Building a Campus Grid: Concepts and Technologies

September 2005

(Revision 1.1, September 21, 2005)

NSF Middleware Initiative (NMI) Integration Testbed Case Study Series: Supplemental Documentation

This document is a technical supplement to the NMI Testbed Grid case study, "Exploring Technical and Policy Considerations for Inter-Institutional Grids", part of the NMI Integration Testbed Case Study Series. The NMI Testbed Grid began as a sub-project of the NMI Integration Testbed program in September 2003 and is continuing beyond the program as SURAgrid, a multi-institutional cooperative effort to model the use of grid technology to share distributed resources across institutional boundaries while maintaining local autonomy.

For more information on the NMI Integration Testbed Case Study Series, the NMI Testbed Grid project, or SURAgrid, contact:
Mary Fran Yafchak
Southeastern Universities Research Association (SURA)
maryfran@sura.org

The NMI Integration Testbed Case Study Series was sponsored by the NMI Enterprise and Desktop Integration Technologies (NMI-EDIT) Consortium of EDUCAUSE, Internet2, and SURA, under National Science Foundation Cooperative Agreement NSF 02-028, ANI-0123937

Contributors

Name	**Institution**
Adiga, Ashok	Texas Advanced Computing Center at UT Austin
Barzee, Kathleen	Southeastern Universities Research Association (SURA)
Bolet, Victor	Georgia State University
Henderson, Shelley	University of Southern California
Jokl, Jim	University of Virginia
McKee, Shawn	University of Michigan
Perez, Jerry	Texas Tech University
Robinson, John-Paul	University of Alabama at Birmingham
Vandenberg, Art	Georgia State University
Yafchak, Mary Fran	Southeastern Universities Research Association (SURA)

Introduction

Grid technologies represent a significant step forward for effective use of network-connected resources. A campus grid in particular can be a powerful vehicle to realize the full research potential of a university, facilitating the sharing of distributed resources while respecting the distinct administrative priorities of individual resource centers, building inter-disciplinary collaboration where it might not otherwise have occurred. A campus grid provides a means for resource owners to trade their unused cycles for access to significantly more compute power when needed for short periods. In addition, the availability of a campus grid can bring about organizational and cultural change, with participants more willing to invest in common infrastructure (networks, computing center floor space, institutional storage, etc.) if it is felt that such infrastructure will be available for broad benefit and impact.

This paper brings together the insights of campus grid building efforts at a variety of institutions in various stages of their campus grid deployment (mainly using Globus grid software, www.globus.org). The authors wish to share their experience (both good and bad) and perspectives gained while building their campus grids in order to help other institutions make informed decisions as they contemplate uses of grid technology. We believe that the research community at large can benefit from encouraging, participating in and supporting this type of collaboration.

What is a Campus Grid?

What is a Grid?

The most general definition of grids is that they consist of shared heterogeneous resources networked across administrative boundaries. Grid software infrastructure assumes

- Hardware & software exists to support the network
- Out-of-band agreements for resource sharing

For any particular grid, the elements might be specialized. For instance, for a dynamic company consisting of numerous autonomous business units that get bought and sold, a company grid could be implemented with the assumption that individual units and resources may change but, when present, they are available to all company users. Another example would be sharing within an industry, where the primary resource shared are data exchanged and/or transformed under conditions of strict security. A third example is a cooperative research project, where the resources are truly heterogeneous – ranging from sensors in the field to large storage arrays to computational resources to visualization resources such as immersive displays. To gain benefit from running on a grid, the task under consideration must require sharing something large, unwieldy or dispersed across administrative boundaries, such that the ability to combine the specific set of resources is what makes the task possible.

There are also several products available for implementing grids today and, since standards are still developing, product selection can affect the definition of a grid. Any given grid is partially defined by the functionality, focus and features of the product(s) used to implement it. In this paper, high level and general examples consider a variety of “grid types” but specific examples generally refer to grids based on Globus[1], currently the most popular open-source grid software and the primary product commonly implemented by the authors to date.

A Grid or a Cluster?

Clusters are often compared to, and confused with, grids. A cluster can be defined as a group of closely coupled computers joined together through a common operating system, security infrastructure and configuration that are used as a group to handle users’ computing jobs.[2] Types of clusters include:

- High performance computing (HPC) clusters offering the significant computational capacity of SMP computers. These clusters often have high-speed interconnections between the compute nodes to support efficient inter-process communication.
- Beowulf clusters comprised of commodity-hardware compute nodes running Linux software.

