Data Analytics Driven Cyberinfrastructure Operations, Planning and Analysis Using XDMoD

Thomas R. Furlani*, Barry I. Schneider† Matthew D. Jones*, John Towns‡, David L. Hart§, Abani K. Patra*,
Robert L. DeLeon*, Steven M. Gallo*, Charng-Da Lu*, Amin Ghadersohi*, Ryan J. Gentner*,
Andrew E. Bruno*, John R. Boisseau¶, Fugang Wang‖, and Gregor von Laszewski‖
*Center for Computational Research,
University at Buffalo, State University of New York,
701 Ellicott Street, Buffalo NY 14203
Email:{furlani, jonesm, smgallo, charngda, ag28, rgentner, abani, rldeleon, aebruno2}@buffalo.edu
†Office of Cyberinfrastructure,
National Science Foundation,
4201 Wilson Boulevard, Arlington VA 22230
Email: bschneid@nsf.gov
‡National Center for Supercomputing Applications,
University of Illinois at Urbana-Champaign,
1205 W. Clark St., Urbana IL 61801
Email: jtowns@ncsa.illinois.edu
§National Center for Atmospheric Research,
P.O. Box 3000, Boulder CO 80307-3000
Email: dhart@ucar.edu
¶Texas Advanced Computing Center,
Research Office Complex 1.101,
J.J. Pickle Research Campus, Building 196,
10100 Burnet Road (R8700), Austin TX 78758-4497
Email: boisseau@tacc.utexas.edu
‖Pervasive Technology Institute,
Indiana University,
2719 East 10th Street, Bloomington IN 47408
Email:{kevinwangfg, laszewski}@gmail.com

Abstract—**Historically high-end cyberinfrastructure planning at the campus, regional and national levels has been based on episodic analysis of limited data and/or projections of demand with minimal supporting comprehensive usage data. However, a repository of usage data for the TeraGrid and the follow-on XSEDE program provides a unique source of extensive data that can be exploited to guide planning efforts. The XDMoD tool deployed by the Technology Audit Service (TAS) component of XSEDE is designed to facilitate access to these data by multiple stakeholder groups, providing a unique opportunity to carry out comprehensive analysis of cyberinfrastructure usage. To complement usage data and strengthen XDMoD's utility for overall system analysis, a suite of application kernels has been developed to help provide control data on system performance. Current and past utilization metrics, coupled with application kernel-based performance analysis, can be used to help guide future cyberinfrastructure investment decisions, plan system upgrades, tune machine performance, improve user job throughput, and facilitate routine system operation and maintenance. In this paper we present analysis of historical usage data from the TeraGrid and the follow-on XSEDE program and derive interesting insight into the nature of usage by discipline over time. The analysis shows the remarkable growth in resources and the impact this has had on the number of users, the number and size of allocations, the job size in terms of number of cores, and the growth in simulation based engineering and science in many fields. The utility of the XDMoD framework for facilitating system performance assessment through the implementation of application kernels is also demonstrated.**

I. INTRODUCTION

Planning of high-end cyberinfrastructure (CI) can be done best when it is based upon reliable, extensive data from past usage. In addition, as described by Katz et. al. [1], 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. The National Science Foundation (NSF) recognized the value of this data and through the Technology Audit Service (TAS) of XSEDE made a significant investment in developing tools and infrastructure to make this sort of data and data analysis easily accessible to a broader range of users and resource managers. In this context, the XDMoD (**XSEDE** Metrics on Demand) auditing tool provides an extensive range of metrics that gives users and XSEDE management the ability to rapidly access historical data broken out by various categories such as resource, user and field of science. While XDMoD has made reporting a much simpler and less time-consuming task, the range of metrics available has also provided insight into the operation of TeraGrid/XSEDE that was not readily available, and in some cases not even possible previously. Thus XDMoD augments qualitative past methods, such as surveys, one-on-one interviews with users, and workshop reports, aimed at better understanding and improving service to TeraGrid/XSEDE users [2], [3], [4], [5].

Usage data is dependent on the idiosyncrasies of user/vendor implementation of algorithms – a significant drawback in analyzing systems performance based on usage data. To alleviate this, the TAS XDMoD framework was expanded to also include an auditing system that utilizes computationally lightweight application kernels to provide a measure of overall system performance. These kernels, reminiscent of the “Berkeley dwarfs” [6], are representative of the major computational applications that are routinely run on HPC resources. With the right selection of kernels, this allows continuous resource auditing to measure all aspects of system performance including file-system performance, processor and memory performance, and network latency and bandwidth. The deployment of these application kernels, which are standardized programs run on XSEDE resources exactly as a typical user runs them, can give the user insight on potential performance for their application and provide resource managers with information on how applications are performing on their systems. Routine use of such kernels also provides resource managers early warning of anomalous systems behavior.

