2011 IEEE International Parallel & Distributed Processing Symposium

Cyberinfrastructure Usage Modalities on the TeraGrid

Daniel S. Katz
Computational Institute
University of Chicago & Argonne National Laboratory
Chicago, USA
d.katz@ieee.org

David Hart
Computational & Information Systems Laboratory
National Center for Atmospheric Research
Boulder, USA
dhart@ucar.edu

Chris Jordan
Texas Advanced Computing Center
University of Texas Austin
Austin, USA
ctjordan@tacc.utexas.edu

Amit Majumdar
San Diego Supercomputer Center
University of California San Diego
San Diego, USA
majumdar@sdsc.edu

J.P. Navarro
Computational Institute
Argonne National Laboratory & University of Chicago
Chicago, USA
navarro@mcs.anl.gov

Warren Smith
Texas Advanced Computing Center
University of Texas Austin
Austin, USA
wsmith@tacc.utexas.edu

John Towns
National Center for Supercomputing Applications
University of Illinois
Urbana, USA
jtowns@ncsa.illinois.edu

Von Welch
Indiana University
Bloomington, IN
vwelch@indiana.edu

Nancy Wilkins-Diehr
San Diego Supercomputer Center
U. of California San Diego
San Diego, USA
wilkinsn@sdsc.edu

Abstract—**This paper is intended to explain how the TeraGrid would like to be able to measure "usage modalities." We would like to (and are beginning to) measure these modalities to understand what objectives our users are pursuing, how they go about achieving them, and why, so that we can make changes in the TeraGrid to better support them.**

Keywords-component; production grid infrastructure; cyberinfrastructure

I. INTRODUCTION

The TeraGrid represents one of an emerging class of entities that can be referred to as "production cyberinfrastructures." These cyberinfrastructures, which also include the Open Science Grid in the US, DEISA and EGI in Europe, and RENKEI in Japan, harness distributed resources, including high-performance computing (HPC), storage, and specialized computing systems. These combinations of integrated resources make possible a variety of user and usage scenarios. To better support our user community, TeraGrid wants to understand how users are actually using these resources to achieve their scientific objectives, so that we can make changes in the TeraGrid environment to improve operations and services. Because TeraGrid will be transitioning into a new project (eXtreme Digital, or XD) in mid-2011, much of this data gathering will be given to the new project and can be used in planning its growth.

A great deal of attention has gone into the study of HPC workloads, with the goals of improving scheduler performance and maximizing resource utilization. A study of the TeraGrid's workload patterns shows that, in many ways, the TeraGrid's HPC resources demonstrate many of these same patterns, both on individual systems and across the federation [1].

1530-2075/11 $26.00 © 2011 IEEE
DOI 10.1109/IPDPS.2011.239

However, while standard workload analyses can describe what users are doing on TeraGrid's HPC systems, they cannot easily be used to understand *why* users do what they do and how they leverage multiple types and instances of cyberinfrastructure (CI) resources. We have labeled these generalized, CI-spanning classes of user activity "usage modalities" and have begun efforts to instrument TeraGrid infrastructure services so that we can conduct quantitative analysis of user behavior in terms of these modalities.

The paper is organized as follows. We first define the modalities we think are important to measure. We then discuss what is possible to measure in general, and on the TeraGrid. We next explain the characteristics of the modalities we have selected: user intent, when-to-run, submission mechanism, resources, job coupling, support, and level of software development, and present some early measurements we have taken, for allocations questions asked during June 2010 through January 2011 and from jobs run during Q4 (October to December) 2010. We finally discuss what else we would like to do, both internally on TeraGrid and with other infrastructures, and then conclude.

II. Defining Modalities

Our first challenge was defining what modalities we needed to measure. Initially we attempted to enumerate all possible modalities, but it quickly became apparent that this approach was insufficient, as it quickly became a lengthy list that was hard to group into useful elements. Rather, we concluded that we needed to define a matrix of modalities, with each modality defined by a set of possible values for a number of common usage attributes. These attributes are not necessarily continuous dimensions, so a user's modality is more correctly defined as an attribute-value "tuple" within a tuple-space that describes a usage behavior: how a user is using TeraGrid at a particular time. Thus, these modalities suggest the session classes that can be inferred from HPC job logs [2], but represent a more complex construct, potentially spanning many types of CI resources.

The attribute-value approach to defining a modality offers several advantages. First, in addition to a fully defined matrix, it also allows us to express modalities at different levels of granularity. We can express, for example, the set of modalities defined by only two characteristics while ignoring other attributes that are not important to the analysis at hand. We can then refine the analysis, by incorporating additional attributes as needed.

