporting on the collection, however, would be difficult without a way to associate the data (and the aggregations) from these different instances. *Federation* is the answer to reporting on this imagined collection of XDMoD instances. Federation provides a combined, master view of job and performance data collected from individual XDMoD instances. Refer to Figure 2 for a simple example of a federation of three XDMoD instances.

Thus, federation provides resource managers with a unified XDMoD monitor that consolidates the data from a network of disjoint XDMoD instances. Once data is ingested on the individual XDMoD instances, it undergoes live replication to the central federation hub database, where it is then aggregated as appropriate to the requirements of the whole collection (aggregation is customized on each instance using local configuration files). The federation hub can then provide an integrated view of job and performance data collected from entirely independent XDMoD instances. Refer to Figure 3, which illustrates how the data monitored by multiple XDMoD instances are replicated to a single XDMoD federation hub. Section II-C3 describes data aggregation in the context of federation.

The independent XDMoD instances in a federation need have no knowledge of one another. They may be closely related, and managed with the intent to fully share all data. Alternatively, their data can be kept isolated from one another, and only made visible to the federation hub. The only requirement is that each individual XDMoD instance must run the same version of XDMoD.

B. *Federation example*

The simplest federated XDMoD example can be envisioned as a set of computing clusters. Each cluster is instrumented with its own independent XDMoD instance. A central master (or "federation hub") XDMoD instance can combine the data from the independent instances and provide an amalgamated view of their joint contents. This is referred to as a "fan-in" replication topology, in which the satellite databases are replicated and combined into a single federated hub database. The federation hub does not alter the raw, replicated data from the individual instances; it only enables reporting on the entire ensemble of their data, in a single location. A federation of three XDMoD instances is shown in Figure 2.

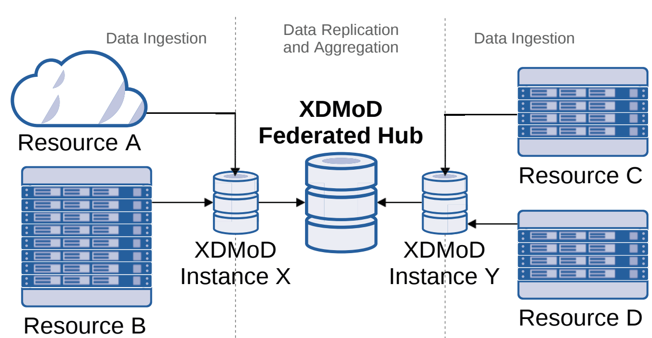

Fig. 3. Example data flow from heterogeneous computing resources to an XDMoD federated hub. XDMoD instances X and Y ingest data into their databases from the computing resources that they monitor. Following ingestion on the satellite instances, job data are replicated to the Federated Hub's database, where they are aggregated for use in the Federated XDMoD user interface.

The federated hub provides a unified view of satellite XDMoD instances and their resources, even while the individual instances retain full functionality and control. On each satellite instance associated with the federation, users access the XDMoD instance's web interface, with its charting, exploration, export, and reporting of resource consumption metrics. Users logging into a satellite XDMoD instance have access to the standard functionality for all metrics on associated resources; and users logging into the federation hub have access to XDMoD functionality on the unified overview of all the federation data.

C. *Elements of federation*

Federated XDMoD offers substantial opportunity for configuration and customization, so the federation can reflect the underlying computing resources monitored by each satellite XDMoD instance. We touch on a number of those considerations here, describing the component parts and configuration issues behind XDMoD federation's design.

1) Database replication: The technology we chose for replicating XDMoD instance data into the federation master hub is Continuent's Tungsten Replicator, an open-source utility. [32] Tungsten reads binary logs on the XDMoD instance databases, copying their tables into new, uniquely named schemas (one schema per XDMoD instance) on the XDMoD federation hub's database. Tungsten supports renaming the data schema during transfer, and selective replication of data from satellite instances, both of which we have opted to do for federation. During initial development, all data in the satellite instances' HPC Jobs realm is being replicated to the federation hub (other realms' data, and user profile information, are presently excluded). While other solutions exist [33], Tungsten Replicator offers the needed features and is compatible with numerous databases.