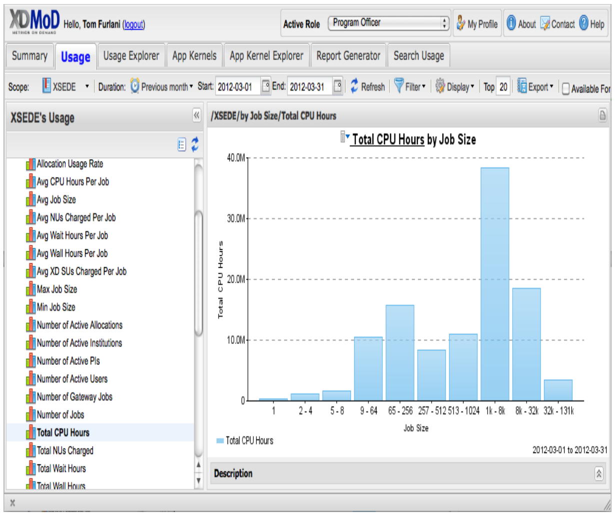
The remainder of this paper is organized as follows. We first provide a brief overview of XDMoD to provide context for the discussion that follows. We then present the results of several XDMoD usage case studies, beginning with an analysis of historical usage data from the TeraGrid and the follow-on 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.

II. XDMoD OVERVIEW

XDMoD provides a role-based web portal for mining HPC system usage data and performing statistical analysis. This role-based framework is designed 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 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. Here we present a brief overview of XDMoD's functionality, a more detailed description is contained in the appendix and an earlier publication [7].

In the present implementation of XDMoD, data is ingested daily from the XSEDE central data base (XDCDB) and queries are tuned to provide results within a few seconds. To date, the focus has been on XSEDE's compute resources, though future versions will be extended to include XSEDE's visualization and storage resources. Furthermore, while XDMoD is currently tailored to work with the data stored in the XDCDB, future releases will allow for custom databases containing similar data collected by individual HPC centers. In the meantime, the open source package UB Metrics on Demand (UBMoD) is available to provide useful utilization metrics for academic HPC centers [8].

The XDMoD portal [9] provides a rich set of features accessible through an intuitive graphical interface, which is tailored to the role of the user. Currently five roles are supported: Public, User, Principal Investigator, 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 2011 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. Metrics that focus on scientific impact, such as publications, citations and external funding, will be incorporated in future versions to help quantify the important role HPC centers play in advancing research and scholarship.


Hello, Tom Furlani (logout)

Active Role: Program Officer

Scope: XSEDE Duration: Previous month Start: 2012-03-01 End: 2012-03-31 Top 20

Fig. 1. The XDMoD interface. For the Program Officer role, seven tabs near the top of the screen allow for navigation around the site to access the various metrics. The Plot shows the total CPU hours provided by all XSEDE resources in 2011 by job size.

III. XDMoD USAGE CASE STUDIES

A. A Data History of TeraGrid/XSEDE Usage: Defining a Strategy for Advanced Cyberinfrastructure

XSEDE is the most advanced, powerful, and robust collection of integrated advanced digital resources and services in the world [10]. 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 understood 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 2011.

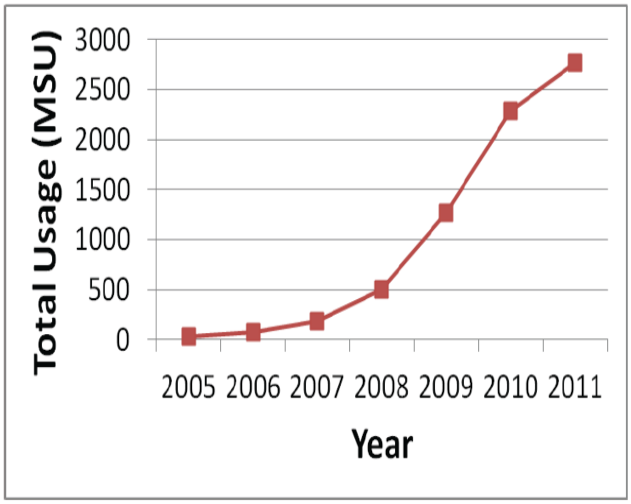


Fig. 2. Total XSEDE usage in millions of service units (SUs) for the years 2005-2011. Note: for the purpose of this paper, service units should be understood as core hours with the caveat that the value of an SU varies across resources utilizing varying technologies and, by implication, varies over time as technology advances.

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. Thus, the largest allocation of a single user today exceeds the total usage of all users in 2005 and 2006.

Not surprisingly, over the same time period there has been a substantial increase in the number of users as shown in Figure 4. The number of allocations today, which can roughly be thought of as the number of PIs, is around 1500 with on average two users per allocation. Thus, not all registered users are always active. It is often the case that a project PI does not actually use SUs, with the hands-on tasks going to graduate students and postdocs. In fact, historical analysis of TeraGrid/XSEDE data shows that only a third of registered users regularly appear in the usage data, that is, this situation is not abnormal but rather the status quo.

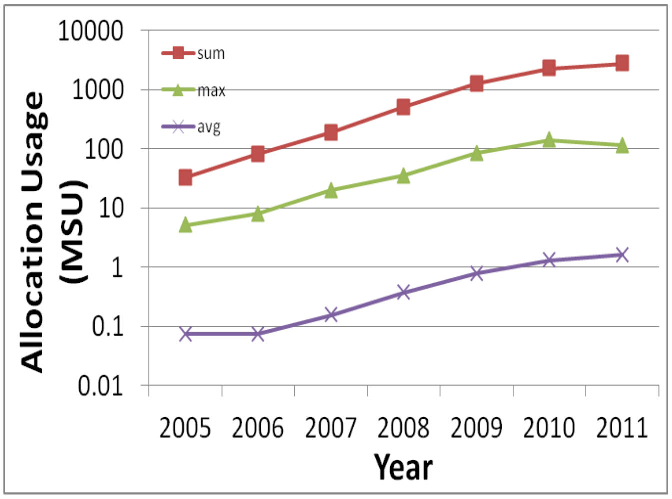


Fig. 3. Largest, average and total SU usage over time. Note that the average and largest allocations have increased by more than an order of magnitude over the time period shown.