Second, the challenge of measurement becomes much more tractable. To measure a modality, TeraGrid needs only to be able to measure the set of characteristics that express that modality. The converse is also true; each of the modality characteristics must be defined as measureable properties for which instrumentation could, in theory, be implemented. Characteristics that cannot be measured are not useful discriminators of different modalities. It also makes it possible to measure some modalities without having to be able to measure all modalities. That is, the characteristics we can measure define some parts of the overall modality tuple-space, which we can usefully study while designing the instrumentation for the characteristics we cannot yet measure.

III. General Instrumentation and Measurement Requirements

The first implicit requirement in the characteristic-based definition of modalities is the ability to measure each characteristic in terms of a common set of user identities. That is, it must be possible to know that a given user in the measurement for one characteristic is the same as a given user in the measurement of a second characteristic. Without this identity linkage, we are left only with independent measures of each characteristic. We would be unable to make assertions about which modalities, as defined by the intersection of those two characteristics, are most prevalent.

Second, these characteristics must apply to a common unit of usage, which currently in the HPC-oriented world of TeraGrid is a batch-queue job. A job is most often measured in two dimensions: time (duration) and space (size, cores, etc.). Most commonly, a job is a single run of an application, but a job may also be part of a larger ensemble of jobs, or it could be part of a workflow consisting of a set of sequentially dependent jobs, or a job could be one of a set of jobs that together comprise an MPI application that is run across two distinct clusters. In some cases, a job may also be an interactive session, for example, on a visualization system. To accurately understand usage modalities, we must be able to implicitly infer or explicitly annotate these sorts of relationships between jobs.

Thus, understanding usage modalities requires tracking usage by user, by job, and by job usage (time x space). Most HPC jobs are measured in terms of core-hours (number of hours x number of cores). TeraGrid uses the concept of "normalized units" (NUs) as a unit of compute usage that permits usage comparisons across heterogeneous systems. One NU is equivalent to a Cray X-MP core-hour and the conversion from a system's local core-hours to NUs is calculated based on its HPL benchmark performance. As an example, one core-hour on either Ranger, the Sun Constellation Cluster at TACC equals about 35 NUs, or on a modern desktop PC, about 45 NUs.

Storage consumption is measured by terabyte-year (TB/yr) or gigabyte-year (GB/yr), intended to encapsulate both the amount of space consumed and the time that data spends resident on storage devices. For purposes of allocating storage, we only recognize usage in units of a year; this presents challenges in measuring user modalities including scratch or temporary storage. Therefore, there are two metrics of interest used to measure usage in this context: a simple absolute value of storage used, which is used to measure data moved or generated in association with a computation or a visualization/analysis task, and which may be assumed to be temporary; and a terabyte-year metric, which is used to measure persistent data stored by TeraGrid users. These measures are applied to whole resources based on the usage model defined for those resources. For example, because an archive storage resource is not used for temporary storage, usage will always be measured in terabyte-years.

Analyzing usage modalities that span resource types requires conversion factors between the local units of each type. While several individual resource providers within the TeraGrid have their own conversions between storage and compute metrics (for example, NCAR's "generalized accounting unit," or GAU, formulas allow conversions between HPC core-hours and archival system storage units), there is no general TeraGrid conversion factor and we do not assert any specific equivalence here.

Another set of resources that is important to TeraGrid is network bandwidth. However, as we don't have any means to tie network usage to specific users or usages currently, we reluctantly do not have any modalities that involve network usage currently. We recognize this as a shortcoming, and would like to change this in the future.

IV. TeraGrid Modality Characteristics

Once we recognized the need for multiple modality characteristics, our initial enumeration led us to identify seven attributes of usage modalities: user intent, when-to-run, submission-mechanism, resources, job coupling, support, and level of software development. It is likely that other infrastructures may have slightly different characteristics and possible values for defining modalities that would be applicable, but we think this is a good starting point.

We also want to state that we are aware we may not be able to measure all of these characteristics with complete accuracy; our goal is just to be representative.

A. User Intent

TeraGrid has identified "user intent" as an important modality characteristic with three possible values: production, exploration/porting, and education. In the TeraGrid, user intent information is captured as part of the allocations process. Each project requests an allocation, and provides certain information about what they plan to do as part of that request. This information is captured in the POPS allocation request system.

Production means actually doing computational research and has several flavors. Some users do production for a short burst and then stop. Other users have been running large-scale production runs for years. These variants may not matter, except in terms of the type of user support needed. Exploration/porting includes when a new system, application, or problem is being examined and the phase where a user is porting/scaling their code to use the system in question. Education as a user-intent value meaning that the resource is being used for classroom instruction or training, perhaps that a number of students are learning how to use a system or application.

