Using XDMoD to Facilitate XSEDE Operations, Planning and Analysis

Thomas R. Furlani[1], Barry L. Schneider[2], Matthew D. Jones[1], John Towns[3], David L. Hart[4], Steven M. Gallo[1], Robert L. DeLeon[1], Charng-Da Lu[1], Amin Ghadersohi[1], Ryan J. Gentner[1], Abani K. Patra[5], Gregor von Laszewski[6], Fugang Wang[6], Jeffrey T. Palmer[1], Nikolay Simakov[1]

[1]Center for Computational Research, University at Buffalo, SUNY, Buffalo, NY

[2]Office of Cyberinfrastructure, National Science Foundation, Arlington, VA 22230

[3]NCSA, University of Illinois, Urbana IL 61801

[4]National Center for Atmospheric Research, Boulder, CO 80307-3000

[5]Mech. & Aerospace. Eng. Dept., University at Buffalo, SUNY, Buffalo, NY 14260

[6]Pervasive Technology Institute, Bloomington, IN 47408

ABSTRACT

The XDMoD auditing tool provides, for the first time, a comprehensive tool to measure both utilization and performance of high-end cyberinfrastructure (CI), with initial focus on XSEDE. Here, we demonstrate, through several case studies, its utility for providing important metrics regarding resource utilization and performance of TeraGrid/XSEDE that can be used for detailed analysis and planning as well as improving operational efficiency and performance.

Measuring the utilization of high-end cyberinfrastructure such as XSEDE helps provide a detailed understanding of how a given CI resource is being utilized and can lead to improved performance of the resource in terms of job throughput or any number of desired job characteristics. In the case studies considered here, a detailed historical analysis of XSEDE usage data using XDMoD clearly demonstrates the tremendous growth in the number of users, overall usage, and scale of the simulations routinely carried out. Not surprisingly, physics, chemistry, and the engineering disciplines are shown to be heavy users of the resources. However, as the data clearly show, molecular biosciences are now a significant and growing user of XSEDE resources, accounting for more than 20 percent of all SUs consumed in 2012. XDMoD shows that the resources required by the various scientific disciplines are very different. Physics, Astronomical sciences, and Atmospheric sciences tend to solve large problems requiring many cores. Molecular biosciences applications on the other hand, require many cycles but do not employ core counts that are as large. Such distinctions are important in guiding future cyberinfrastructure design decisions.

XDMoD's implementation of a novel application kernel-based auditing system to measure overall CI system performance and quality of service is shown, through several examples, to provide a useful means to automatically detect under performing hardware and software. This capability is especially critical given the complex composition of today's advanced CI. Examples include an application kernel based on a widely used quantum chemistry program that uncovered a software bug in the I/O stack of a commercial parallel file system, which was subsequently fixed by the vendor in the form of a software patch that is now part of their standard release. This error, which resulted in dramatically increased execution times as well as outright job failure, would likely have gone unnoticed for sometime and was only uncovered as a result of implementation of XDMoD's suite of application kernels.

XSEDE '13, July 22 - 25 2013, San Diego, CA, USA

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]: [**Performance of Systems**]: Reliability, availability, and serviceability; C.5.1 [**Computer System Implementation**]: Large and Medium ("Mainframe") Computers---Super (very large) computers; K.6.1 [**Management of Computing and Information Systems**]: Project and People Management---Strategic information systems planning; K.6.4 [**Management of Computing and Information Systems**]: System Management---Quality Assurance.

General Terms

Management, Measurement, Performance, Reliability, Standardization, Verification.

Keywords

XSEDE, XDMoD, Technology Audit Service, HPC Metrics, Application Kernels, CI performance metrics

1. INTRODUCTION

While individual tools to measure the utilization, performance, and to a lesser extent the scientific impact of high-end cyberinfrastructure (CI) have been developed over the years [1-13], a comprehensive auditing framework that includes all three of these important measures of the efficacy and operational efficiency of CI had been lacking. The National Science Foundation (NSF) recognized the value of this capability and through the Technology Audit Service (TAS) of XSEDE made a significant investment in developing tools and infrastructure to make this capability easily accessible to a broad range of users and resource managers.

The National Science Foundation (NSF) recognized the value of this capability and through the Technology Audit Service (TAS) of XSEDE made a significant investment in developing tools and infrastructure to make this capability easily accessible to a broad range of users and resource managers. In this context, the XDMoD (XSEDE Metrics on Demand) auditing tool provides a comprehensive framework for auditing the utilization and performance of high-end cyberinfrastructure [14]. It is designed

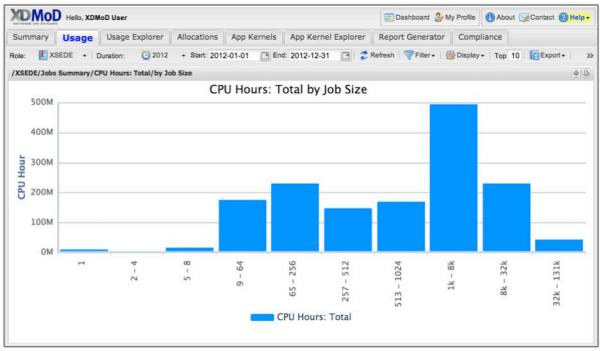

Figure 1 The XDMoD interface. Plot shows total CPU hours provided for all of XSEDE broken down by job size in 2012.

to meet the following objectives: (1) provide the user community with a tool to more effectively and efficiently use their allocations and optimize their use of CI resources, (2) provide operational staff with the ability to monitor and tune resource performance, (3) provide management with a diagnostic tool to facilitate CI planning and analysis as well as monitor resource utilization and performance, and (4) provide metrics to help measure scientific impact. Currently XDMoD is designed to function within the XSEDE framework, although a future version will provide academic and industrial HPC centers with similar functionality. Application of XDMoD to other types of cyberinfrastructure are also being investigated.