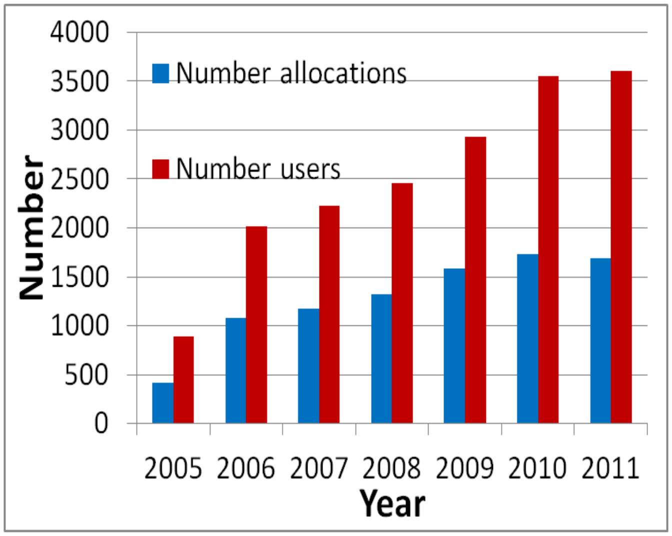


Fig. 4. Number of TeraGrid/XSEDE allocations and users. The number of allocations is roughly analogous to the number of PIs and there are approximately 2 users per allocation.

Note that in spite of the increase in computational resources, Figure 5 indicates that requests are growing but that the success rate, which is defined here as the ratio of allocation requested to the allocation actually awarded, has dropped from a high of about 75%, when the large Track 2 resources became available, to 40% currently. The success rate shown in Figure 5 is based on the fact that while few proposals are rejected most proposals are granted substantially less than the requested number of SUs. This clearly demonstrates that the number of requests for SUs substantially outstrips currently available resources.

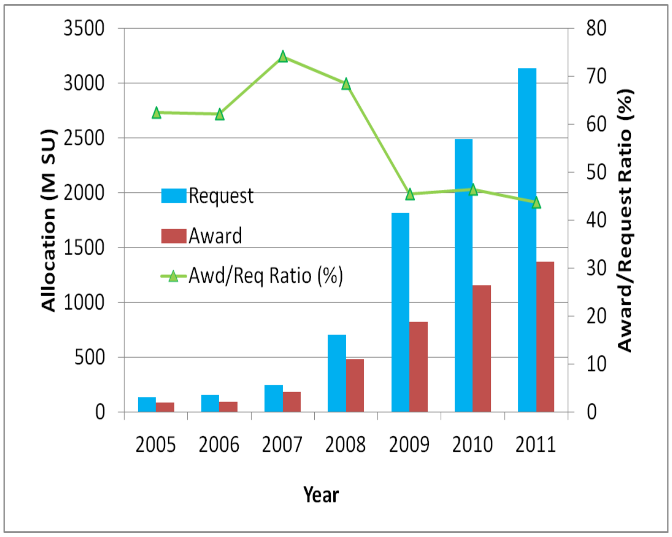


Fig. 5. Allocation Success Rate. The success rate is defined here as the ratio of the allocation request to the allocation awarded.

The TeraGrid/XSEDE usage by parent science is shown in Figure 6. 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 6 and Figure 7 can easily be related to current NSF dirvisions. 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.