There are currently three types of allocations that a TeraGrid user can request: Research, Startup, and Education. It's likely that education allocations correspond to education as a user-intent, and hence, jobs that are run using an Education allocation have the education user-intent modality. However, the correspondence between the other two types of allocation, Research and Startup, to the other two user-intent modalities, production and exploration/porting, are not as clear. In general, jobs run within a Research allocation can be almost completely mapped to the production modality, but jobs run within a Startup allocation currently fall into two categories: small-scale production (which should be mapped to the production modality) and exploration/porting (which should be mapped to the exploration/porting modality.)

We have begun asking users the following question, as part of their allocation request:

Please estimate what percentage of the work you expect to do in this allocation will be the following types (the 3 numbers should sum to 100):
__ Production (actually doing research)
__ Exploration/porting (preparing to do research)
__ Education (teaching others to do research)

Once we have this information, we can then map the jobs run under this allocation with these weights. For example, to measure this modality for a given user, we just use the numbers the user provided – a user who answers 90% production and 10% exploration and porting would count as 0.9 users doing production, and 0.1 doing exploration/porting. Similarly, if this user runs 10 jobs, we would count this as 9 production jobs and 1 exploration/porting job. And if these 10 jobs use 100,000 NUs, we would say that 90,000 NUs had been used for production, and 10,000 for exploration/porting.

Our initial data for this question spans 547 Startup and 231 Research allocations submitted from June 2010 to January 2011 (Figure 1). As expected about 90% of the Research allocation work was classified as "Production." Within the Startup allocations, though, only 30% of the work was self-reported as being "Exploration;" 60% of the work was classified as "Production" by the request submitters.

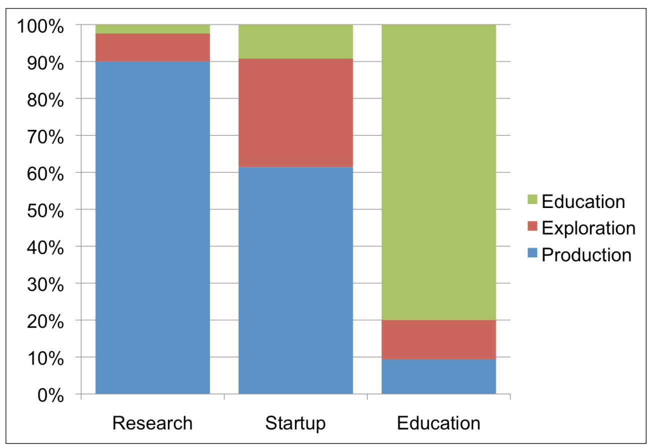

Figure 1. Self-reported user intent, by allocation type, on TeraGrid.

B. When-to-run

We identified four values for the "when-to-run" characteristic: batch, interactive, high-priority, and reservation. Batch means that the job is not time-sensitive or time-critical. Interactive means that they job should run when the user is available. High-priority means that the job should run soon, possibly immediately (the job should run now, and may need to preempt other running jobs), or possibly next-to-run (other running jobs shouldn't be preempted, but the job should run as soon as otherwise possible). Finally, reservation means that the job should run at a time the user specifies (e.g., it is co-scheduled with jobs running on other resources or with an instrument reservation, or at a time when an audience will be available to watch the job). (Note that in many cases, all of these jobs are run through the batch scheduler; the term batch in this modality is not the same as just using the batch scheduler, but it's using the batch scheduler in a "normal" mode.)

On TeraGrid today, most jobs are run in the batch modality, and this is the default category for counting jobs, i.e., we assume the total number of jobs that have been run are in the batch category, then subtract off others as we count them. We don't currently have an implementation of the interactive modality, so there are no jobs to count for it. For the high-priority modality, we have two implementations: SPRUCE [3] and site-specific policies. SPRUCE is a software tool (client/server) that permits urgent jobs. It could count the jobs that are run on TeraGrid under both urgent modalities. Alternatively, the sites that run urgent jobs could add a flag to the accounting records when they run such jobs. Reservations are similar to urgent jobs in terms of tracking, since there are two TeraGrid tools that are used for reservations, plus additional site-specific means. Again, either the tools or the sites could count and flag reserved jobs.