In this paper we present several XDMoD usage case studies to demonstrate XDMoD's utility to aid in analysis, planning, and performance tuning as applied to the CI of XSEDE. The first case study is an analysis of historical usage data from the TeraGrid and the follow-on XSEDE program. The second case study demonstrates the utility of the XDMoD framework for facilitating system performance assessment through the implementation of application kernels. The third and final case study shows, through several examples, how, like most analysis tools, care must be exercised in the interpretation of data generated by the XDMoD tool. The final section covers conclusions and future work. We begin with an overview of XDMoD to provide a context for the discussions that follow.

2. XDMoD OVERVIEW

Here we present a brief overview of XDMoD, a more detailed description can be found in Reference [14]. The XDMoD portal [15] provides a rich set of features accessible through an intuitive graphical interface, which is tailored to the role of the user. Currently six roles are supported: Public, User, Principal Investigator, Campus Champion, Center Director and Program Officer. Metrics provided by XDMoD include: number of jobs, service units (see next section for definition) charged, CPUs used, wait time, and wall time, with minimum, maximum and the average of these metrics, in addition to many others. These metrics can be broken down by: field of science, institution, job size, job wall time, NSF directorate, NSF user status, parent science, person, principal investigator, and by resource.

A context-sensitive drill-down capability has been added to many charts allowing users to access additional related information simply by clicking inside a plot and then selecting the desired metric. For example, in Figure 1, which is a plot of total CPU hours in 2012 by job size for all XSEDE resources, one can select any column in the plot and obtain additional information (such as field of science) specific to the range of data represented by the column. Another key feature is the Usage Data Explorer that allows the user to make a custom plot of any metric or combination of metrics filtered or aggregated as desired.

The XDMoD framework is also designed to preemptively identify

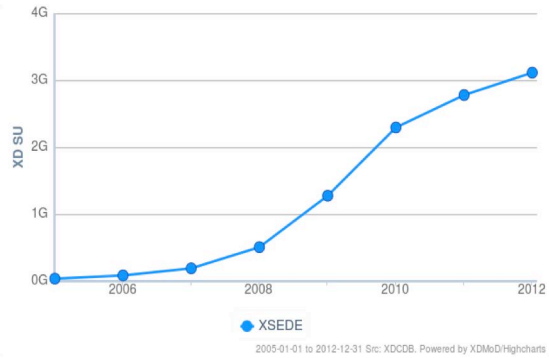

Figure 2 Total XSEDE usage in billions (G) of service units (SUs) for the years 2005 - 2012. Note: For the purpose of this paper, service units should be understood as core hours with the caveat that the value of SU varies across resources and over time as technology advances.

potential bottlenecks from user applications by deploying customized, computationally lightweight "application kernels" that continuously monitor CI system performance and reliability from the application users' point of view. The term "application kernel" is used in this case to represent micro and standard benchmarks that represent key performance features of modern scientific and engineering applications, and small but representative calculations carried out with popular open-source high performance scientific and engineering software packages. The term "computationally-lightweight" is used to indicate that the application kernel runs for a short period (typically less than 10 min) on a small number of processors (less that 128 cores) and therefore requires relatively modest resources for a given run frequency (say once or twice per week). Accordingly, through XDMoD, system managers have the ability to proactively monitor system performance as opposed to having to rely on users to report failures or underperforming hardware and software. The detection of anomalous application kernel performance is being automated through the implementation of process control techniques. In addition, through this framework, new users can determine which of the available systems are best suited to address their computational needs.

Preliminary versions of metrics that focus on scientific impact, such as publications, citations and external funding, are now being incorporated into XDMoD to help quantify the important role advanced cyberinfrastructure plays in advancing research and scholarship.

3. XDMoD USAGE CASE STUDIES

3.1 Data History of TG/XD Usage: Providing a Foundation for Data-driven CI Planning

Measuring the utilization of high end cyberinfrastructure such as XSEDE is obviously important as it helps provide a basic understanding of how CI resources are being utilized and can lead to improved performance of the resource in terms of job throughput or any number of desired job characteristics. Indeed, as described by Katz et. al. [16], the ability to readily measure usage modalities for cyberinfrastructure leads to a greater understanding of the objectives of end users and accordingly insight into the changes in CI to better support their usage. Furthermore, given the rapid pace of hardware upgrades, access to reliable, extensive data from past usage is also essential for planning purposes.

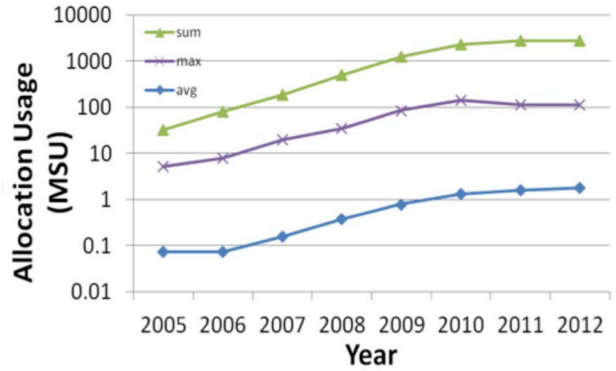

Figure 3 Largest, average and total SU allocations over time. The average and largest allocations have increased by more than a factor of 10 over the time period above.

XSEDE is the most advanced, powerful, and robust collection of integrated advanced digital resources and services in the world [17]. It is a single virtual system that scientists can use to interactively share computing resources, data, and expertise. XDMoD, through the TeraGrid/XSEDE central database, provides a rich repository of usage data. Here we demonstrate, through several examples, the extent of the data as well as its utility for planning. In what follows, the terminology Service Units (SUs) is liberally used. It should be interpreted as core hours with the caveat that an SU is defined locally in the context of a particular machine. Thus, the value of an SU varies across resources utilizing varying technologies and, by implication, varies over time as technology advances. We begin with a historical look at utilization. The data displayed in Figure 2 shows the total number of service units (SUs) delivered to the community on a year-by-year basis from 2005 through 2012.