---

[1] http://www.globus.org

[2] http://www.beowulf.org/overview/index.html

- Condor[3] clusters (flocks) comprised of distributively-owned computing resources (not necessarily running the same operating system) designed to offer high throughput computing (HTC) over long periods of time.

In this paper, a *grid* is most basically defined as two or more loosely coupled clusters, SMP (symmetric multi-processing) or cycle scavenging installations (e.g., Condor flocks) that connect across administrative domains. A computational grid uses collective nodes for simulation or other processing; a data grid stores and provides access to collected data across grid nodes.

Basic Elements of a Grid

A campus grid requires a minimum set of basic elements in order to function properly. Functionality may be driven by the particular needs of researchers who utilize the grid, however, there is a basic design that should define a minimum set of grid computing functions. The minimum functional elements of a grid are:

- Accessibility (via Portals)
- Data movement
- Resource management
- Job submission
- Monitoring
- Administration

Metascheduling and Accounting are also desirable.

Portals significantly enhance the accessibility of grid computing to non-technical users. However, even with this improved user experience, portals still require users to be aware of resource specific details and make educated decisions in selecting a specific computational resource for their job. This problem is complicated by the fact that different users on the same portal may be authorized to use a variety of compute resources. Today a user must be aware of the resources they are authorized to use and must explicitly check each resource's current operational status before picking one and submitting their job to it. A user would prefer to simply have their job run on whatever resource or combination of resources would ensure the best performance.

Data movement is required to provide a reliable access point to stored data that is used or created by compute resources. The amount of data to be moved may be huge, depending on the particular application. Data transfer may occur under a couple of different scenarios:

- Autonomously - independent of any particular submitted job (e.g., ad hoc file transfer or a scheduled data transfer via dedicated network shares or data grids).
- Staging - manually uploading the data to the clusters to ensure that data is available when and where it is needed for a particular job.
- As a result of computation, conditional on the outcome of the computation.
- During a computation, as an intermediate stage of the computation "data pathways".

Resource management is a necessity as grids require some information about the resources that can be allocated for a given grid job. A resource management service can test the conditions of the allocated grid resources and launch the actual execution if all the conditions are met – possibly under conditions where real-time interaction with the user is impractical (e.g., remote location, time difference) – and report back on what happened. Resource allocation is facilitated by the use of grid monitoring, with grids being instantiated using constantly running background processes (daemons). In Globus, examples of these background processes include a gatekeeper for submission of a job for execution on a node, a GRIS process that maintains information on software and hardware configuration for a specific node, and a GIIS process that aggregates GRIS information for a collection of nodes.

Once the resources have been identified and codified, the information about the grid resource can be sent to a **metascheduler** for more effective scheduling of grid resources. Metaschedulers operate at the grid level across potentially numerous resources and assign user jobs to the most suitable resources. Building a metascheduler is an advanced use of the grid and will significantly improve the utility of portals

---

[3] http://www.cs.wisc.edu/condor/

to end-users. This is an active area of research. One such effort is MARS[4], an NSF funded project for the University of Michigan's MGRID to address the lack of a metascheduler in the NMI components.

Job submission by end-users requires some method to define job parameters such as the location path of software or data, the chosen set of computational resources, conditional execution or triggers/blocks, or the required authentication/authorization information.

Prior to job submission, **job monitoring** can give grid users important information about existing grid workloads at various grid sites. Once the job is submitted, grid job monitoring becomes a necessity, as end-users want to keep track of the progress of their job submissions. Grid job monitors can gather vital information about job submissions by harvesting data from local cluster job managers such as PBS, LSF, and Ganglia. An example of a grid monitor is MonALISA (Monitoring Agents in A Large Integrated Services Architecture)[5]. The MonALISA system provides a distributed service for monitoring, control and global optimization of complex systems. MonALISA is based on a scalable Dynamic Distributed Services Architecture (DDSA) implemented using Java / JINI and Web Services technologies. The scalability of the system derives from the use of a multi-threaded execution engine to host a variety of loosely coupled, self describing, dynamic services or agents, and the ability of each service to register itself in order to be discovered and used by other services, or clients that require such information.

Grid administration tools are mostly for controlling authorization and authentication; however, ideally, they should model those used by a typical workstation system administrator. The goal of grid administration tools is to give the administrator the sense of localized control of resources even though the grid resources may not be geographically near the administrator.