C. *Submission-mechanism*

There are four submission-mechanism modalities: command line, grid tools, science gateway, and metascheduling. Command-line can also be thought of as including the idea of local, as this is meant to indicate that a user has logged into a system and used a local interface, such as the command line, to submit a job. Grid tools means that the job has been submitted through grid tools, such as Globus GRAM [4], gLite CREAM [5], and UNICORE JMS [6] (OGSA-BES [7] is a service that sits on top of these tools), usually from a system that is different than the system on which the job will be run. Science gateway is meant to indicate that the user does not submit a job directly, but that the job is submitted based on a request from the user, possibly through a portal. Metascheduling can underlie any of the other submission-mechanism modalities – it indicates that the user does not pick a specific resource on which to run the job, but rather, a set of resources are specified, and some additional logic decides on which of those resources to run the job.

TeraGrid uses Globus GRAM as its primary grid submission mechanism. With a Globus "listener" capability [8], TeraGrid is able to record the number of jobs submitted to each resource via GRAM. However, the linkage between GRAM jobs and accounting records of jobs has not yet been fully made. Until recently, the local resource provider accounting systems did not record how a job was submitted. Now, four sites have deployed "GRAM audit tools" that annotate batch job accounting records with any GRAM-related submission data. Initially designed to record end-user data from gateways, the tools also allow a site to annotate any GRAM-submitted job, prior to sending that job to the TeraGrid Central Database (TGCDB) to charge the relevant project. With GRAM audit tools and other information, we can more accurately extract the number of jobs submitted via gateways. We do not currently have instrumentation that can annotate jobs associated with metaschedulers, though the process would be essentially the same as with the GRAM audit tools, except that data from metascheduler logs would have to be retained throughout the job accounting process.

Because of the difficulty of instrumenting these submission mechanisms, we also asked allocation requestors to estimate how they intended to submit their jobs:

Please estimate what percentage of the jobs you expect to run in this allocation will be the following types (the 3 numbers should sum to 100):

__ *Submitted through command line/script*
__ *Submitted using Grid tools (such as GRAM)*
__ *Submitted through a metascheduler (to run on one of a set of resources, without user control over which of the set is chosen)*

Our initial data from June 2010 to January 2011 in Figure 2 show that Research allocation users are the most conservative, with 93% of the jobs expected to be submitted via the command line. Startup and Educational users were slightly more likely to use grid tools—10% for Educational and 9% for Startups—and a metascheduler—10% for Startups and 5.5% for Educational allocations. Examination of the individual responses shows that the planned use of grid tools and metaschedulers was not evenly distributed, but affected by a small subset of the projects. Only about 30% of the Startup requests planned to make any use of grid tools or metaschedulers.

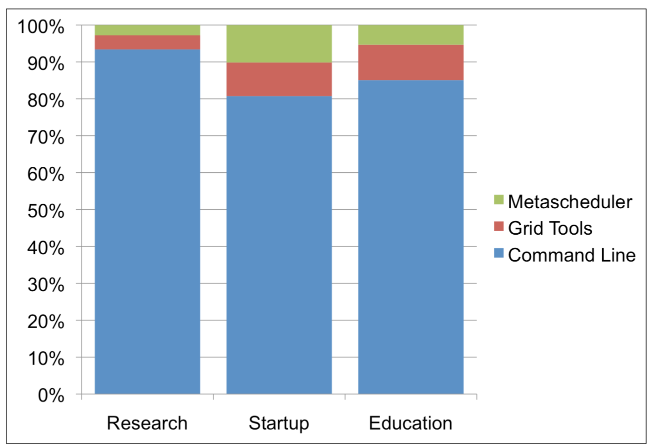

Figure 2. Self-reported planned job submission mechanism, by allocation type, on TeraGrid.

Using our early data, we have performed a coarse comparison of the self-reported intent to use grid tools with the actual reported GRAM jobs recorded by the TeraGrid Globus listener. In Q4 2010, TeraGrid's Globus listener recorded 41,363 GRAM jobs across TeraGrid resources, while the TeraGrid accounting system recorded 993,404 job records, with 85% associated with Research allocations, 11% with Startup allocations, and the remainder with Educational and other projects (such as staff projects). Using the average expected percentage of grid jobs in each category, we would predict then that just over 44,000 jobs would have been submitted via GRAM in the quarter. Further analysis is required to understand whether the actual and predicted numbers of grid-submitted jobs are within 6% of one another by coincidence or by correlation.

D. Resources

We have identified eight values that characterize the resources used on an infrastructure by a job: One HPC resource, one HTC resource, multiple HPC resources, visualization resource, Data Intensive Resource, Archival storage resources (amount), Multi-site storage resources (amount), and Non-TG resources. In general, this is the characteristic we can most accurately measure today, with the exception of "non-TeraGrid resources," by which we would like to understand a user's or project's usage of TeraGrid resources in concert with other resources outside of TeraGrid.