2) Loose vs. tight federation: While ingested data will be provided to the federation hub using live replication ("tight" federation), this is not the only way to provide instance data in the federation. Instead, log files or database dumps could be periodically shipped to the federation hub, and batch processed there to make their data available to the federation. This latter method would be considered "loose" federation. A heterogeneous model could also be employed, in which a federation hub is provided with data using loose federation from some member instances and tight federation from others.

3) Aggregation levels: Data aggregation is a key data processing step in which XDMoD pre-bins raw dimension data, enabling the application to respond quickly to complex user queries. Every day, aggregation processes run against newly ingested data in the XDMoD data warehouse, binning numeric data in aggregation tables. XDMoD can then use these tables to group metrics by appropriately-sized dimensions. For example, job wall time is aggregated into bins that reflect the typical wall times on the underlying resources; refer to Table I for an example.

TABLE I
EXAMPLE AGGREGATION LEVELS ON XDMOD FEDERATION HUB AND SATELLITE INSTANCES

Job Wall Time aggregation level		
Instance A	**Instance B**	**Federation Hub**
1-60 seconds	-	-
1-60 minutes	-	0-60 minutes
1-5 hours	-	1-5 hours
-	1-10 hours	5-10 hours
-	10-20 hours	10-20 hours
-	20-50 hours	20-50 hours

The individual instances in a federation may be configured to aggregate their data differently. A simple job wall time example illustrates different aggregation levels in an XDMoD federation; refer to Table I. Suppose Instance A provides XDMoD data to a federation. Instance A monitors resources having a wall time limit of 5 hours, so its job wall time aggregation levels are configured at 1-60 seconds; 1-60 minutes; and 1-5 hours. Another XDMoD instance in the same federation, Instance B, monitors resources having a wall time limit of 50 hours. So, job wall times on Instance B are configured to use aggregation levels of 1-10 hours; 10-20 hours; and 20-50 hours. The federation hub to which Instances A and B federate their data defines its own set of aggregation levels that best represent all the data from the federation's component instances. All raw instance data are fully replicated to the master, then aggregated there, according to the federation hub's aggregation levels, so no data are lost or changed. Users of instances A, B, and the federation hub can then plot metrics grouped by wall times that suit the underlying resources monitored by each instance.

Aggregation levels, which are managed by JSON configuration files, apply only to numeric dimensions, such as job wall time, job size (core count), CPU User value, and peak memory usage. Deciding on the aggregation levels that best suit an XDMoD instance is a task for the system administrator at installation time; aggregation levels are fully configurable on each instance and on the federation hub. Aggregation levels may be changed. If, for example, job wall time aggregation levels must be redefined on the federation hub to accommodate a new satellite instance, the administrator will update the appropriate configuration file on the federation hub, then re-aggregate all raw federation data.

4) Routing resource data to different federation hubs: We are developing a configuration strategy to individually manage the destinations of resource data. For instance, data from certain resources managed by a member instance could be selectively excluded from a federation. Refer to Figure 3, which shows the data from multiple resources being replicated to a single XDMoD Federation Hub. Data targeted for federation could be filtered, for example by omitting Resource B and Resource D from federation, but retaining Resource A and Resource C. There are a number of use cases (See Section II-E) that can make use of this model, which could ensure that potentially sensitive data does not ever get replicated to the federation hub. Alternately, data from all resources could be replicated to multiple federation hubs, to provide a live backup or load-balancing strategy for XDMoD instance data.

5) Performance data and federation: The initial release of the federation module replicates only the HPC Jobs realm data to the XDMoD federation hub. Performance data is not yet incorporated in federation. Note that individual job performance data is storage-intensive and quite detailed, including timeseries plots of nine individual job metrics over the life of the job (such as CPU user and memory bandwidth), and the job script for each job. The replication, processing, and storage of this level of detail from the satellite instances runs counter to the goal of federation, which is to provide a summarizing overview of usage and performance. Therefore, we plan to replicate *summarized* performance data to the federated hub database in a subsequent release.

6) Standardizing metrics across a federation: The current discussion of federating HPC jobs data assumes similarly-provisioned systems. What happens when disparate resources are federated together, or when GPUs or other special-purpose hardware are used differently across sites, are absent at one site, or are exposed differently in the resources' queuing systems?