Grid-centric **accounting** has a long way to go; currently, it relies for the most part on local accounting on the compute or storage resources. Grid accounting should give grid users and administrators feedback on the resources used by various groups and users. This is very important to let the grid stakeholders realize the impact the grid has had and could be an important component for fairly scheduling future job priority based upon previous use. A "meta-view" is needed for the future. One example of a current grid monitoring and accounting mechanism is Nimrod/G[6]. Nimrod/G is a grid-aware version of Nimrod. It takes advantage of features supported in the Globus toolkit such as automatic discovery of allowed resources. Furthermore, it introduces the concept of computational economy as part of Nimrod/G. The architecture is extensible enough to use any other grid-middleware services such as Legion, Condor and NetSolve.

The Campus Grid Defined

The concept of a campus grid is emerging and may be defined in various ways depending one's perspective. There may be a grid or grids on campus but not necessarily functioning as a campus grid. In addition, the concept of an enterprise grid is emerging in the corporate sector and is sometimes also called a campus grid, but the nature of "resource ownership" and sharing are most likely different from a similar implementation in an academic or research setting.

For the purposes of this paper, a *campus grid* is a grid that

- Leverages centralized campus authentication (AuthN) and authorization (AuthZ).
- Allows heterogeneous resources, for example, different hardware and operating systems.
- Allows heterogeneous users with varying needs.
- Facilitates sharing of computing resources, for example, sharing between departments at the same institution and/or sharing between different institutions.
- A campus grid provides the capability of integrating and orchestrating distributed capabilities to solve a single problem or serve a single inquiry.

---

[4] http://www-personal.engin.umich.edu/~abose/website/mars.htm

[5] http://monalisa.cern.ch

[6] http://www.csse.monash.edu.au/~davida/nimrod/nimrodg.htm

Who Needs a Campus Grid?

Information technology within a campus or other enterprise is intended to support users and applications as they are engaged in some aspect of the organization’s business. On a typical higher education research campus, “business” includes research and teaching in addition to a myriad of administrative and organizational tasks. When deploying a campus grid, although researchers are most often the earliest users, it is important to plan for a wide range of eventual users and uses – some of which cannot be predicted - and architect both the processes and the technology to provide flexibility, efficiency and capability for expansion.

Perhaps the easiest way to discover and classify user groups that can benefit immediately from the use of a campus grid is to describe the nature of the demands of the applications they use, rather than the disciplines in which they would be found. Typical applications poised to benefit from a campus grid likely meet one or more of the following criteria:

- Large, distributed group collaborations, predominantly externally focused
- Applications requiring significant computing cycles (i.e., lots of CPU time)
- Applications that have significant data handling requirements (e.g., access to or transfer of large or distributed data sets)
- Visualization-intensive applications

While applications meeting the above criteria can be found in a multitude of disciplines, high performance computing users working in applied sciences and mathematics are frequently the most regular campus grid users. Still, a campus grid builder should cast a broad net when searching for users that can benefit from the campus grid. Innovative users ready to apply grid technologies to their work may be found in all disciplines, including those not historically thought of as compute intensive, such as the social sciences and the arts. This can be attributed, in part, to the increasing size and use of application databases and the increasing need to manipulate and visualize this data within a variety of disciplines.

Building a Campus Grid

Starting Points

Building a campus grid is a complex process and requires expertise, buy-in, perseverance and, most importantly, time. In addition, implementation must address policies, organizational structure and culture along with technology deployment and, as likely, development. Campuses differ in their preparedness in these areas and also in the amount of human and technical resources that can be brought to bear on the implementation. Because of this, the process of campus grid building is difficult to summarize step-by-step. Common principles and directions do apply, however, and can serve as a framework to develop an effective, tailored implementation plan. It is also important to note that a campus grid isn’t likely to be built in one step. Perhaps the most significant first step for a successful campus grid implementation is to “pick the low hanging fruit”. In this context, the low hanging fruit may be in the form of a willing administration, an enthusiastic researcher, visionary IT staff, a critical collaborative project, or even a timely technology acquisition

One starting point that works for many campuses is to target “cycle-scavenging” – making use of unused compute power – from the many machines already placed in public labs. Matching this unused or under-utilized resource to a “grid-needy” project or department can highlight the benefits of grid capability to a broader campus audience without significant initial investments in acquiring new hardware, or the policy decisions related to sharing new hardware, or sharing resources that are more heavily used. The University of Michigan has been successful with this approach, using PBS to harness unused cycles in Macintosh labs to benefit several departments on campus:

At the University of Michigan we have approximately 300 G5 processors available in some labs around campus owned by our IT Central Services (ITCS). During the day these machines are used for classrooms, labs, as video/editing booths and for general-use access in dorms. Most of these machines sit idle during the night and on weekends and some fraction are even idle during weekdays. MGRID had *some users interested in the MAC OS X platform and saw an opportunity to utilize these otherwise unused cycles. We contacted ITCS and discussed with them our idea to cycle-scavenge. They were happy to have someone make productive use out of these systems as long as we didn't impact their machines intended uses. We initially tried Condor (known for cycle-scavenging) but there were problems with the MAC version of Condor. Since we had a lot of experience with PBS we deployed the MAC version of PBS into the lab, after careful testing on a separate machine. After hours, these machines are now available to run jobs submitted from our MGRID portal. The machines are typically used to run BLAST (bioinformatics) searches, MATLAB jobs, GAUSSIAN and GAMESS (quantum chemistry) and NETLOGO (business: models natural and artificial systems) applications. We are looking to extend this model to encompass other campus computing resources which could be harnessed to provide additional computational power.*

The Texas Advanced Computing Center has two sets of resources on their campus grid that fall into this category – (1) Approximately 1000 desktop PCs belonging to around 10 departments, running the United Devices Grid MP software and (2) a set of desktops, workstations and cluster machines running Condor. The Condor resources are grouped into 3 pools (corresponding to 3 departments at UT) and, using Condor flocking, a total of approximately 1000 CPUs are shared by campus grid users. Both United Devices and Condor proved to be fairly easy to install and manage, and are viable solutions if a campus has desktop and workstation class machines available for use in their grid. Some applications that have been enabled on these desktop grids include graphics rendering applications (POV-Ray[7]), reservoir modeling (IPARS[8]), and protein sequence analysis applications such as BLAST and HMMer.

Georgia State University is also deploying a United Devices GridMP™ capability to harvest spare cycles available on computer-lab and classroom PCs, to support applications in physics simulation, graphics rendering, and neuroscience modeling. Importantly, the campus administration approved the acquisition of the United Devices capability (1000 concurrent licenses) as a cost effective way to harness existing, underutilized resources.

Since the benefits of a grid are often most obvious for compute intensive applications, these are typically the first applications to be run over a campus grid. However, on some campuses, other types of applications (e.g., visualization) may be the lowest hanging fruit. This may be due to the availability of particular technology or unique resources, or the dynamics of working relationships between people or departments, or an application or discipline that is strategically important to the institution. The goal is to have one or two applications to demonstrate as the grid is publicized, then use these examples to illustrate the potential of the campus grid and encourage additional participation and support.

This process of identifying a relevant application, putting it on the grid, and showcasing the benefits to encourage additional usage and contributions is likely to be necessary for each of the various types of services that are brought up on the grid, e.g., computation, storage, data services, visualization, etc. Additional policy and technology principles are covered in the following sections.

Creating a Campus Grid Initiative

There are a number of ways a campus grid initiative can begin. The initiation and evolution on any particular campus will be unique to their set of circumstances, such as the technical and political drivers for a grid, or the size and variety of academic departments on campus. It's not surprising, then, that the campus grid initiatives on the campuses of the authors began in different ways. Several of their experiences are summarized below as models for those who are similarly situated and to illustrate the range of possibilities for getting started.

University of Southern California, High Performance Computing and Communications – At USC, the CTO obtained formal administrative approval to solicit vendor donations in order to build a Beowulf cluster to improve the University's standing in the HPC research community in collaboration with and support of the University's Information Sciences Institute (ISI) and the grid research project group led by Dr. Carl

---

[7] http://www.povray.org

[8] http://www.ticam.utexas.edu/Groups/SubSurfMod/ACTI/ipars.html

Kesselman. To enhance the utility of the Beowulf cluster, the CTO recruited Shelley Henderson, IT Systems Programmer, to act as chief cheerleader for NMI (NSF Middleware Initiative)-related efforts (such as installation of grid, cycle-scavenging, and authN/authZ software) realized by members of the University's Unix systems administration staff. Once the cluster was initially installed, USCGrid was then created to tie together the cluster, a new Sunfire 15k, and a Condor flock. The SCEC (Southern California Earthquake Center) was an early user of USCGrid. The NMI authN/authZ infrastructure initially installed to enable the creation of USCGrid is now in ever-widening use across campus, and central administration now fully supports the infrastructure initiative to overhaul the enterprise directory for better enterprise authN/authZ. All this, properly shouted from the rooftops, is indeed used as a recruiting tool for high-end research faculty.