Because the current TeraGrid accounting system includes the resource as part of every job record, we can determine whether one or more resources have been used in any project or by any user. We can also identify usage modalities by resource type (e.g., HPC, HTC, storage). Non-TeraGrid resource information would have to be captured via surveys or as part of allocation requests, as with User Intent information. Figure 3 shows how many of the 959 TeraGrid projects that were active in Q4 2010 used resources of different types. Of the three projects using 6 and 9 different HPC resources, two are TeraGrid staff projects. The consumption of SUs according to resource type is similarly skewed toward projects using a small number of HPC resources. The same data set also allows us to show that most of the active projects (86%) used only one resource type during the quarter, but 13% of projects did make use of two resources types.

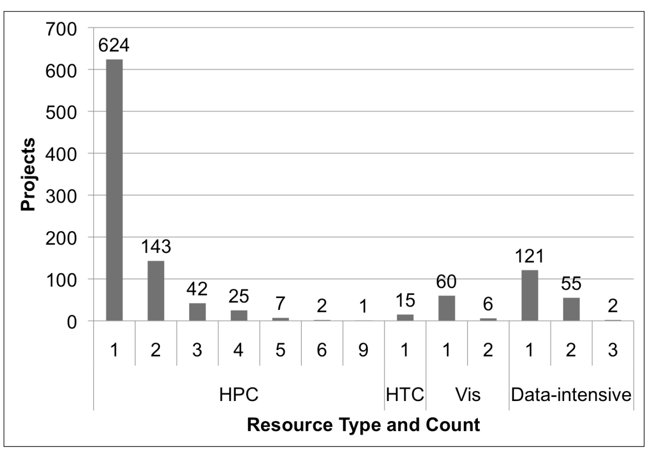

Figure 3. TeraGrid projects making use of different numbers and types of resources in Q4 2010.

Figure 4 shows how many of the 1,904 individual TeraGrid users that were active in Q4 2010 used resources of different types. Again, the use of one HPC resource dominates the pattern. The users associated with seven and eight HPC resources are a science gateway community user account (nanoHUB [9]) and TeraGrid's Inca [10] monitoring user account, respectively. Compared to projects, a greater percentage of users (92%) used only one resource type; 8% of users used two resource types.

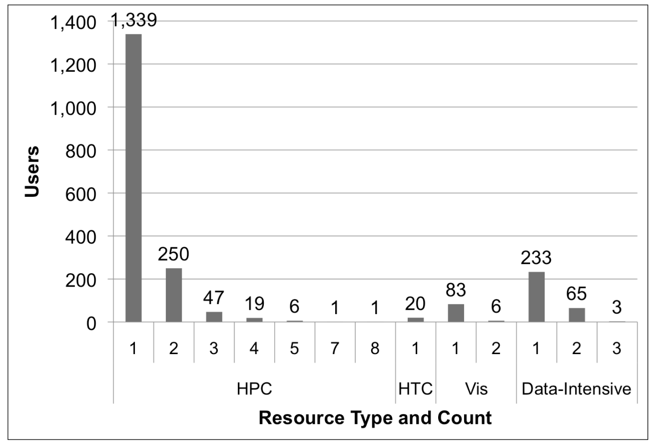

Figure 4. TeraGrid users making use of different numbers and types of resources in Q4 2010.

In capturing usage modalities related to data, there are a few basic patterns of usage we would like to track, which imply some specific measurements corresponding to jobs or processes launched by users. We wish to measure the persistent use of archive and other storage resources over the course of allocations; we wish to measure the use of multi-site storage resources such as GPFS- and Lustre-WAN file systems within computation and visualization tasks which may span multiple sites; and we wish to measure the extent of data movement in preparation for and in cleanup after a task is run on the TeraGrid. Each of these presents specific information-gathering challenges.

Technically, the simplest measurement to make is of persistent use of a storage resource. Most archives and file systems store metadata describing the size, ownership, and group ownership of all data objects; thus it is possible on all major storage systems to generate summaries of numbers of files, sizes of files, and total usage on a per-user or per-group basis. However, depending on the technical implementation, generating such reports can be extremely time- and resource-intensive, meaning that the frequency of measurement must be carefully chosen to avoid disruption to normal operations. Currently, total usage information over all TeraGrid users is gathered for persistent TeraGrid storage resources on a monthly basis, but more detailed reporting is possible.

Measurement of multi-site storage usage is similarly simple at a technical level, but the availability of a GPFS-WAN or Lustre-WAN file system on multiple resources leads to complexities of measurement, since what we really want is not just a total usage number, but an association of data objects or data sets with one or more resources that generate or consume this data. This will provide us with a more complete picture of both the transience of computing on TeraGrid resources and the extent to which multi-site storage resources encourage this transience. To fully generate this information, instrumentation is needed on the compute nodes of TeraGrid resources. With appropriate information, statistics could be generated either in real-time or at the end of jobs, showing the total data flows into and out of each node. Again, however, extensive instrumentation of this type could have a performance impact and must be carefully designed to balance the desire to gather complete data about the use of storage resources with the need to minimize the impact on users.