The large increase in the number of delivered SUs beginning in 2008 is not surprising since it was during that period that the NSF funded two very large computational resources, Ranger at TACC and Kraken at UTK/ORNL, which provided more cycles than the previous set of resources combined. Figure 3 is a plot which is designed to provide an indication of the largest, average and total usage on XSEDE resources, showing for example that the largest XSEDE allocation has increased by more than an order of magnitude since 2005 to more than 100M SUs. Remarkably, the largest allocation of a single user today exceeds the total usage of all users in 2005 and 2006.

The TeraGrid/XSEDE usage by parent science is shown in Figure 4. Parent Science is an aggregation of fields of science defined by a previous (ca. 1995) organizational structure of the NSF and corresponds to NSF divisions (or previous divisions). This aggregation is used to categorize the TeraGrid/XSEDE allocations and usage. Given the modest number of organizational changes at NSF at the divisional level, the classifications in Figure 4 and Figure 5 can easily be related to current NSF divisions. Physics and molecular biosciences are the top consumer science fields using between 600M-700M SUs per year after the Ranger and Kraken resources were deployed. Usage by the molecular biosciences has become comparable to physics in recent years as the bioscientists become more dependent on simulation as a part of their scientific arsenal. Materials research is also a significant and growing consumer of CPU time.

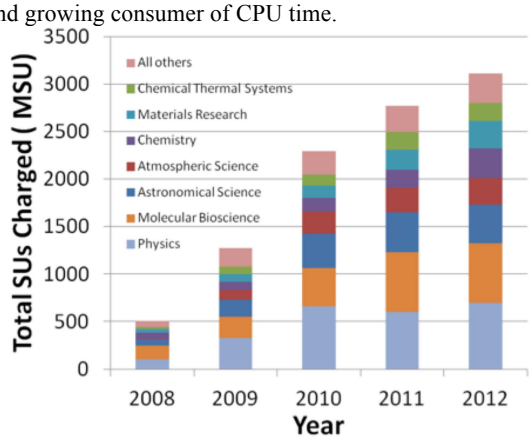

Figure 4 Total SU Usage by Parent Science

However, as Figure 5 shows, the average core count by parent science varies widely. Note, as shown in [18], [19] and Section 3.3 below, when examining the average core counts run on XSEDE resources, it can be misleading to report only the average core count for a particular metric or resource. Accordingly, we find it more informative to compute the average core count by weighting each job by the total SUs it consumes. Traditionally, fields in the Mathematical and Physical Sciences (MPS)

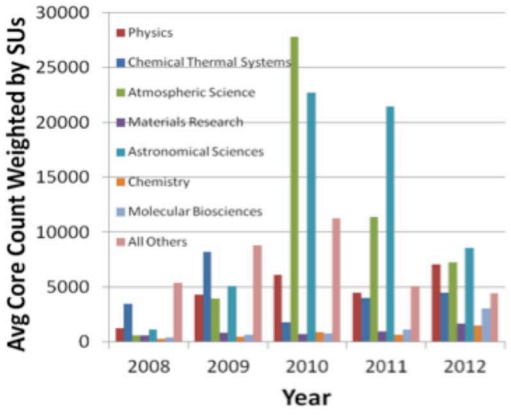

Figure 5 Average Core Count (weighted by SUs) by Parent Science

directorate of NSF have been thought to be the largest users of XSEDE computational resources. While MPS users are still significant, it is clear from Figure 4 that the molecular biosciences community, which falls predominantly within the Biological Sciences Directorate, has been on the rise for some time and has

harnessed the capabilities of these resources to advance the field. Researchers in this area have passed their colleagues in all of the divisions within MPS with the exception of Physics, which it is clearly on par with at this point. However, from Figure 5 it is also clear that the type of jobs that are typical of the molecular biosciences use a relatively small number of compute nodes. Physics and fluid dynamics (which dominates Chemical Thermal Systems), fields long characterized by the need to solve complex partial differential equations, typically require careful attention being paid to parallelization and by default, large core count jobs. Many of the biological applications are dominated by complex workflows, involving many jobs but relatively few cores, often with large memory per core. In general, the average number of cores used is moderate in size. It is interesting to speculate on the reasons that is the case. Certainly, it could be algorithmic. As we know, the development of effective software is an extremely time consuming and human intensive problem. Also, there are practical issues of turnaround. Many users have learned to structure their jobs for optimal turnaround and that often can be in conflict with optimal core count use. In addition, the use of average core count as a measure of the need for machines with many processors, can be misleading. The job mix submitted by most users ranges over core count. Often it is necessary to run a significant number of smaller core count jobs as a preliminary to the single large core count run. These all contribute to lowering the average core count number.