XSEDE has addressed this question for its HPC systems, and the answer is seen in XSEDE XDMoD. In short, XSEDE has benchmarked disparate systems and then derived appropriate conversion factors, so that the resources consumed on different systems can be compared to one another. When an XSEDE resource reports CPU hours consumed, XDMoD applies the appropriate conversion factor to this usage data. This converted data is represented in standardized units called XSEDE Service Units (XDSUs).

For example, Figure 1 plots 2017 utilization in standardized XD SUs, permitting a meaningful comparison of usage across Stampede, Stampede2, and Comet, resources with very different configurations.[2] [8]–[10] When setting up a federation of XDMoD instances, similar care must be taken so that federation metrics make valid comparisons.

In order to make a federation of XDMoD instances useful and meaningful, the metrics being reported must be standardized by including benchmarking-based conversions. Additionally, appropriate queues should be used to indicate access to special purpose hardware. Only then can metrics from differently-provisioned systems be fairly compared.

D. Authenticating for federation

Users must sign on to XDMoD to use most of its advanced features, to see their individual job-level performance data, and to access certain metrics. Therefore, adding data federation capabilities to XDMoD instances requires handling user authentication across federation member instances. We have enabled web-browser Single-Sign On (SSO) for XDMoD by means

[2]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 NUs. NUs measure resources consumed as charged to a roaming allocation. [34]

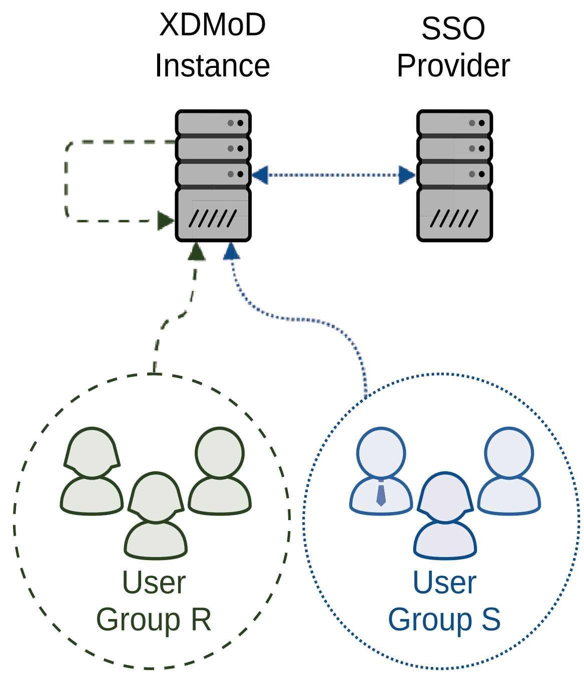

Fig. 4. Two methods of user authentication are shown. User Group R authenticates directly on the XDMoD instance; User Group S authenticates to the same instance using web-browser Single-Sign On (SSO). SSO is being adopted by many Open XDMoD installations.

of Security Assertion Markup Language (SAML), a common standard for exchanging user authentication and authorization data on the web. The resulting SSO authentication solution can be extended across security domains and tailored to the needs of different XDMoD federations. Users retain the ability to authenticate directly on the XDMoD instance; the schematic in Figure 4 shows XDMoD users signing onto an SSO-enabled XDMoD instance using either their local XDMoD password, or their SSO credentials.

1) Strategies for authentication: XDMoD with SSO can be set up to authenticate users using several mechanisms. For example, we have employed two different approaches, Globus for XSEDE XDMoD, Shibboleth for Open XDMoD. The XSEDE implementation uses Globus Auth authentication, which is nearly seamless to the user. Before they can utilize SSO, XSEDE users must simply link their Globus account with their XSEDE credentials, whereupon they can authenticate themselves to XDMoD using their Globus-linked institutional credentials.