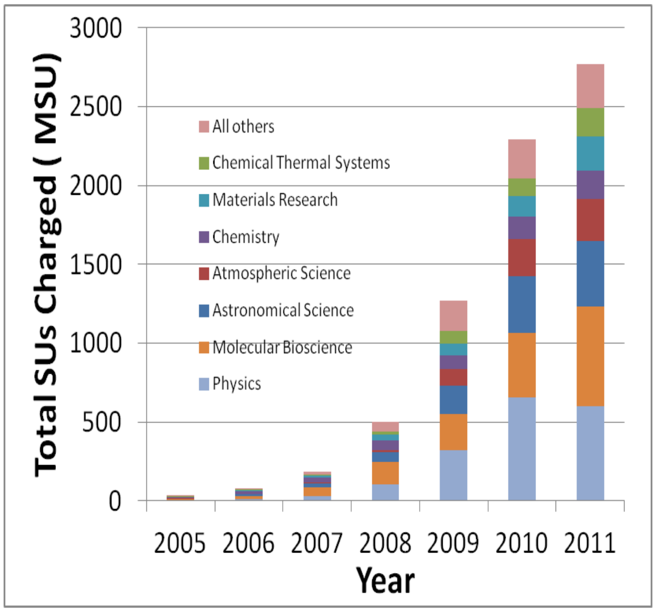


Fig. 6. Total SU Usage by Parent Science

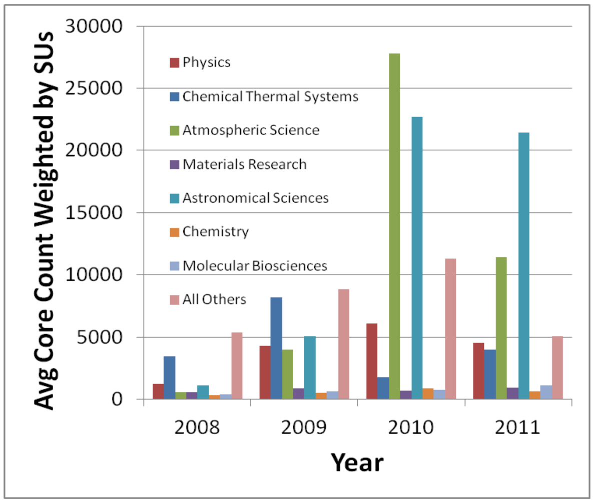


Fig. 7. Average Core Count (Weighted by SUs) by Parent Science

However, as Figure 7 shows, the average core count by parent science varies widely. Note, as shown in [11], [12] and Section III-C 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) directorate of NFS have been thought to be the largest users of XSEDE computational resources. While MPS users are still significant, it is clear from Figure 6 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 7 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.

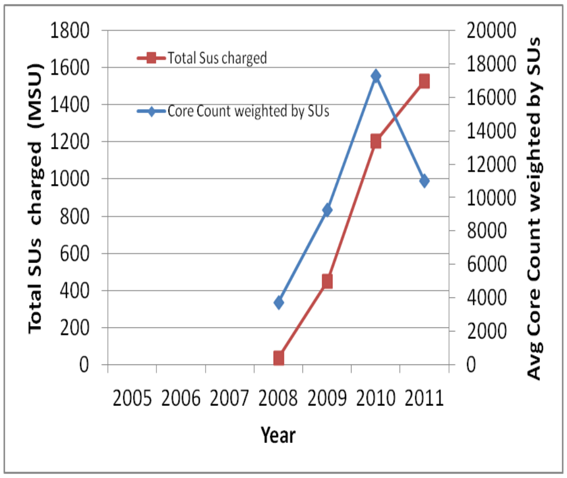


Fig. 8. Kraken Usage:Total SUs and Average Core Count Weighted by SUs

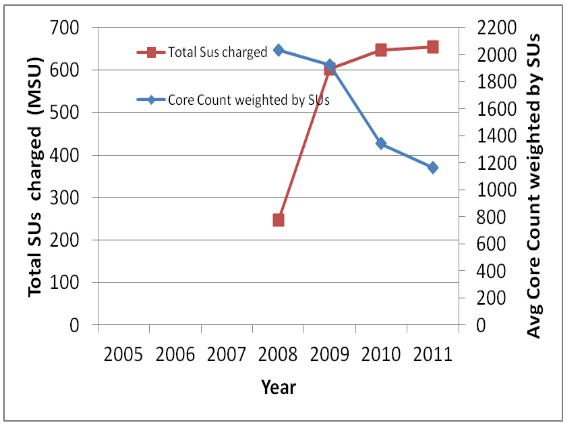


Fig. 9. Ranger Usage:Total SUs and Average Core Count Weighted by SUs

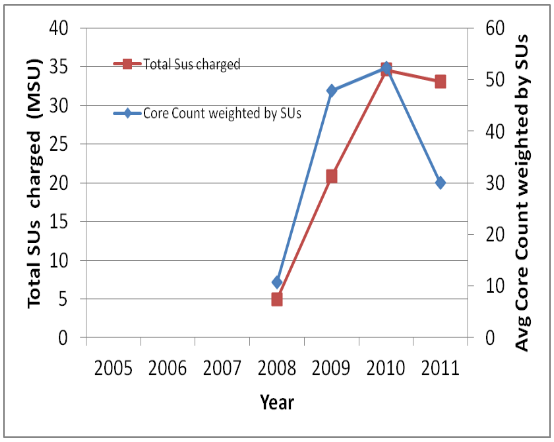


Fig. 10. Steele Usage:Total SUs and Average Core Count Weighted by SUs

In this section, three of the XSEDE resources, namely Kraken, Ranger, 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. This is not an accident. 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 8 to 10 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 Ranger are ideal, and this is reflected in the average core count. In the future, Stampede and Blue Waters will likely be the systems of choice for such users. Lonestar, a very recent addition to the portfolio (not shown since the data is limited), is a smaller resource in terms of core count 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 Lonestar (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. The PSC system, Blacklight, (also 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 science gateways, campus champions and advanced user support programs.

An interesting observation looking at Figures 8 to 10, 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.

B. 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 [13] 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 on an idle system under preferred conditions and not a 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 optimal performance across all applications is not being realized. 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 [7]. We have distilled 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 some XSEDE resources and will soon be running on all XSEDE resources as part of the Technology Audit Service of XSEDE.

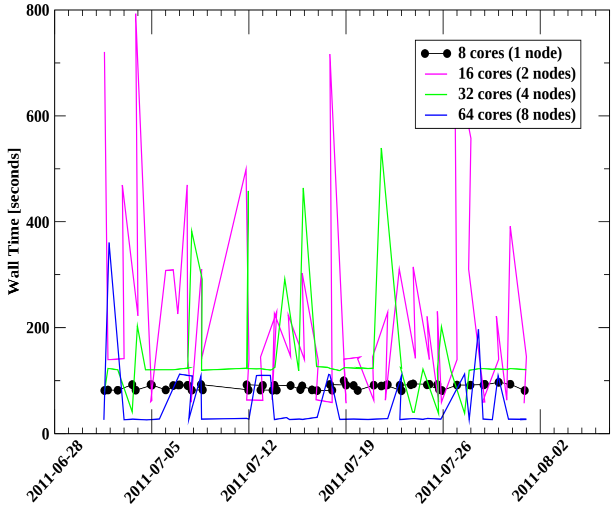


Fig. 11. Plot of execution time of NWChem application kernel on 8, 16, 32, and 64 processors over a one month time period on CCR’s production cluster. The execution time for increasing core count should be consistently improving. The behavior on 8 cores is as expected but 16, 32 and 64 core calculations show sporadic performance degradation, later traced to a bug in the global parallel filesystem.