The measurement of data transfer modalities in the TeraGrid is greatly assisted by the existence of a Globus GridFTP listener, similar to the Globus GRAM listener described previously, that generates centrally collected reports of all transfers initiated on TeraGrid resources. Since GridFTP accounts for the majority of data transfer traffic within the TeraGrid, these listener statistics are an excellent way to generate high-level summaries of data flows through the TeraGrid. At present, however, these statistics do not include per-user summaries, nor are they in any way correlated with projects or with specific resources, i.e., if multiple storage resources are available at a given network endpoint, the listener only gathers information about the endpoint, not the individual storage systems. To truly provide a characterization of usage modalities in data transfer, this more fine-grained information is needed.

E. *Job coupling*

There are four modalities for job coupling: independent, independent but related, interdependent, and dependent. Independent jobs are those that are not immediately connected to any other job. Independent but related jobs include those jobs that make up an ensemble or parameter sweep, for example. Interdependent jobs involve multiple jobs that must run simultaneously, and communicate often. Dependent jobs are elements of a set of multiple jobs, such as in a workflow, where the completion of one or more jobs enables one or more other jobs in the set to run. Note that a job that is broken into chunks by queue limits might be considered a set of dependent jobs under these definitions. The only way TeraGrid can currently categorize jobs is by asking the user. Ideally, if all users used a fixed set of tools, we could instrument the tools to gather this information, but this is not feasible in practice, as while many users will use a standard set of tools, others will write their own scripts or create other mechanisms for controlling their jobs.

We have begun asking users the following question, as part of their allocation request:

Please estimate what percentage of the science runs you expect to perform in this allocation will be the following types (the 4 numbers should sum to 100):

__ *Independent (a job that is not immediately connected to any other job - a job that is artificially broken into chunks by queue limits should still be placed this category)*

__ *Independent but related (such as jobs that make up an ensemble or parameter sweeps)*

__ *Tightly coupled (multiple jobs that will run simultaneously)*

__ *Dependent (multiple jobs such as in a workflow)*

Once we have this information, we can then map the jobs run under this allocation with these weights. For example, to measure this modality for a given user, we just use the numbers the user provided – a user who answers 70% independent but related and 30% dependent would count as 0.7 users doing independent but related work, and 0.3 doing dependent work. Similarly, if this user runs 10 jobs, we would count this as 7 independent but related jobs and 3 dependent jobs.

Our initial data from allocation requests during June 2010 to January 2011 in Figure 5 indicate that submitters expect most jobs to be Independent (54%) or Independent/Related (28%), with about 10% categorized as Tightly Coupled and 7% as Dependent. The values for both Startup and Research requests were aligned with the overall averages. Among Educational requests, a slightly larger percentage of jobs (68%) were classified as Independent, which is consistent with jobs submitted as part of small-scale class assignments.

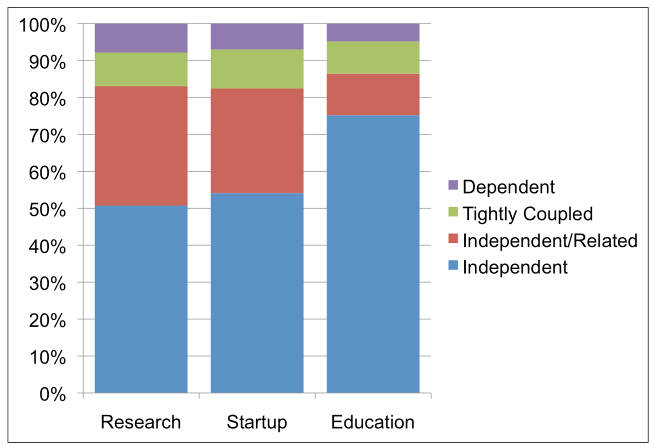

Figure 5. Self-reported job coupling breakdown, by allocation type, on TeraGrid

F. *Support*

In TeraGrid, there is currently one attribute value for the support characteristic, ASTA support. ASTA (Advanced Support for TeraGrid Applications) is the advanced support mechanism where users request advanced support as a part of their allocations request for resources. The allocation review committee then reviews this request, and a recommendation score is provided to the Advanced User Support (AUS) area director. Based on this score, further discussions with the requesters, and other factors, such as availability of staff with the necessary expertise, resource provider site(s) where the user requested allocation, AUS staff are matched with the allocations request to provide long term collaboration and support for the group of users who made the allocation request. An ASTA may last up to the lifetime of the CPU allocation, up to one year.