On the Open XDMoD instance at CCR, meanwhile, our academic users can specify their university credentials to authenticate via Shibboleth. Shibboleth authentication provides metadata fields describing the associated user, enabling Open XDMoD to pre-populate some filters and fields in the application, and provide a more customized user experience for first-time XDMoD users. In addition to Shibboleth and Globus, we support other SSO mechanisms, such as institu-

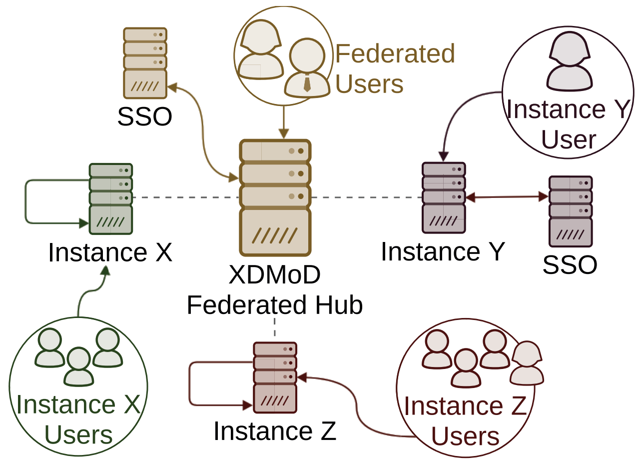

Fig. 5. User authentication to instances of an XDMoD federation. Users of XDMoD Instances X and Z are shown authenticating directly to their XDMoD instances. User Y and the Federated Users are shown authenticating to their respective XDMoD instances using SSO. Users can sign onto an SSO-enabled XDMoD instance using either their local XDMoD password or SSO.

tional Lightweight Directory Access Protocol (LDAP).

2) Adoption of SSO: SSO authentication, while critical for the federation module, is a popular feature in its own right. We have offered SSO authentication since XDMoD version 6.5; at least ten known Open XDMoD installations have implemented it on their non-federated instances. These installations use identity providers such as Keycloak, LDAP, and Shibboleth; presently, an installation can specify only a single SSO authentication source. Users can sign onto an SSO-enabled XDMoD instance using either their local XDMoD password, or their SSO credentials. The schematic in Figure 5 shows users in an XDMoD federation authenticating to member instances through the satellite instances or using SSO.

3) Flexible configuration: In the future, XDMoD administrators will be able to configure multiple SSO authentication sources, to better support a federated core instance that may consist of data originating from multiple institutions that may use varied protocols. Authentication responsibility may rest with the federation hub or with the satellite instances (“identity provider” or “service provider” mode). For example, in one possible configuration, the federation hub can do the job of authenticating users of the federation’s satellite instances. This flexibility will let administrators tailor XDMoD user authentication to fit project, institutional, and federation needs.

4) Authentication vs. identity: While SSO enables *user authentication* across the member instances of an XDMoD federation, handling *user identity* is a separate question. We do not yet offer any automated means of mapping or de-duplicating users from different XDMoD satellite instances in the federated master hub. For example: consider a CCR user who also has an XSEDE allocation. Suppose XSEDE and CCR instances were federated, and we sought to identify all jobs run by that individual across all federated resources. At this time, the user would appear twice in the federation; once as the CCR user, once as the XSEDE user. The work necessary to federate such user identities must be performed separately on the federation database; it is not yet handled by the Federation module, though this is a goal for a future release.

E. Federation use cases

The XDMoD Federation module offers an opportunity to unify monitoring of HPC, cloud, and storage resources in a single place. In the case of a particularly diverse, geographically distributed, or large HPC project, it provides a natural solution to management and reporting. Some use cases for federation include:

1) National computing infrastructure
2) Distributed computing-intensive scientific project
3) Collaborative research cloud

In each of these examples, the XDMoD federation module collects data from multiple sites, performs unifying aggregations, and assembles disparate data sources to provide a consolidated view of a large and diverse computing enterprise. More detailed investigations of these use cases for XDMoD federation are as follows:

1) National computing infrastructure: Research-class machines, high-memory machines, cloud infrastructure, data-intensive machines, and an enterprise class HPC installation are components of a national computing infrastructure. These infrastructure components are widely geographically distributed, in addition to being highly heterogeneous. Management of such an infrastructure, along with funding, subscription, and planning for the next generation, requires a unified view of resource usage that only federation can supply.

2) Distributed computing-intensive scientific project: A large scientific project builds geographically distributed computing locations and data storage farms to support intensive data acquisition and computing. Each installation adjoins a different laboratory, where scientists perform research, or collect and process data from instruments. Concrete examples include telescopes, particle colliders, and large genomics facilities. Once again, unified monitoring and management for such a sizable and diverse project dictate a federated approach.