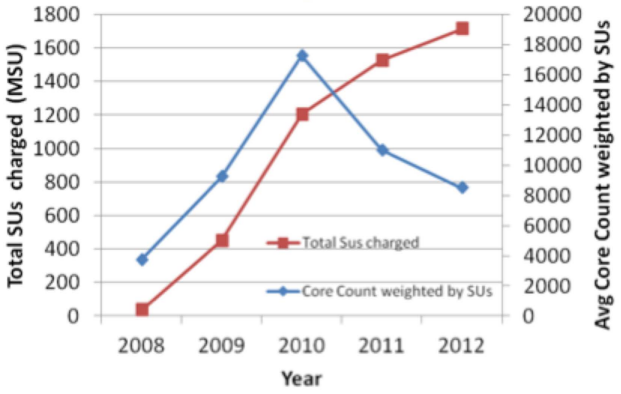

Figure 6 Kraken Usage: Total SUs and Average Core Count Weighted by SUs

In this section, three of the XSEDE resources, namely Kraken, Lonestar4, and Steele, have been chosen as illustrative of what appears in the current NSF portfolio and importantly, what each brings to the mix that is unique and valuable to specific users. It has been characteristic of the NSF program to try to provide a mix of compute systems each designed to be optimal for specific types of job flows. Figures 6 to 8 show total usage and average core count (weighted by SUs) on each of these three resources. A number of scientific disciplines are positioned to use systems containing very large numbers of cores and requiring fast communications. For such users, systems such as Kraken and to a lesser extent Lonestar4 are ideal, and this is reflected in the average core count. With the decommissioning of Ranger and the near term future decommissioning of Kraken, Stampede and Blue Waters will likely be the systems of choice for such users. Lonestar4, a more recent addition to the portfolio, is a smaller resource in terms of core count than Kraken but with its more modern CPU (Westmere) has become the most highly requested resource in XSEDE, perhaps as much as 10 times over-requested. Clearly, users not needing many thousands of cores can make very effective use of Lonestar4 (average SU weighted core usage around 750 for NSF users), and since its performance is between 2 to 4 times faster than Kraken per core, it is preferred for those types of jobs. For users that are primarily conducting high throughput serial computations, the Purdue facilities such as Steele are preferred, as shown in Figure 8. The PSC system, Blacklight, (not shown) is a small core-count, very large shared memory SGI system and also a very recent addition to the NSF portfolio. It is ideal for users needing random access to very large data sets and to problems involving the manipulation of large, dense matrices which must be stored in central memory. So, problems in graph theory, large data sorts, quantum chemistry, etc., need such a resource to perform optimally. While the resources are dominated by disciplines that can make effective use of what was once called ”big iron” there are also many users that fall outside that category. This has always been part of the mantra of the TeraGrid/XSEDE program (deep and wide) and strong efforts continue in these directions today with the Open Science Grid (OSG), science gateways, campus champions and advanced user support programs.

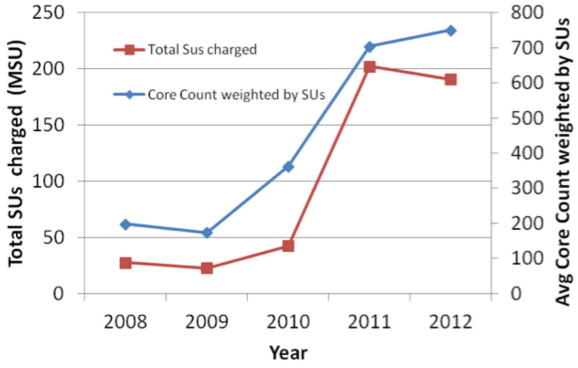

Figure 7 Lonestar4 Usage: Total SUs and Average Core Count Weighted by SUs

An interesting observation looking at Figures 6 to 8 is the fact that early on in the life of a resource, the average core count is larger than in the later life period. In the initial phase, the resource tends to have fewer users, and by design those users are chosen to push the capability limits of the resource. As the machine ages and particularly as newer resources are deployed, the profile of the user base evolves: the capability users are moved to the newer resources and the broader user community has prepared itself to run on the machine. Thus, again by design, the average core count decreases to accommodate the larger user base. The leveling off of SU count in most resources is typical.

3.2 Facilitating System Operation and Maintenance

Most modern multipurpose HPC centers mainly rely upon system related diagnostics, such as network bandwidth utilized, processing loads, number of jobs run, and local usage statistics in order to characterize their workloads and audit infrastructure performance. However, this is quite different from having the means to determine how well the computing infrastructure is operating with respect to the actual scientific and engineering applications for which these HPC platforms are designed and operated. Some of this is discernible by running benchmarks; however in practice benchmarks are so intrusive that they are not run very often (see, for example, Reference [20] in which the application performance suite is run on a quarterly basis), and in many cases only when the HPC platform is initially deployed. In addition benchmarks are typically run by a systems administrator

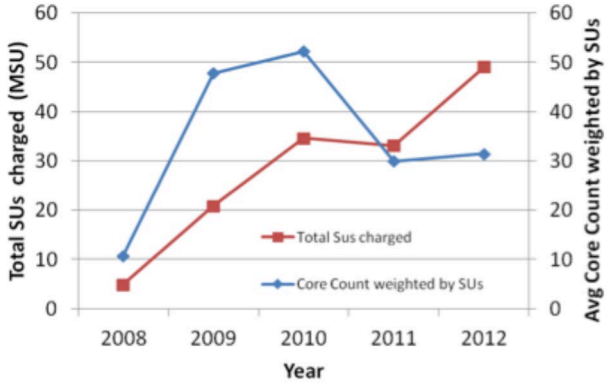

Figure 8 Steele Usage: Total SUs and Average Core Count Weighted by SUs.

on an idle system under preferred conditions and not as user in a normal production operation scenario and therefore do not necessarily reflect the performance that a user would experience. Modern HPC infrastructure is a complex combination of hardware and software environments that is continuously evolving, so it is difficult at any one time to know if optimal performance of the infrastructure is being realized. Indeed, as the examples below illustrate, it is more likely than not that performance is less than optimal, resulting in diminished productivity (CPU cycles, failed jobs) on systems that are typically over subscribed. Accordingly, the key to a successful and robust science and engineering-based HPC technology audit capability lies in the development of a diverse set of computationally lightweight application kernels that will run continuously on HPC resources to monitor and measure system performance, including critical components such as the global filesystem performance, local processor and memory performance, and network latency and bandwidth. The application kernels are designed to address this deficiency, and to do so from the perspective of the end-user applications.