Application Kernels have already successfully detected runtime errors on popular codes that are frequently run on XSEDE resources. For example, Figure 11 shows the execution time over the course of a month for an application kernel based on NWChem [14], 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, 32, and 64 cores showed widely 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 in the form of a software patch that is now part of their 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.

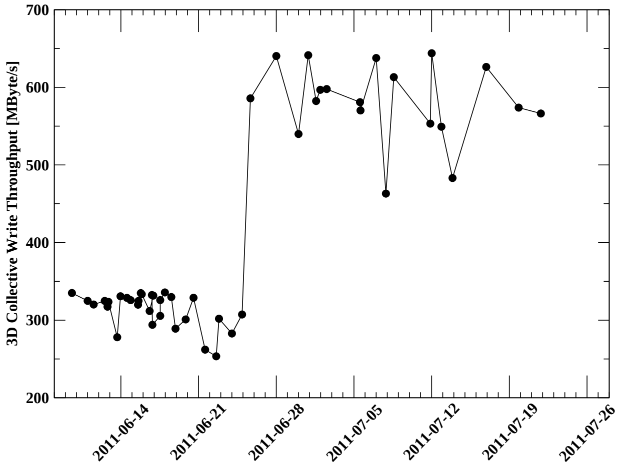


Fig. 12. Application kernels detect I/O performance increase of a factor of 2 for MPI Tile IO in library upgrade from Intel MPI 4.0 to Intel MPI 4.0.3, which supported PanFS MPI I/O file hints (in this case for concurrent writes). Provides an alert to center staff to rebuild scientific applications that utilize MPI I/O to improve performance.

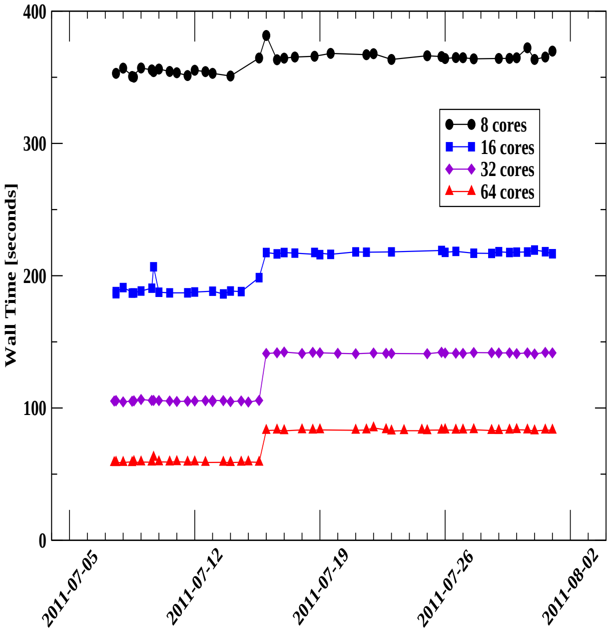


Fig. 13. Plot of execution time of NAMD based application kernel on 8, 16, 32, and 64 cores over a one month time period. Jump indicates when NAMD application underwent routine system-wide upgrade in the application version, which resulted in a performance degradation.

As a further indication of the utility of application kernels, consider Figure 12, 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.

Figure 13, shows the unanticipated results brought to light by a periodically running application kernel based on the popular NAMD molecular dynamics package [15]. The application kernel detected a 25% degradation in the NAMD baseline performance that was the unanticipated result of a routine system-wide upgrade of the application version. Possible strategies for restoring the pre-upgrade performance include the use of more aggressive compiler options, but care will need to be exercised to ensure the desired level of accuracy is maintained. Once again, without application kernels periodically surveying this space, the loss in performance would have gone unnoticed.

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 report damaged or underperforming resources, often after investigations that are very expensive both in terms of computational resources and personnel time. Indeed, as shown in Figure 14, a single loose cable on a cluster consisting of just 1000 nodes resulted in the sporadic failure of user jobs that was difficult to diagnose or even know that a problem existed. However, an analysis of system log files was able to readily identify the single faulty node which malfunctioned due to a loose cable. Without such a capability, the loose cable would have likely gone undetected, resulting in failed jobs, frustrated users, and under-performance 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.