3) Collaborative research cloud: A research cloud project spans multiple institutions, with complementary computing installations housed at diverse sites. These cloud implementations are instrumented to collect metrics that report usage and performance both to the project’s funding agencies and to the partner institutions. The maintenance and management of the individual sites vary by institution (implying local XDMoD installations for the sites), but the funding agency and project partners require summary reports that describe the project as a whole. This presents another case ripe for federation. In fact, CCR, Cornell, and University of California, Santa Barbara (UCSB) are currently collaborating on a geographically-distributed cloud that will be monitored by federated XDMoD; see Section III-B.

4) Other uses: Federation is a sensible model when regulatory issues or security concerns dictate that separation be maintained between data sources. When data are replicated to the master instance, only the needed data are included; sensitive data may be omitted from the replication step.

The act of federation can also be regarded as a backup procedure. Since the XDMoD federation hub does not summarize or reduce the data it acquires from the member instances, the hub itself could be used to regenerate the databases for the member instances. This functionality may be offered as a part of the federation module.

III. Federations of Heterogeneous Resources

Modern computing centers encompass various types of resources, which may include HPC, GPUs, specialized nodes, cloud facilities, high-performance storage, and others. For example, many centers are now adding cloud-based facilities to complement their HPC facilities or even replace them entirely. Most computing centers also maintain substantial storage facilities; some specialize in computations involving large data sets, large memory nodes, or other specializations.

Although XDMoD was conceived as a monitoring system for HPC facilities, it is natural to extend its capabilities, adding metrics that describe the other, non-HPC resources that feature prominently in computing centers. Supporting these metrics together with Jobs, Performance, and Allocations data will offer a complete overview of the usage of both the computational and non-computational resources in a modern computing facility. Going further, combining these metrics and multiple XDMoD instances in an XDMoD federation enables the assembly of unified views of a large computing enterprise. Suitable reporting and auditing can be performed in order for funders, stakeholders, and management to assess and direct the evolution of their computing intrastructure.

Accordingly, the XDMoD team is developing additional data realms for cloud resources and for storage, helping to fulfill the need for non-HPC resource monitoring. These realms are described below.

A. Storage facilities

The XDMoD team is developing a Storage realm to harvest and present metrics describing compute storage. The storage realm will assist centers in tracking storage utilization, user quota utilization, and eventually, storage performance and metadata measures as well. Furthermore, XDMoD installations will be able to monitor both volatile and persistent storage. The initial set of Storage realm metrics includes:

- File Count
- Logical and Physical Usage
- Hard and Soft Quota Thresholds
- Logical Quota Utilization
- User Count

Supported dimensions for drill-down on these metrics are Resource (Filesystem), Mountpoint, Resource Type, User, PI, and System Username. The storage realm is being developed against CCR's Isilon and GPFS storage, both persistent and scratch. Refer to Figure 6 for a plot of the file count and physical storage usage at CCR, aggregated monthly in 2017.

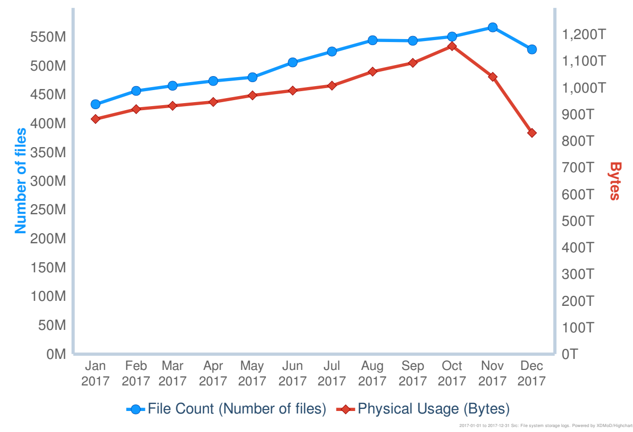

Fig. 6. XDMoD's storage realm, now in development, will enable charting of storage quota and usage metrics. This XDMoD plot shows CCR's file count (blue circles) and physical storage usage (red diamonds), by month of 2017.