Texas Tech University, High Performance Computing Center - Texas Tech's Grid initiative was spearheaded by TTU's Information Technology Division Office of the CIO, and implemented by Jerry Perez, Senior Administrator at the High Performance Computing Center. All IT university student lab compute resources were initially dedicated to the project and other departments such as the College of Business, Mathematics, and Computer Science joined the grid several months later after observing the success within IT-controlled labs. Given the high-level directive to get started, gaining buy-in to move ahead with a wider campus grid was not an issue and energy could be focused on the technology implementation and educating the potential user base. Though this may not be a typical situation, it is useful to remember that having or gaining the understanding and approval of an accessible visionary at a high level in the organization may be the "lowest hanging fruit" in some cases.

Georgia State University, Advanced Campus Services - When Georgia State set about deploying a central identity management infrastructure several years ago, their deployment path provided them the *opportunity to work with grids, which helped them come to embrace the new grid paradigm. A number of key elements facilitated the grid paradigm shift, including participation in the NMI Integration Testbed, visionary campus leaders, and an inspired, effective team of grid champions on campus. Tasked with spearheading their identity management infrastructure deployment and integration of middleware for the campus, Georgia State's Advanced Campus Services (ACS) unit built the campus grid initiative on this foundation of collaboration, involving a wide range of constituents from the campus community, as well as individuals from peer institutions and higher education IT organizations such as Internet2. ACS employed several methods to engage the campus community in a grid exploration, including creative use of limited resources, multiple communication forums that focused on understanding Georgia State's potential grid applications, and engaging application users directly in an ongoing exploration of grids. Staff tasked with scouting future IT potential for a University, such as Georgia State's Advanced Campus Services team, are a natural group to begin a campus grid initiative and build the necessary bridges between Central IT, researchers and faculty.*

University of Michigan – UMich's campus grid effort, MGRID (Michigan Grid Research and Infrastructure Development), arose out of informal interactions between various groups on campus including Physics, CITI (Center for Information Technology Integration), Engineering, ITCS, SI (School of Information) and Bio-informatics. A successful project to demonstrate a secure signaling system for network Quality-of-Service (http://www.citi.umich.edu/projects/qos/index.html) prompted the formation of MGRID as a way to foster similar collaborative efforts to advance shared infrastructure. MGRID officially formed in late 2002 with funding from the office of the Vice-President for Research, the office of the Provost and the Deans of the colleges of Literature, Science and the Arts, Engineering, the School of Information, MCBI (Michigan Center for Bioinformatics) and MHRI (Mental Health Research Institute). A notable point in this "grass roots" process is that the need arose from various researchers and departments but then some time was spent to gain more formal support, thereby creating top-down agreement to insure the ongoing availability of resources to succeed.

Texas Advanced Computing Center – TACC has been providing High Performance Computing services to researchers on campus for several years and it was a natural extension for them to drive grid computing activities on the University of Texas campus. However, from the beginning of the UT Grid project, TACC looked beyond the "traditional" HPC user community and involved other stakeholders on campus (resource providers, potential grid users, grid researchers) such as IT Services, the Center for

Weather Research, ICES, and the Computer Sciences department, to define the requirements and direction for a campus grid. Ongoing and close interaction and collaboration between these groups insures that UT Grid meets the needs of the diverse set of communities on campus.

Developing Policies for Sharing

As a new implementation on campus, grids often lack established processes and policies in their start-up phase. This start-up needs to be worked through on each campus, and is dependent on campus environmental factors (e.g., political, administrative, financial). The campus grid lead(s) should use a cooperative approach based on open communication that encourages users to express their concerns. Once expressed, the campus grid team and intended users can work together to co-develop policies for resource sharing and address any grid-related security factors that might arise during implementation.

There are two primary areas where potential grid users may have concerns about using grids. One concern is that users will lose control over their computing jobs or their contributed resources. To convince users to submit jobs and resource owners to share, it is important to insure that grid resources will be available and used in the manner intended. The campus grid team will need to work with resource owners to establish policies about how their resources interact within the grid environment and develop a common understanding about how available tools can be used to implement the intended policies. Since this is an area of high interest and ongoing development in grid technology, it is important to keep up with both the latest standards and the most recent products. In addition, it could be very useful to consult with the Computer and Information Science departments on campus during the policy development and tool selection processes. These faculty and researchers might very well be investigating or even developing related technologies and could provide valuable recommendations towards the most effective choices and use.