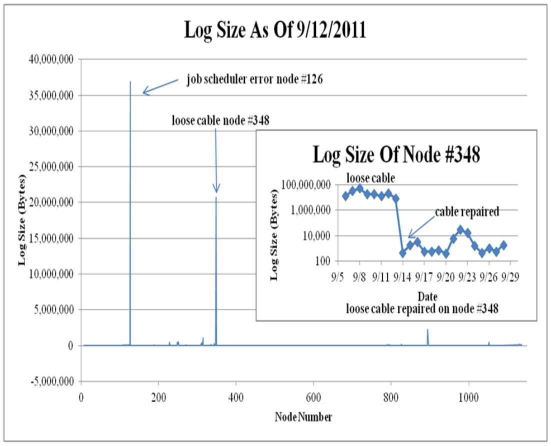


Fig. 14. Plot of log file size for each node in CCR's production cluster. Large log file size can be indicative of an error. In this case, two nodes produce log files that are 3 orders of magnitude larger than normal. One node was found to have a loose cable, causing sporadic errors (failed jobs) and the other had a error in the job scheduler, again resulting in failed jobs.

C. *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 [11], [12]. 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 2006-2011, which can be misleading given the distribution of job sizes as shown in Figure 15. 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.

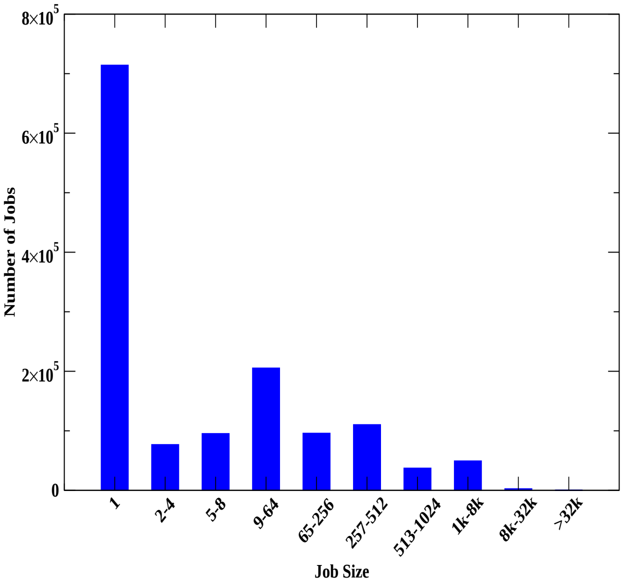


Fig. 15. Distribution of job sizes for all parent science Physics jobs in TeraGrid/XSEDE resources for the period 2006-2011.

One should also 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 7. The breakdown of core count by quartile is shown in Figure 16. 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 17 is a plot of mean job size (core count) from 2006-2011, showing both the naive mean calculated with all jobs as well as the mean of all *parallel* jobs. 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 18, the dramatic increase in serial jobs comes from several physics allocations ramping up on the high-throughput resources at Purdue during 2010-2011.

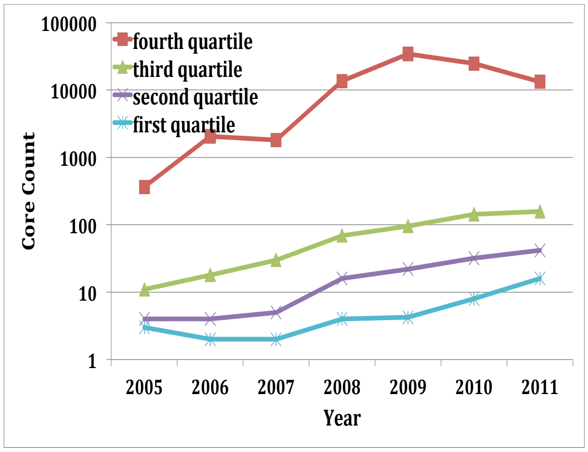


Fig. 16. Average core count for all XSEDE resources broken out in quartiles, showing a significant fraction of very large core count jobs.

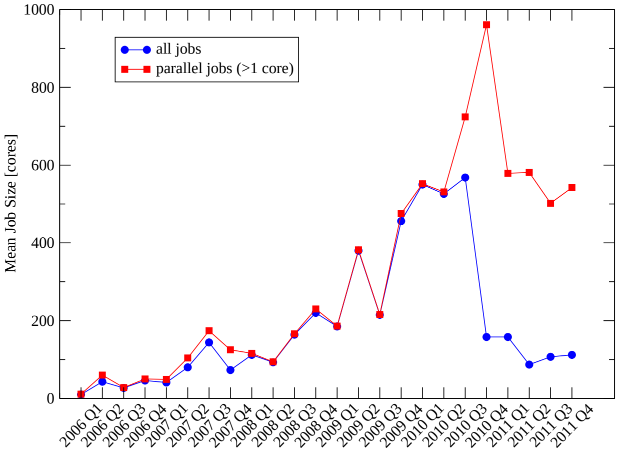


Fig. 17. Mean core count for Physics (parent science) jobs in Teragrid/XSEDE resources for the period 2006-2011, including (blue circles) and excluding (red squares) serial runs.

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 on the mean job size for all jobs in Figure 17, 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 enterprising subgroup of them started exploiting high throughput systems on an unprecedented scale (for TeraGrid/XSEDE at least).