Storage data will be acquired from monitoring tools (e.g. TACC_Stats, PCP) or filesystem APIs, then populated in a fashion independent of the storage filesystem. Data from filesystems such as Isilon, GPFS, Lustre, and Ceph can be accommodated; installations must only ensure their data validates against our provided JSON schema. We will provide tools for the subsequent ingestion of these JSON data into XDMoD. An eventual goal is to include storage performance metrics in this realm, notably, metadata such as file creations, deletions, attribute requests, and throughput. An additional caveat for bursty storage involves the sampling frequency for filesystem data. The data sampling rate must be high enough to capture filesystem behavior, without monopolizing available bandwidth or compromising processing.

B. Cloud facilities

Computational centers that rely on cloud computing or a combination of HPC and cloud computing will benefit from the ability to monitor Cloud Metrics. However, monitoring cloud resources, with their highly configurable Virtual Machines (VMs), is a disparate task from tracking traditional HPC batch jobs. To take a simple example, two VMs on a single cloud resource may be configured with vastly different hardware and software characteristics, such as allocated memory and installed operating system. Certain characteristics of a VM, such as allocated memory, can even be changed during the life of the VM. This contrasts with the fixed characteristics of compute nodes in a traditional HPC cluster.

Furthermore, in cloud computing, wall hours indicate the amount of time that a *VM* spent running, whereas in HPC, wall hours refer the amount of time that a *job* spent running. While VMs are used to run jobs, the analogy between the runtime of a VM and that of a compute job is not complete. Specifically, a user may create a VM for a particular job, then let the VM continue to run after the job itself completes; thus, the VM's wall time and the job's wall time may differ. VMs can also be stopped, restarted, and paused, so their changes of state are

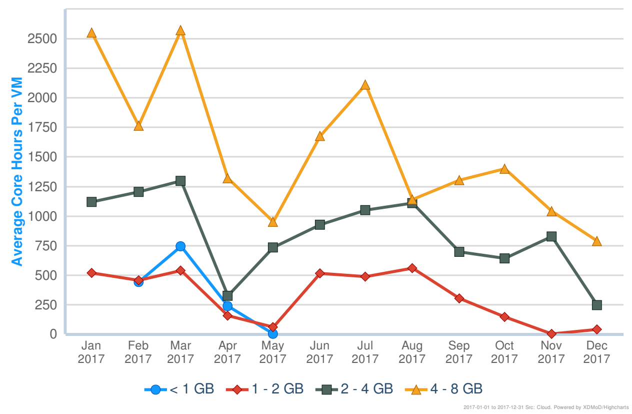

Fig. 7. XDMoD's cloud metrics realm, now in development, will enable charting of cloud usage metrics. Shown in this XDMoD plot are average core hours used per VM, by VM memory size, CCR research cloud, 2017. VM memory size is aggregated into bins of <1 GB (blue circles), 1-2 GB (red diamonds), 2-4 GB (grey squares), and 4-8 GB (yellow triangles).

important to monitor. Clearly, metrics for cloud resources must account for their highly dynamic nature, tracking measures such as VM image configuration information and state, as well as VM wall time.

We are developing an XDMoD Cloud Metrics realm to acquire and present metrics relevant to the assessment and management of cloud computing. For the initial release, the metrics offered in the Cloud realm acknowledge the contrasts with traditional HPC, including measures that track VM utilization figures, such as:

- Average Cores per VM
- Average Cores, Disk, or Memory Reserved[3]
- Core or Wall Hours: Total
- Cores: Total
- Number of VMs Ended, Running, or Started

Other metrics are being considered for addition in subsequent releases, such as Operating System, VM Events, Count of State Changes, and Time Spent per State.

The dimensions available for grouping and drill-down in this realm must also account for the characteristics of cloud computing, and include Instance Type; Project; Resource; Submission Venue; User; and VM Size (Cores or Memory). See Figure 7 for an example plot showing average VM core hours by VM memory size.

The Cloud Metrics realm is populated with usage data from CCR's installation of the widely-deployed Open Stack platform API, backed by the Ceph storage platform. [35] The work on this realm is motivated by an NSF Data Infrastructure Building Blocks (DIBBs) project, which will shortly be the first XDMoD federation featuring cloud-based resources. This project, called "Aristotle", entails the development of three integrated computational clouds located at CCR, Cornell, and UCSB. The entire project will be monitored by Federated XDMoD.