A second common area of concern in the use of grids is the belief that computing resources connected via a campus grid pose unique security risks inherent to the nature of a grid. Users may worry that grids are more vulnerable to “attack” and misuse than other resources connected to the campus network - that resources could be compromised or will be unstable, or that jobs and data cannot be protected. The experience of the authors to-date in running their campus grids has led them to conclude that, as a general rule, this is not the case. They have not had to implement procedures to secure their grids beyond standard campus IT security practices that are already required to protect non-grid networks and computing resources. Security issues thus far arise from the same causes as those of non-grid resources, such as lax password procedures, operating system vulnerabilities or insecure networks.

Insofar as grids are implemented with certain specific software, however, that software needs to be configured and secured properly. This can be an issue if documentation for the products being used is not available, complete or accurate, or if systems are being configured by users/owners with varying levels of expertise. Rather than train or rely on resource owners and grid users implementing product-specific set-up procedures, MGRID staff at the University of Michigan set up grid resources themselves. They automate the build process as much as possible with current tools (see “Maintenance and Support” section below), to minimize the time spent. Attention also needs to be applied to the specific means by which users gain access to grid resources. For example, Globus-based grids require that the process for validating a user’s identity be done through the use of PKI (public key infrastructure) and certificates. A related security aspect is in the duration for which certificates are issued. If they are issued on a long-term basis, this provides increased ease-of-access for users and applications but there is also increased potential exposure for the certificate to be compromised, and therefore increased exposure for the resources that accept the certificate as valid. In-depth discussion on recommended management of PKI certificates is beyond the scope of this document and the reader is encouraged to read the NMI Testbed Grid Case Study Technical Supplement: “Authentication & Authorization in SURAgrid: Concepts and Technologies”[9].

---

[9] The *Authentication & Authorization in SURAgrid: Concepts and Technologies* technical supplement: http://www1.sura.org/3000/NMI-Testbed/SURA-AuthNauthZ.pdf

Tools that Facilitate Sharing

Resource owners that choose to make their resources available through a grid want tools that allow them to retain autonomy and control over their resources. The ability to assure resource owners that they can set policy and monitor that usage is in alignment with policy will ease contention over shared resources, particularly if resources are intended to serve their local user base first and the broad grid community second. Tools should allow pre-emption of running or scheduled jobs only under circumstances predetermined by the resource owner. The primary tool for controlling use of grid-based resources today is a scheduler, most of which can be configured to enforce a number of typical policies and prioritization methods. Accounting packages are also becoming available, or are already available in proprietary implementations, where more complex policies can be defined and tied to more detailed reporting mechanisms or, if indicated, billing. A robust accounting package will be able to show usage – who and how much – of individual resources on both a real-time and a historic basis, presented in a standardized format[10].

However, scheduling/accounting packages today are often compute-resource-specific management tools, largely transparent to other elements and processes of campus grid deployment, and may not be necessary in the early states of a campus grid. The grid initiatives at several of the authors' institutions are in early stages where the need for these types of tools is not yet apparent, either due to low usage (so no contention among jobs) or because the grid is based on resources specifically intended to provide useful but open services to inform further campus grid development. In these cases, ready review of active and submitted jobs per node and real-time load statistics may be all the information that is needed to properly monitor the grid. If your anticipated usage of resources is similar, you could consider focusing on other means to build understanding of how resources are shared, such as making test resources available via Globus, exploring identity management via PKI, and exploring the requirements of job submission interfaces from the point of view of end-users.

Ultimately, the specific scheduling/accounting application selected will depend on which technology is being used with a specific compute resource, e.g. Condor for desktop-clusters, PBS[11] or SGE (Sun grid engine) for homogeneous clusters. Various accounting and scheduling methods are in use on the grids managed by the authors, including "home-grown" applications at TACC, TTU, and the University of Michigan. Other mechanisms used include Nimrod-G[12], QBank[13], or simply Node Wall Clock Hours[14] (NWCH). TTU also utilizes product-specific functionality built-into the commercial Sybase Avaki product they are using for cycle scavenging. More about specific implementations can be found through Appendix A: Detail on Authors' Campus Grid Initiatives.

Understanding Applications & User Requirements

To realize the benefits of access to grid-based resources, applications must be "grid-aware" or "grid-enabled" – with functionality in place to take advantage of transparently distributed resources for the purposes of a inquiry or problem. Understanding which applications would benefit from this type of access (since not all applications will) and grid-enabling them can be a very challenging facet of a campus grid deployment. Meeting this challenge will take collaboration and creativity, particularly for applications that were not initially conceived or designed to run in a grid environment. The process will become easier once grid design and support staff, application users, and programmers have been able to gain experience with a few initial applications. Consulting and support along these lines should be available to potential grid users on campus, with some iterative education and interaction required. Users may need to be educated about the benefits and requirements of grids, while grid staff may need to learn more about both the intention and current functionality of users' applications.