An additional potential value for this characteristic is campus champion support. As of January 2011, there were 82 universities with one or more campus champions.

Campus champions are campus representatives who area local source of knowledge about TeraGrid computing opportunities and resources. Campus champions have TeraGrid allocations that they give out to users on their campuses, and these allocations are flagged in TGCDB. This means that we can say that user X ran a job, and it was charged to allocation Y, which is a campus champion allocation. However, it is also possible that a campus champion could be helping a user with the user's own allocation, in which case there isn't any way of knowing that a campus champion supported the job. For this reasons, we don't plan to track the campus champion value at this point.

In the current TeraGrid, the POPS system records allocations requests and submissions, and supports the review process. Once allocations are approved, the award information is communicated to the TGCDB, which supports the accounting of usage against those allocations. There are two databases associated with the allocations process. Currently, POPS is able to record ASTA request information and reviewer ratings of ASTA requests. However, there has been no automated record in the TGCDB of the ASTA requests that are initiated. The AUS area director manually keeps his own records of which ASTA requests were initiated, and their start and end dates. These are reported in TeraGrid quarterly and annual reports. While sufficient for managing staffing for ASTA requests, this disconnect has prevented us from understanding resource usage as it relates to supporting ASTA requests.

We have implemented a process, using existing mechanisms, such that the AUS area director can record an "allocation" for each of the projects that have ASTA support. This allocation will be the estimated number of months for the support. By leveraging existing mechanisms, the PI of the project will also get a notification of this ASTA support allocation, similar to how the PI would receive information about other resource requests that have been approved. The benefit of this is that we will be able to track (and query) which PIs received ASTA support and for how many estimated months, and we will then be able to match jobs that are run to ASTA support in place while the jobs were run.

For example, using records of ASTA allocations, it is possible to query the accounting database to ask questions such as, "In fiscal 2009, how many core-hours were charged by projects during allocation periods in which they had ASTA allocations?" The resulting data show 27 ASTA-supported projects consumed 112 million core-hours, or about 10% of the core-hours recorded by the accounting system (Figure 6).

G. *Level of software development*

While the level of software development for a given user in a given situation can range from writing completely new software, to modifying an existing code, to developing a script to use a code or developing a code that is based on a library, to executing a preexisting application, here we simplify and only allow two modalities: custom and commodity. Custom applications range from those that are developed from scratch by the user to those created by modifying an already existing application. Commodity applications are those that are executed without being modified by the user, such as applications that are pre-installed on a system by an administrator, or that are part of a science gateway, installed by a developer.

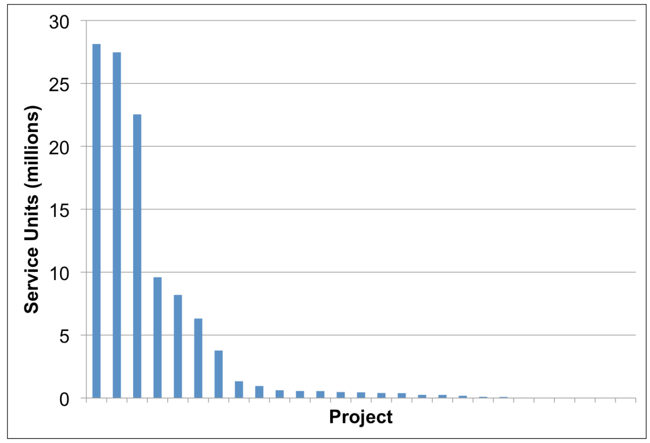

Figure 6. Service Units (SUs) consumed by projects during period of their TeraGrid ASTA activity.

We can likely track the use of installed applications vs. the use of applications in the user's home directory or scratch space. Of course, this is very inaccurate. Another option is to ask the users their intent in the process of their allocation request, as we are planning to do for other modalities.

V. FUTURE WORK

There are a number of activities that would be interesting to undertake, and which may be undertaken by the eXtreme Data (XD) project that will follow TeraGrid. These include: measuring network usage and correlating this to specific users/jobs; looking in more detail about how specific systems are used; and measuring campus champion support as tied to users/jobs. The first and last of these activities have been discussed earlier in this paper. However, the middle idea has not been discussed, because it is not completely clear that it is related to TeraGrid, or that is should be measured by TeraGrid. However, it is clearly an issue that is important to TeraGrid users, so we briefly discuss it here.