[3]weighted by Wall Hours

Two additional components will eventually be included in the XDMoD cloud realm, completing the picture of cloud computing utilization. First, the XDMoD cloud realm will track VM reservation, or payment, information. This piece of the puzzle will enable centers to evaluate whether users purchase more capacity than they use. Second, raw job performance data for cloud jobs will be linked to the cloud realm, so that VMs and the actual jobs run on them can be compared directly. This piece will enable centers to compare actual VM usage with available cloud capacity. These enhancements to the cloud realm will be available in a future release.

In particularly diverse, geographically distributed, or large HPC projects, federated XDMoD combined with these new cloud and storage metrics provides consolidated management and reporting. These metrics will present a more complete view of the entire computing facility, folding in new data that describe additional compute and non-compute resources.

IV. Conclusion

XDMoD is a comprehensive tool for managing computational resources. It collects, aggregates, and presents metrics describing computational jobs, the resources they consume (computation, memory, disk, network, etc.), wait times, and quality of service. The XDMoD web interface offers tools for charting, visualizing, exporting, and reporting these metrics. Users can chart metrics for varied time ranges, and across any computing resources, using interactive filtering, grouping, and drill-down. With XDMoD's Job Viewer, users can probe performance data about a job's executable, its accounting data, job scripts, application, and timeseries plots of metrics such as CPU user, flops, parallel file system usage, and memory usage. Open XDMoD installations can be found at HPC centers worldwide, helping hundreds of academic, industrial, and governmental centers run, assess, and direct their high-performance computing. With its charter to monitor the U.S. XSEDE supercomputer network, and to provide workload analyses for the full scope of the XSEDE project, XDMoD is a proven tool for HPC evaluation and management.

Our recent work on the federation of XDMoD instances provides resource managers with new capabilities: a federated XDMoD hub collects and consolidates data from a network of separate XDMoD installations. Charting and reporting on Federated XDMoD can range from a single resource to the entire network. Federation supports loosely or tightly coupled resources, and a variety of information sharing and authentication paradigms. The federation, combined with new data realms, enables institutions to collect HPC, Storage, and Cloud metrics into a unified view of the computing enterprise, providing tools to assess and direct the evolution of large computing projects, even across the boundaries of institutions and their protocols and networks. Even so, the individual XDMoD instances associated with the federation retain the control they require to monitor and evaluate computing in their own centers. Medium and large projects such as national computing infrastructures, large scientific computing projects, and collaborative research clouds stand to benefit from the expanded capabilities offered by federated XDMoD.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under grant numbers OCI 1025159 (Technology Audit Service), OCI 1203560 (SUPReMM), ACI 1445806 (XD Net Metrics Service for High Performance Computing), and ACI 1541215 (Data Infrastructure Building Blocks).

REFERENCES

[1] Center for Computational Research, University at Buffalo, "XD Metrics on Demand (XDMoD)," https://xdmod.ccr.buffalo.edu, accessed 17 July 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, and F. Wang, "Comprehensive, open-source resource usage measurement and analysis for HPC systems," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 13, pp. 2191–2209, 2014, cPE-14-0027.R1. [Online]. Available: https://dx.doi.org/10.1002/cpe.3245

[3] 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, and A. Zimmerman, "Performance metrics and auditing framework using applications kernels for high performance computer systems," 2012.

[4] 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, and 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.

[5] 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, and 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.

[6] "XSEDE," https://www.xsede.org/, accessed 17 July 2018.

[7] 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, and R. T. Evans, "Open XDMoD: A tool for the comprehensive management of high-performance computing resources," *Computing in Science and Engineering*, vol. 17, no. 4, pp. 52–62, 2015. [Online]. Available: https://dx.doi.org/10.1109/MCSE.2015.68

[8] "SDSC Comet Computing Cluster," http://www.sdsc.edu/support/user_guides/comet.html, San Diego Supercomputer Center at UC San Diego.

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

[10] "TACC Stampede Supercomputer," https://www.tacc.utexas.edu/systems/stampede, Texas Advanced Computing Center, Austin.

[11] "Performance Co-Pilot," https://pcp.io/, accessed 16 July 2018.

[12] M. L. Massie, B. N. Chun, and D. E. Culler, "The Ganglia distributed monitoring system: design, implementation, and experience," *Parallel Computing*, vol. 30, no. 7, pp. 817 – 840, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167819104000535

[13] The Ganglia Project, "Ganglia Monitoring System," https://ganglia.info, accessed 16 July 2018.