Application Analysis

If an application already exists, there needs to be some motivating factor for rewriting it for the grid. As new applications are developed, it's important to understand if the inquiry or problem being addressed

---

[10] For more information, see: http://www.mgrid.umich.edu/projects/accounting.html and http://www.psc.edu/%7elfm/Grid/UR-WG/

[11] http://pbsaccounting.sourceforge.net/

[12] http://www.csse.monash.edu.au/~davida/nimrod/nimrodg.htm

[13] http://www.emsl.pnl.gov/docs/mscf/qbank/

[14] http://www.usc.edu/hpcc/systems/account-resource.php

could be solved more effectively through access to grid-based resources and, if so, what specific functions might be enabled or improved. To motivate application developers and users, consider some of the benefits possible by running an application in a grid environment:

Increased processing capability. Consider a genome alignment algorithm, where two gene sequences are being compared to find similarities. A single CPU processor can complete the job, but using a grid of processors might enable the work to be parallelized – the gene comparison could be split among, for instance, 10 processors that each do a smaller piece of gene-to-gene comparison and communicate their results to one another to produce the final result. While the inter-processor communication adds overhead beyond that involving only one processor, the speed and efficiency of having 10 processors working together offsets this.

Increased speed through simultaneous processing. It may not be a question of speeding up a particular process within an application but, rather, enabling several discreet components of a problem-solving application to run at the same time, e.g., ensemble runs. For instance, at Georgia State University, some graphics students have projects that render a complete animation file by rendering each frame separately, then further acting on those results. If there are 1000 frames, each may take only several minutes, but a single processor must do each one in turn; 1000 frames at just one minute a frame may end up taking 16 hours (depending on the platform). Having access to even 10 processors in a grid could significantly reduce the overall time to render. Increased speed may be especially important where human processing is integrated into the workflow, since people have a hard time staying focused on a task which has long periods of waiting between decision points.

Managed access to unique or highly distributed resources. Grid benefits are not all derived from improved computing power or speed. If an application relies on access to highly distributed resources – large databases, for example – or use of specialized resources (e.g., visualization, instruments), access to these resources can be considerably facilitated from within a grid environment. Grid services such as resource discovery and meta-scheduling can often be used to replace time-consuming, manual and human resource-intensive access. In the case of data access, a grid can provide orchestrated transparent access to distributed sets of information, including verification, version control and similar services. For access to unique resources, a grid may be able to provide network-based access in place of travel to a resource's physical location.

The following list can assist in thinking about the benefits as well as the process for grid-enabling a particular application. While not exhaustive, it illustrates the range of areas that should be considered:

- Understand the purpose of the application and also its current approach and functionality, such as whether it is a serial or parallel application, the sets of platforms it currently runs on, etc. Describe the application flow in conjunction with any specific compute, data or similar requirements to ensure that there are sufficient resources via the intended grid.
- Determine the preferred user application interface, whether or not variations are possible and how the current application interface might integrate with the grid interface or portal. If the user typically uses a browser to run the application, for instance, it might indicate the need to create a user application portal to hide the complexities of running the application on the grid.
- Adaptability and accessibility of the source code is a factor in determining deployment options. If the source code is not available or cannot be modified for other reasons, it could still be a candidate for grid deployment but possibly with limited options, or by using an application adaptor or toolkit (e.g., GridLab's GAT[15]). If the application is constantly being modified, it might not be a good candidate for a portal-based approach or an approach where the application is cached on remote grid resources.
- Does the application have dependencies on specific operating systems, libraries or software packages? This could limit the grid nodes on which the application will be able to run or how many concurrent instances could be active.

---

[15] http://www.gridlab.org/WorkPackages/wp-1/

- Are there specific performance requirements or hard requirements for response times within or between particular steps? This could determine whether features such as on-demand scheduling are necessary, or if network or grid service latencies will prohibit useful results.
- Is there a requirement for advanced features like co-scheduling (running on more than one resource simultaneously), workflow (job consists of several sequential or concurrent steps), or advanced reservations (resources can be reserved ahead of time if it is known exactly when they will be required for an application)?
- How versed is the application scientist in utilizing computer and network technology? If there is low understanding in this area, it's important to include someone in discussions early on who can translate between the "language of the science" and the "language of the grid," to avoid time and effort being spent to grid-enable an application that realizes little or no benefit, or, conversely, not grid-enabling an application that could benefit greatly.