We use the term ”Kernel” in this case to represent micro and standard benchmarks that represent key performance features of modern scientific and engineering applications, as well as small but representative calculations done with popular open-source high-performance scientific and engineering software packages. Details can be found in Reference [14]. We have distilled

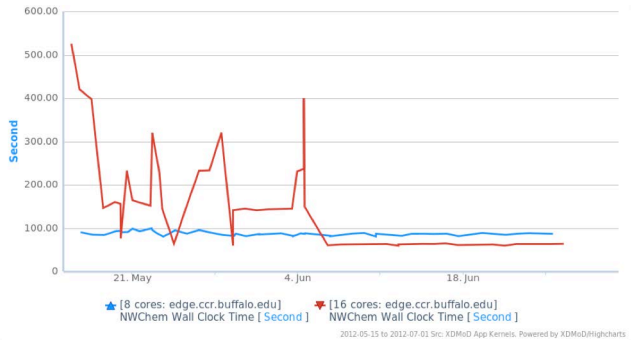

Figure 9 Plot of execution time of NWChem application kernel on 8 cores (blue line) and 16 cores (red line) over a several month time period. Calculations on 16 cores show wildly sporadic performance degradation until early June when a patch to a bug in a parallel file system was installed.

lightweight benchmarking kernels from widely used open source scientific applications that are designed to run quickly with an initially targeted wall-clock time of less than 10 minutes. However we also anticipate a need for more demanding kernels in order to stress larger computing systems subject to the needs of HPC resource providers to conduct more extensive testing. While a single application kernel will not simultaneously test all of these aspects of machine performance, the full suite of kernels will stress all of the important performance-limiting subsystems and components. Crucial to the success of the application kernel testing strategy, is the inclusion of historical test data within the XDMoD system. With this capability, site administrators can easily monitor the results of application kernel runs for troubleshooting performance issues at their site. Indeed, as the cases below illustrate, early implementation of application kernels have already proven invaluable in identifying underperforming and sporadically failing infrastructure that would have likely gone unnoticed, resulting in wasted CPU cycles on machines that are already oversubscribed as well as frustrated end users.

While the majority of the cases presented here are the result of the application kernels run on the large production cluster at the Center for Computational Research (CCR) at the University at Buffalo, SUNY, the suite of application kernels is currently running on most XSEDE resources and will soon be running on all XSEDE resources as part of the Technology Audit Service of XSEDE. Application Kernels have already successfully detected runtime errors on popular codes that are frequently run on XSEDE resources. For example, Figure 9 shows the execution time over the course of two months for an application kernel based on NWChem [21], a widely used quantum chemistry program, that is run daily on the large production cluster at CCR. While the behavior for 8 cores is as expected, calculations on 16 cores in May showed wildly sporadic behavior, with some jobs failing out right and others taking as much as seven times longer to run. The source of performance degradation was eventually traced to a software bug in the I/O stack of a commercial parallel file system, which was subsequently fixed by the vendor, as evidenced by the normal behavior in the application kernel after June 4th. Indeed, the software patch to fix this problem is now part of the vendor’s standard operating system release. It is important to note that this error was likely going on unnoticed by the administrators and user community for sometime and was only uncovered as a result of the suite of application kernels run at CCR.

As a further indication of the utility of application kernels, consider Figure 10, which shows a performance increase of a factor of two in MPI Tile IO after a system wide library upgrade from Intel MPI 4.0 to Intel MPI 4.0.3, which supports Panasas file system MPI I/O file hints. Since CCR employs a Panasas file system for its scratch file system, this particular application kernel alerted center staff to rebuild scientific applications that can utilize MPI file hints to improve performance. This would have gone unnoticed without the performance monitoring provided by the application kernels.

Figure 11, shows a sudden decrease in file system performance on Lonestar4 as measured by 3 different application kernels (IOR, MPI-Tile-IO, and IMB). The IOR and MPI-Tile-IO both show a sudden decrease in the aggregate write throughput bandwidth, while IMB, which measures latency, shows an equally sudden increase in latency. Once again, without application kernels periodically surveying this space, the loss in performance would have gone unnoticed. We are currently working with TACC to understand this file system performance deterioration.

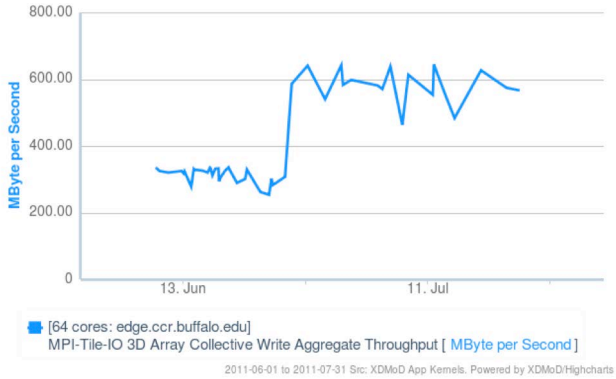

Figure 10 Application kernels detect I/O performance increase of a factor of 2 for MPI Tile IO in a library upgrade from Intel MPI 4.0 to Intel MPI 4.0.3, which supported PanFS MPI I/O file hints (in this case for concurrent writes).