IV. 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, including TeraGrid/XSEDE. The XDMOD platform already enables systematic data driven understanding of the

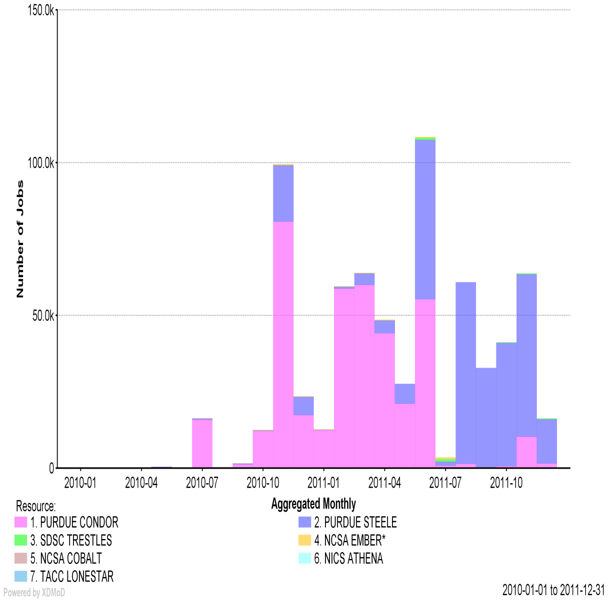


Fig. 18. Number of jobs by resource for the parent science of physics and job size of 1 core.

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. As additional data is captured and ingested it will also allow more outcome centric measures of return on the national cyberinfrastructure investment.

In the case of TeraGrid/XSEDE, a detailed historical analysis of usage data clearly demonstrates the tremendous growth in the number of users, overall usage, and scale of the simulations routinely carried out. For example, both the average allocation and largest allocation on TeraGrid/XSEDE have increased by more than an order of magnitude since 2005 to 1 million and 100 million SUs respectively. Not surprisingly, physics, chemistry, and the engineering disciplines were 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 2011. 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 planning future advanced cyberinfrastructure.

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. Since CCR employs a Panasas file system, a substantial performance gain can be realized by rebuilding scientific applications that can utilize MPI file hints.

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 this tool. 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 [16], [17]. 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. 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.

ACKNOWLEDGEMENTS

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

APPENDIX – XDMoD Architecture and Use

Figure 19 provides a high-level schematic of the XDMoD framework architecture. The system is comprised of three major components: the XDMoD Data Warehouse, which ingests data daily from the XSEDE Central Database, the XDMoD REST API that provides hooks to external applications, and the XDMoD Portal, which provides an interface to the world and is described as follows.

As shown in Figure 1, which is a screen capture of the XDMoD portal for the Program Officer role, the interface is organized by tabs with different functional tabs displayed for the XDMoD roles. For illustrative purposes, we will focus on the Program Officer role. The Summary tab provides a snapshot overview of XSEDE, with several small summary charts visible that can be expanded to full size charts through a simple mouse click. Clicking on the XSEDE button (in the row underneath the tab row) brings up a drop down menu that allows one to narrow the scope of the metrics displayed to a particular service provider. The default is to show utilization over the previous month, but the user may select from a range

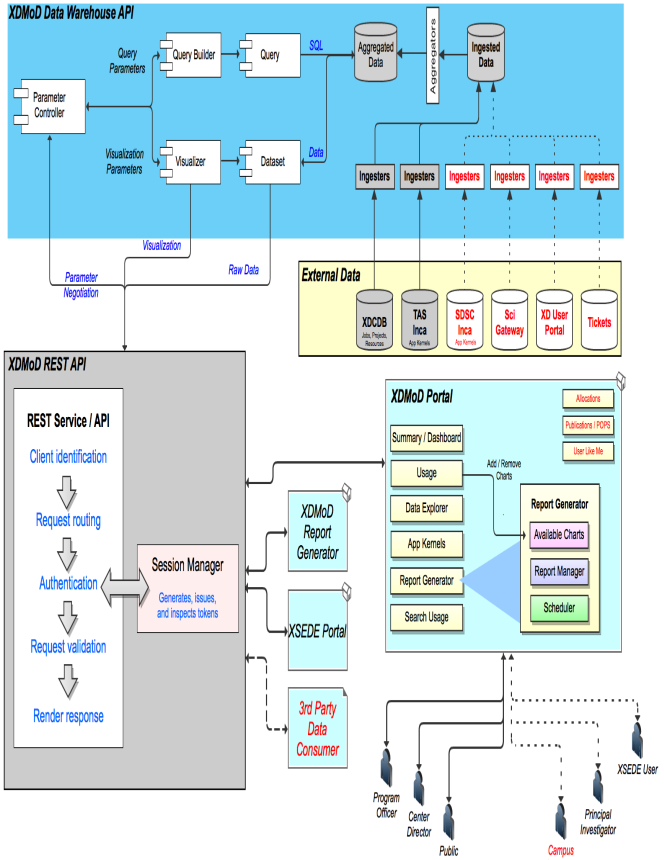


Fig. 19. The XDMoD architecture schematic

of preset date ranges (week, month, quarter, year to date, etc) or choose a custom date range. Clicking the Usage tab, as shown in Figure 1, provides access to an expansive set of XSEDE-wide metrics that are accessible through the tree-structure on the left-hand side of the portal window. If logged in under the User role or Center Director role as opposed to the Program Officer role, then the usage tab provides details specific to your utilization or that of your center as opposed to all of XSEDE. Accessing XDMoD through the Public role requires no password, and while not all the functionality listed above is available in this view, it does allow users to explore utilization and performance metrics of XSEDE resources over an adjustable timeframe.

The Usage Explorer tab provides a powerful tool for organizing and comparing the XSEDE data from a wide variety of metrics. The App Kernels tab provides information on the application kernel performance on XSEDE resources. The data generated by the application kernels is substantial, making the exploration of the data challenging. However, the App Kernel Explorer tab provides an interface that facilitates exploration of the application kernel performance metrics. Here the user can easily select a specific application kernel or suite of application kernels, a specific resource, and a range of job sizes for which to view performance. The Search Usage tab allows the Program Officer to view the utilization data for any XSEDE user.