[14] A. Agelastos, B. Allan, J. Brandt, P. Cassella, J. Enos, J. Fullop, A. Gentile, S. Monk, N. Naksinehaboon, J. Ogden, M. Rajan, M. Showerman, J. Stevenson, N. Taerat, and T. Tucker, "The lightweight distributed metric service: A scalable infrastructure for continuous monitoring of large scale computing systems and applications," in *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov 2014, pp. 154–165.

[15] Sandia National Laboratories, Albuquerque, NM, "OVIS (LDMS)," https://github.com/ovis-hpc/ovis/wiki, accessed 16 July 2018.

[16] K. Agrawal, M. R. Fahey, R. McLay, and D. James, "User environment tracking and problem detection with XALT," in *Proceedings of the First International Workshop on HPC User Support Tools*, ser. HUST '14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 32–40. [Online]. Available: https://dx.doi.org/10.1109/HUST.2014.6

[17] "XALT," https://github.com/Fahey-McLay/xalt, accessed 16 July 2018.

[18] "Zabbix," https://www.zabbix.com/, accessed 16 July 2018.

[19] T. Evans, W. L. Barth, J. C. Browne, R. L. DeLeon, T. R. Furlani, S. M. Gallo, M. D. Jones, and 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*, ser. HUST '14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 13–21. [Online]. Available: https://dx.doi.org/10.1109/HUST.2014.7

[20] Texas Advanced Computing Center, University of Texas, Austin, "TACC _STATS," https://github.com/TACCProjects/tacc_stats, accessed 16 July 2018.
[21] T. Furlani, M. Jones, S. Gallo, A. Bruno, C. Lu, A. Ghadersohi, R. Gentner, A. Patra, R. DeLeon, G. Laszewski, L. Wang, and A. Zimmerman, "Performance metrics and auditing framework using application kernels for high-performance computer systems," *Concurrency and Computation: Practice and Experience*, vol. 25, p. 918, 2013.
[22] "HOPSA-Holistic Performance System Analysis," http://www.vihps.org/projects/hopsa/overview, accessed 17 July 2018.
[23] "Threadspotter," http://threadspotter.paratools.com/, accessed 17 July 2018.
[24] "Scalasca," http://www.scalasca.org/, accessed 17 July 2018.
[25] "Paraver," https://tools.bsc.es/paraver, accessed 17 July 2018.
[26] "Vampir," http://www.vampir.eu/, accessed 17 July 2018.
[27] Center for Computational Research, University at Buffalo, "Open XDMoD," https://open.xdmod.org/, accessed 17 July 2018.
[28] 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, and 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*, ser. PEARC17. New York, NY, USA: ACM, 2017, pp. 6:1–6:8. [Online]. Available: http://doi.acm.org/10.1145/3093338.3093348
[29] N. A. Simakov, J. P. White, R. L. DeLeon, S. M. Gallo, M. D. Jones, J. T. Palmer, B. D. Plessinger, and T. R. Furlani, "A workload analysis of NSF's innovative HPC resources using XDMoD," *CoRR*, vol. abs/1801.04306, 2018. [Online]. Available: http://arxiv.org/abs/1801.04306
[30] N. A. Simakov, J. P. White, R. L. DeLeon, A. Ghadersohi, T. R. Furlani, M. D. Jones, S. M. Gallo, and A. K. Patra, "Application kernels: HPC resources performance monitoring and variance analysis," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 5238–5260, 2015, cPE-14-0402.R1. [Online]. Available: https://dx.doi.org/10.1002/cpe.3564
[31] Center for Computational Research, University at Buffalo, "XDMoD SUPReMM Module," http://supremm.xdmod.org/, accessed 17 July 2018.
[32] "Tungsten Replicator," https://alternativeto.net/software/tungsten-replicator/, accessed 17 July 2018.
[33] "SymmetricDS Replicator," http://www.symmetricds.org/, accessed 17 July 2018.
[34] XSEDE (Extreme Science and Engineering Discovery Environment), "XSEDE SU Converter," https://www.xsede.org/su-converter, accessed 17 July 2018.
[35] "OpenStack," https://www.openstack.org/, accessed 17 July 2018.