One of the most problematic scenarios entails a single node posing a critical slowdown in which the cumulative resources for a job (possibly running on thousands of processing elements) are practically idled due to an unexpected load imbalance. It is very difficult for system support personnel to preemptively catch such problems, with the result that the end-users are the ”canaries” that

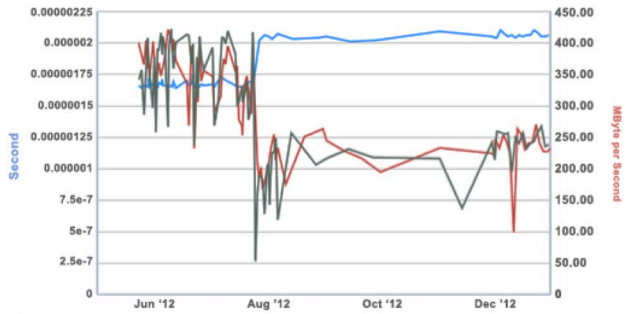

Figure 11 Application kernel data for IMB (blue), IOR (red) and MPI-Tile-IO (black) on Lonestar4. The IOR and MPI-Tile-IO data show a sudden drop of aggregate write throughput bandwidth and the IMB data shows a sudden increase in latency starting on 7/24-25/2012.

report damaged or underperforming resources, often after investigations that are very expensive both in terms of computational resources and personnel time. An active monitoring capability designed to automatically detect such problems is

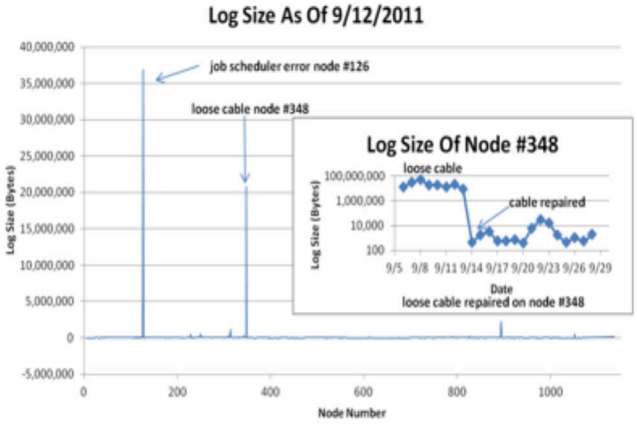

Figure 12 Plot of log file analysis for each node in CCR's production cluster. Two nodes produce very large log files. One node was found to have a loose cable and the other a job scheduler error, both resulting in failed jobs.

therefore highly desirable. For example, Figure 12 shows the results of a log file analysis of CCR’s large production cluster consisting of more than 1000 nodes. By examining only the size of the log files generated on each node (large log file size is indicative of errors) we were able to detect a loose cable on one node and a job scheduler error on another node, both of which resulted in failed jobs. Without such analysis, the loose cable or job scheduler error would have likely gone undetected, resulting

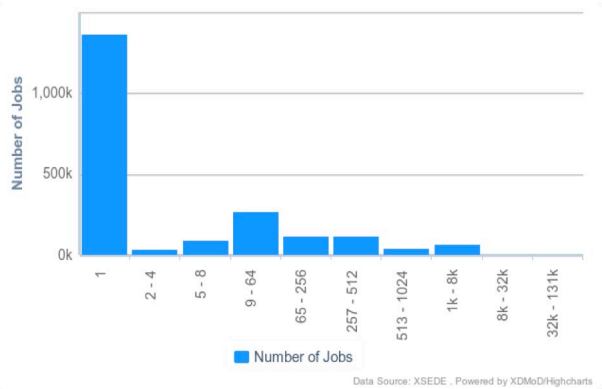

Figure 13 Distribution of job sizes for all parent science Physics jobs in TeraGrid/XSEDE resources for the period 2008-2012.

in many failed jobs, frustrated users, and underperformance of the resource. While analysis of system log files is not currently included within the XDMoD framework, it is anticipated that future versions will, given its utility in identifying faulty hardware.

3.3 Interpreting XDMoD Data

While XDMoD provides the user with access to extensive usage data for TeraGrid/XSEDE, like most analysis tools, care must be exercised in the interpretation of the generated data. This will be especially true for XDMoD given its open nature, the ease at which plots can be created, and the subtleties in the usage data that can require a fairly detailed understanding of the operation of TeraGrid/XSEDE [18], [19]. This is perhaps best understood through the following examples. Consider, for example, the mean core count across Physics parent science jobs on XSEDE resources during the period 2008-2012, which can be misleading given the distribution of job sizes as shown in Figure 13. The distribution of jobs is highly skewed by the presence of large numbers of serial (single-core) calculations, a situation exacerbated by recent ”high throughput” computing resources, as we will show.

One should not be misled into thinking that the overall resources are dominated by serial or small parallel jobs, a significant fraction are still ”capability” calculations requiring thousands of cores, as shown in Figure 14, which shows the breakdown of core count by quartile. While 75% of the jobs are for core counts of 100 or fewer processors, 25% of the jobs utilize very large core counts (thousands to tens of thousands).

We can elaborate further on this point by considering the mean core count on XSEDE resources for the field of physics (considered as a parent science within the scope of the XSEDE allocations). Figure 15 is a plot of mean job size (core count) from 2008-2012, showing both the naive mean calculated with all jobs as well as the mean of all parallel jobs. Note the divergence in the

mean calculated for all jobs vs all parallel jobs that occurs in 2010. The mean job size in this case is highly skewed by a rapid increase in the number of single core jobs. XDMoD can be used to identify this contribution of serial calculations, and as can be seen in Figure 16, the dramatic increase in serial jobs comes from several physics allocations ramping up on the high-throughput

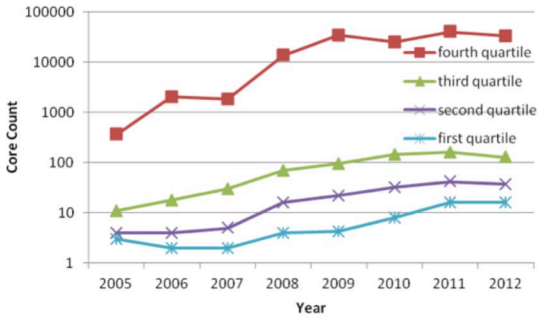

Figure 14 Average core count for all XSEDE resources broken out in quartiles, showing a significant fraction of very large core count jobs.

resources at Purdue during 2010-2012.