The Report Generator tab gives the user access to the Custom Report Builder that allows a user to create and save custom reports. For example, a user may wish to have specific plots and data summarized in a concise report that they can download for offline viewing. The user can also choose to have custom reports generated at a user-specified interval (daily, weekly, quarterly, etc) and automatically delivered to them via email at the specified time interval, without the need to subsequently log into the portal.

Additional features include: the Export button that allows data to be output in a variety of formats (CSV, XML, PNG), the Filter button, which allows the user to select which data to display and which to supress in a given plot, and the Help button, which allows the user access to the XDMoD user guide. The Display button allows the user to customize the type and appearance of the chart and to toggle between the display of a given chart or the data set used in its generation and also to display time series (that is data plotted as a function of time).

References

[1] D. S. Katz, D. Hart, C. Jordan, A. Majumdar, J.P. Navarro, W. Smith, J. Towns, V. Welch, N. Wilkins-Diehr, "Cyberinfrastructure Usage Modalities on the TeraGrid,2011 High-Performance Grid and Cloud Computing Workshop," 2011 Proceedings of 2011 IPDPS Workshops, pp. 927-934, 2011, DOI:10.1109/IPDPS.2011.239.

[2] A. Zimmerman, and T. A. Finholt, Report from the TeraGrid Evaluation Study, Part 1: Project Findings. Collaboratory for Research on Electronic Work, School of Information, University of Michigan. 97 pp., 2008, http://hdl.handle.net/2027.42/61838.

[3] A. Zimmerman, M. Krause, K.A. Lawrence, and T. A. Finholt, Report from the TeraGrid Evaluation Study, Part 2: Findings from the TeraGrid User Survey. Collaboratory for Research on Electronic Work, School of Information, University of Michigan. 47 pp., 2008, http://hdl.handle.net/2027.42/61839.

[4] A. Zimmerman, Ann and T. A. Finholt, TeraGrid User Workshop Final Report. Collaboratory for Research on Electronic Work, School of Information, University of Michigan, 2006, http://hdl.handle.net/2027.42/61841.

[5] D. S. Katz, S. Callaghan, R. Harkness, S. Jha, K. Kurowski, S. Manos, S. Pamidighantam, M. Pierce, B. Plale, C. Song, J. Towns, Science on the TeraGrid, Computational Methods in Science and Technology (CMST), Special Issue, pp. 81-97, 2010.

[6] K. Asanovic, R. Bodik, B. Catanzaro, J. Gebis, P. Husbands, K. Keutzer, D. Patterson, W. Plishker, J. Shalf, S. Williams, K. Yelik. The Landscape of Parallel Computing Research: A View from Berkeley. *Technical report UCB/EECS-2006-183*, EECS Department, University of California at Berkeley, December 2006.

[7] T. R. Furlani*, M. D. Jones, S. M. Gallo, A. E. Bruno, C. D. Lu, A. Ghadersohi, R. J. Gentner, A. Patra, R. L. DeLeon, G. von Laszewski, Li. Wang and A. Zimmerman, Performance Metrics and Auditing Framework Using Applications Kernels for High Performance Computer Systems, Concurrency and Computation, submitted.

[8] University at Buffalo Metrics on Demand (UBMoD): Open source data warehouse and web portal for mining statistical data from resource managers in high-performance computing environments. Developed at the Center for Computational Research at the University at Buffalo, SUNY. Freely available under an open source license at SourceForge at http://ubmod.sourceforge.net/.

[9] XDMoD Portal - https://xdmod.ccr.buffalo.edu.

[10] XSEDE Overview - https://www.xsede.org/overview.

[11] D. L. Hart, Measuring TeraGrid: Workload Characterization for an HPC Federation. IJHPCA, November 2011, 25(4): 451-465. Advance online publication, Feb. 10, 2011. doi:10.1177/1094342010394382.

[12] D. L. Hart, Deep and Wide Metrics for HPC Resource Capability and Project Usage, SC11 State of the Practice Reports (Seattle, WA), 2011. doi:10.1145/2063348.2063350

[13] P. M. Bennett, "Sustained systems performance monitoring at the U. S. Department of Defense high performance computing modernization program," in *State of the Practice Reports*, SC11, Seattle WA, November 2011, http://doi.acm.org/10.1145/2063348.2063352.

[14] M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J.J. van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, W.A. de Jong, NWChem: a comprehensive and scalable open-source solution for large scale molecular simulations, Comput. Phys. Commun. 181, 1477, 2010, http://dx.doi.org/10.1016/j.cpc.2010.04.018.

[15] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, Scalable molecular dynamics with NAMD, Journal of Computational Chemistry, 26:1781-1802, 2005, DOI10.1002/jcc.20289.

[16] J. Hammond. "TACC_stats: I/O performance monitoring for the intransigent," In 2011 Workshop for Interfaces and Architectures for Scientific Data Storage, IASDS 2011.

[17] C. Lu, J. C. Browne, R. L. DeLeon, J. Hammond, B. Barth, A. K. Patra, T. R. Furlani, M. D. Jones, S. M. Gallo; "Comprehensive Resource Measurement and Analysis for HPC Systems," presented at XSEDE 12, 2012.