This middle idea involves understanding usage of specific systems (e.g., an HPC platform). For example, we might explore how our jobs are dependent on the bandwidth and latency of the interconnect, the memory, or I/O system. Some of this could be done through tools such as NERSC's IPM [11]. Answers to this would be different for parallel/MPI/threaded jobs vs. single process/thread jobs.

A fourth activity to undertake would be to compare our results with those of other infrastructures (international, national, campus, etc.). There are three potential types of knowledge we could obtain by doing this. First, other infrastructures might propose other important question to ask, or might have other tools for gathering data that would help us better understand our own usage. Second, comparing our users with the users of other infrastructures also will likely help us understand the user community and our place in the provider community. Finally, sharing this knowledge of our users with other infrastructures might allow various infrastructure providers to determine common approaches to solve common problems, or possibly actually work together to solve them.

VI. CONCLUSIONS

There are a lot of things the TeraGrid would like to understand better about what our users are doing in order for the project to better support these users. In this document, we have presented a first set of characteristics (usage modalities) that we believe will allow us to gain this understanding, and we have started to make the process changes needed to gather data that can be used to generate this knowledge, by asking users new questions when they apply for allocations, and by modifying our advanced support policies and procedures so that we can add more information to our central database about users and their usage modalities.

ACKNOWLEDGMENT

This work was supported by the TeraGrid, under multiple NSF awards, including OCI-0503697 and OCI-0932251.

REFERENCES

[1] D. Hart, “Measuring TeraGrid: Workload Characterization of an HPC Federation,” *International Journal of High Performance Computing Applications*, 2011. DOI: 10.1177/1094342010394382

[2] J. Zilber, O. Amit, D. Talby, “What is Worth Learning from Parallel Workloads? A User and Session Based Analysis,” *Proceedings of the 19th Annual International Conference on Supercomputing* (Cambridge, Massachusetts, June 20 - 22, 2005). ACM, New York, NY, pp. 377-386.

[3] P. Beckman, S. Nadella, N. Trebon, and I. Beschastnikh, “SPRUCE: A System for Supporting Urgent High-Performance Computing,” in *Grid-Based Problem Solving Environments*, IFIP International Federation for Information Processing, v. 239, (Editors: P. Gaffney and J. Pool), pp. 295-311, 2007. DOI: 10.1007/978-0-387-73659-4_16

[4] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S. Tuecke, “A resource management architecture for meta-computing systems,” in: D. Feitelson, L. Rudolph (Eds.), Job *Scheduling Strategies for Parallel Processing*, Lecture Notes in Comp. Sci. 1459, Springer, 1998, pp. 62–82, DOI: 10.1007/BFb0053981

[5] C. Aiftimiei, P. Andreetto, S. Bertocco, S. Dalla Fina, A. Dorigo, E. Frizziero, A. Gianelle, M. Marzolla, M. Mazzucato, M. Sgaravatto, S. Traldi, L. Zangrando, “Design and implementation of the gLite CREAM job management service,” *Future Generation Computer Systems* 26(4), 2010, pp. 654-667. DOI: 10.1016/j.future.2009.12.006

[6] M. S. Memon, A. S. Memon, M. Riedel, B. Schuller, S. van de Berghe, D. Snelling, V. Li, M. Marzolla, P. Andreetto, “Enhanced resource management capabilities using standardized job management and data access interfaces within UNICORE Grids,” *2007 International Conference on Parallel and Distributed Systems* 2, pp. 1-6, 2007. DOI: 10.1109/ICPADS.2007.4447834

[7] A. Grimshaw, M. Morgan, D. Merrill, H, Kishimoto, A. Savva, D. Snelling, C. Smith, D. Berry, “An Open Grid Services Architecture Primer,” *Computer* 42(2), pp. 27-34, 2009. DOI:10.1109/MC.2009.35

[8] Globus Project. “Metrics, a Globus protoproject”. http://incubator.globus.org/metrics/

[9] G. Klimeck, M. McLennan, S. P. Brophy, G. B. Adams III, and M. S. Lundstrom, "nanoHUB.org: Advancing Education and Research in Nanotechnology," *Computing in Science and Engineering* 10(5), pp. 17-23, September/October, 2008. DOI: 10.1109/MCSE.2008.120

[10] S. Smallen, K. Ericson, J. Hayes, and C. Olschanowsky, “User-level Grid Monitoring with Inca 2,” *Proceedings of the 2007 Workshop on Grid Monitoring*, 2007. DOI: 10.1145/1272680.1272687

[11] J. Borrill, J. Carter, L. Oliker, D. Skinner, R. Biswas, “Integrated Performance Monitoring of a Cosmology Application on Leading HEC Platforms,” *International Conference on Parallel Processing*, pp. 119-128, 2005.