XDMoD puts a trove of data in the hands of the public and policy makers in a relatively easy to use interface. This data has to be used in the proper context, however, as it can be too easy to rapidly draw misleading conclusions. Based solely on the mean job size for all jobs in Figure 15, one might be tempted to wonder why the Physics allocations started using fewer cores on average in the latter half of 2010 - the answer is that they did not, rather an

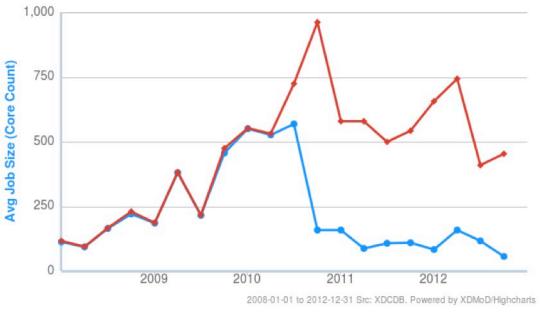

Figure 15 Mean core count for Physics jobs in TeraGrid/XSEDE resources for the period 2008-2012, including (blue circles) and excluding (red squares) serial runs.

enterprising subgroup of them started exploiting high throughput systems on an unprecedented scale (for TeraGrid/XSEDE).

4. CONCLUSIONS AND FUTURE WORK

We have demonstrated, through several case studies, the utility of XDMoD as a tool for providing metrics regarding both resource utilization and performance of advanced cyberinfrastructure, primarily TeraGrid/XSEDE. The XDMOD platform already enables systematic data driven understanding of the current and historical usage and planning for future usage. We believe that this will lead to more appropriate resource management and resource planning. Users will also benefit from the availability of relevant benchmark performance data for their applications from the kernels performance. Managers of CI resources will, through XDMoD, be able to more readily identify underperforming hardware, all to the benefit of the end user. Furthermore, as additional data is captured and ingested it will also allow more outcome centric measures of return on the national cyberinfrastructure investment.

XDMoD's implementation of an application kernel-based auditing system that utilizes performance kernels to measure overall system performance was shown to provide a useful means to detect under performing hardware and software. Examples included an application kernel based on a widely used quantum chemistry program that uncovered a software bug in the I/O stack of a commercial parallel file system, which was subsequently fixed by the vendor in the form of a software patch that is now part of their standard release. This error, which resulted in dramatically increased execution times as well as outright job failure, would likely have gone unnoticed for sometime and was only uncovered as a result of implementation of a suite of application kernels. Application kernels also detected a performance increase of a factor of two in MPI Tile IO after a system wide library upgrade from Intel MPI 4.0 to Intel MPI 4.0.3, alerting center staff to rebuild those applications which utilize MPI I/O file hints to improve performance. IO application kernels were also able to detect a deterioration of performance in Lonestar4's file system write throughput capacity. Many of the more straight-forward usage metrics have already been incorporated into XDMoD, however it should still be viewed as a work in progress.

There are a number of features currently being added to enhance the capabilities of XDMoD. One example is the addition of TACC Stats data to XDMoD. TACC Stats records hardware performance counter values, parallel file-system metrics, and high-speed interconnect usage [22]. The core component is a collector executed on all compute nodes, both at the beginning and end of each job. With the addition of application script recording, this will provide a fine grained job level performance not currently available for HPC systems. Another example is the addition of the PEAK (Performance Environment Autoconfiguration framework) to automatically help developers, system administrators, and users of scientific applications select the optimal configuration for their application on a given platform and to update that configuration when changes in the underlying hardware and systems software occur [23]. The configuration options considered for the performance optimization include the compiler with its settings of compiling options, the numerical libraries and settings of library parameters, and settings of other environment variables. The PEAK framework has been demonstrated to select the optimal configuration to achieve significant speedup for scientific applications executed on XSEDE platforms such as Kraken and Nautilus. In a different direction but just as important, we are in the process of adding metrics to assess scientific impact. While judging scientific impact is difficult it is nonetheless important to quantify in order to demonstrate the return on investment for HPC facilities. We plan on adding publications, citations, external funding and other metrics to establish the contribution that facilities such as XSEDE have on science in the U.S.

In addition we are engaging with the NSF XD FutureGrid (FG) project to bring forward a plan of how to integrate Cloud resources. At this time XD FutureGrid's data is available through its own portal. In contrast to other efforts, FG has provided an

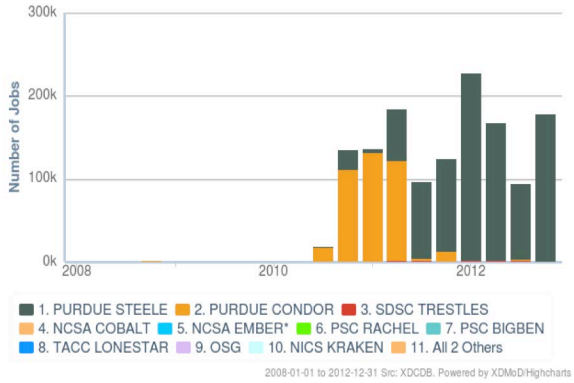

Figure 16 Number of serial (1 core) jobs by resource for the parent science of physics.

integrated monitoring solution for multiple Clouds including Nimbus, Eucalyptus, and Openstack [24]. It is important to note that the data collected for clouds and their metrics is technically significantly different from typical HPC data. Hence, it must be dealt with through different mechanisms. Currently, FG is added to the internal XSEDE backend databases as a special resource. We intend to evaluate how to best integrate the FG's Cloud-based information within the XDMoD framework.

ACKNOWLEDGEMENTS

This work was sponsored by NSF under grant number OCI 1025159 for the development of technology audit service for XSEDE.

5. REFERENCES

[1] Nagios: The Industry Standard. IT Infrastructure Monitoring: (Available from: http://www.nagios.org/[August 19, 2011]).

[2] Matthew, L., Massie, B., Chun, N., Culler, D. E., The ganglia distributed monitoring system: design, implementation, and experience. Parallel Computing 2004; 30(8): 817–840.

[3] Cacti: The Complete RRDTool-based Graphing Solution. (Available from: http://www.cacti.net/ [August 19, 2011]).

[4] Smallen, S., Olschanowsky, C., Ericson, K., Beckman, P., Schopf, J., The Inca test harness and reporting framework. Proceedings of Supercomputing, Pittsburg PA, 2004; 55–64. See also: Inca. http://inca.sdsc.edu [September 1, 2011].

[5] Hawkeye: A Monitoring and Management Tool for Distributed Systems. (Available from: http://www.cs.wisc.edu/condor/hawkeye/ [August 19, 2011]).

[6] von Laszewski, G., J. DiCarlo, and B. Allcock, "A Portal for Visualizing Grid Usage," Concurrency and Computation: Practice and Experience, vol. 19, iss. 12, pp. 1683-1692, [2007].

[7] Martin, S., Lane, P., Foster, I., Christie, M.. TeraGrid's GRAM Auditing & Accounting, & Its Integration with the LEAD Science Gateway, TeraGrid Workshop, 2007. (Available from: http://www.globus.org/alliance/publications/papers/TG_GRAM_auditing_and_LEAD_Gateway_final_2.pdf[August 19, 2011]).

[8] Canal, P., Green, C., GRATIA, a resource accounting system for OSG. CHEP, Victoria, B.C., 2007.

[9] DOD HPC modernization program metrics. (Available from: http://www.hpcmo.hpc.mil/Htdocs/HPCMETRIC/index.html[Dec16, 2011]).

[10] Bennett, P.M., Sustained systems performance monitoring at the U. S. Department of Defense high performance computing modernization program. In State of the Practice Reports (SC '11), Article 3. ACM: New York, NY, USA, 2011; 11 pages, DOI: 10.1145/2063348.2063352. http://doi.acm.org/10.1145/2063348.2063352.

[11] NERSC performance monitoring tools. (Available from: https://www.nersc.gov/research-and-development/performance-and-monitoring-tools/ [December 16, 2011]).

[12] DOE "operational assessment" metrics for various HPC sites, for example ORNL. http://info.ornl.gov/sites/publications/files/Pub32006.pdf[December 16, 2011]).

[13] University at Buffalo Metrics on Demand (UBMoD): Open source web portal for mining data from resource managers in HPC environments. Developed at the Center for Computational Research at the University at Buffalo, SUNY. Freely available at SourceForge at http://ubmod.sourceforge.net/[May 1, 2012].

[14] Furlani, T.R., Jones, M.D., Gallo, S.M., Bruno, A.E., Lu, C.-D., Ghadersohi, A., Gentner, R.J., Patra, A., DeLeon, R.L., von Laszewski, G., Wang, F., and Zimmerman, A., "Performance metrics and auditing framework using application kernels for high performance computer systems," Concurrency and Computation: Practice and Experience, vol. 25, pp.918-931, 2013. [Online]. Available: http://dx.doi.org/10.1002/cpe.2871

[15] University at Buffalo, "XDMoD portal." [Online]. Available: https://xdmod.ccr.buffalo.edu

[16] Katz, D.S., Hart, D., Jordan, C., Majumdar, A., Navarro, J.P., Smith, W., Towns, J., Welch, V., and Wilkins-Diehr, N., "Cyberinfrastructure usage modalities on the TeraGrid," in Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum, ser. IPDPSW '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 932–939. [Online]. Available: http://dx.doi.org/10.1109/IPDPS.2011.239

[17] "XSEDE Overview." [Online]. Available: https://www.xsede.org/overview

[18] Hart, D. L., "Measuring TeraGrid: workload characterization for a high-performance computing federation," International Journal of High Performance Computing Applications, vol. 25, no. 4, pp. 451–465, 2011. [Online]. Available: http://hpc.sagepub.com/content/25/4/451.abstract

[19] Hart, D., "Deep and wide metrics for hpc resource capability and project usage," in State of the Practice Reports, ser. SC '11. New York, NY, USA: ACM, 2011, pp. 1:1–1:7. [Online]. Available: http://doi.acm.org/10.1145/2063348.2063350

[20] Bennett, P.M., "Sustained systems performance monitoring at the U.S. Department of Defense High Performance Computing Modernization Program," in State of the Practice Reports, ser. SC '11. New York, NY, USA: ACM, 2011, pp. 3:1–3:11. [Online]. Available: http://doi.acm.org/10.1145/2063348.2063352

[21] Valiev, M., Bylaska, E., Govind, N., Kowalski, K., Straatsma, T., Dam, H.V., Wang, D., Nieplocha, J., Apra, E., Windus, T., and de Jong, W., "NWchem: A comprehensive and scalable open-source solution for large scale molecular simulations," Computer Physics Communications, vol. 181, no. 9, pp. 1477 – 1489, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010465510001438

[22] Hammond, J., "TACC Stats I/O performance monitoring for the intransigent," 2011, in 2011 Workshop for Interfaces and Architectures for Scientific Data Storage, IASDS 2011. Available: http://www.mcs.anl.gov/events/workshops/iasds11/presentations/jhammond-iasds.pdf

[23] Hadri, B., You, H., Moore, S., "Achieve better performance with PEAK on XSEDE resources", XSEDE '12 Proceedings of the 1st Conference of the XSEDE: Bridging from the eXtreme to the campus and beyond, Article 10 (2012): DOI =10.1145/2335755.2335801

[24] von Laszewski, G., Lee, H., Diaz, J., Wang, F., Tanaka, K., Karavinkoppa, S., Fox, G. C., and Furlani, T. , "Design of an Accounting and Metric-based Cloud-shifting and Cloud-seeding framework for Federated Clouds and Bare-metal Environments," , San Jose, CA., [September, 2012].