Determining that it would be beneficial to run a particular application on a grid eventually leads to the question of who will rewrite the application code to take advantage of grid potential. Some grid-building teams have taken on the tasks of rewriting code for their researchers, at least initially. TTU used this approach to help their campus grid initiative gain momentum but did not have the personnel available to do this an ongoing basis. Georgia State's Advanced Campus Services took the approach of funding a graduate student to convert code as part of the student's thesis work with her professor. It can be difficult to convince busy researchers, users, or departments that they should allocate time and resources to rewriting applications that are already in use, or to develop new grid-enabled applications to replace these. Two tactics that can help with this are 1) to provide references, templates, tips or training towards understanding grid capabilities and rewriting application code, or 2) to provide more tangible incentives for users to rewrite their code. For example, users could be given higher priority access to particular resources, additional CPU or storage in return for early deployment of their applications on the grid.

Technology Selection

Technologies to implement "a grid" are numerous and growing, available from both the commercial and open-source sectors. This landscape is further complicated by the fact that not all products define "grid" in the same way, or provide all of the functionality that might be expected given one's own definition of a grid. It is beyond the scope of this paper to provide product-by-product comparisons or recommendations; as stated earlier, most detailed examples given within this paper refer to a Globus-based grid, as the dominant technology in use by the authors. However, there some common areas and design considerations to explore in advance to insure that the resulting grid will meet campus needs, no matter what technology is ultimately used for implementation.

- User needs should be articulated into policy statements. These policy statements are likely to center on how resources will be shared and will need to be translated into technical requirements for the grid to succeed. This underscores the importance of beginning the outreach process as early as possible in the grid initiative and continuing to dialogue with users about their expectations and experiences as the grid matures.
- Once user needs have been translated into clear policy statements, the task of researching the appropriate tools and technologies to implement these policies can begin. In selecting specific products, one should try to strike a balance between generic simplicity – typically easier and quicker to implement – and customization, which may be needed to meet the needs of key stakeholders. Given that accounting, scheduling and policy management tools are undergoing rapid development – both in terms of the packages available and the standards that are developing in this space – it is best to look for packages that are modular, providing basic features and reporting, that are simple to use, reliable, and have "hooks" to accommodate campus-specific specifications or able to incorporate future add-ons and upgrades.
- Maintain a balanced perspective. The type of resources that are added to the grid (e.g., dedicated vs. shared, contentious vs. "first come, first served") defines what technology can be used. However, particularly as grid technology is emerging, the grid technology selected also defines the resources that can be used and the nature of the resulting grid.
- There are also practical and support considerations in choosing tools and technologies for a specific campus grid implementation. The number of technical personnel available to support the grid,

including their skill set and the scope of their non-grid responsibilities, is an important factor. The budget available for the campus grid initiative must be considered as well. For instance, if the campus finances are such that there is more money for technology purchases than for staff salary, a commercial grid package may be a viable solution. The package components and vendor support structure behind them could enable the campus grid to be supported with fewer grid staff. Conversely, if there is little money to make technology purchases, but there is a sufficient number of technical staff available, then the grid initiative can succeed using a more “home-grown” approach.

Integration with Central Campus Components

It’s strongly recommended that the following two elements of a campus ID management strategy be integrated into the initial design of a campus grid:

- Existing unique and authoritative identifiers
- Campus Certification Authority (CA)

In addition, if a campus file system is present, it is best to integrate this file system with the file space of the campus grid.

Campus ID Management – Identifiers

Rather than developing a set of identifiers for grid-only access, which can lead to identity overlap (e.g., two departments using the same name spaces or unique identifiers), the consistent assemblage of unique identifiers already in regular use by the campus user community should be integrated into the campus grid. Using the campus’s existing unique identifiers also means that the data from grid scheduling and accounting software will provide the authoritative identities of grid users.

If there are no unique, authoritative campus identifiers, or if there are reasons these identifiers can’t or shouldn’t be used, then unique grid ids are needed for grid users For instance, as a supercomputing center for over 20 years, TACC already had a local account creation process in place before the UT Austin campus-wide ID system was initiated. When they deployed their grid, TACC thus chose to use their established process to generate identifiers for grid users. Though they are now moving toward using the campus ID management system as a method of authenticating new users who request IDs, they will continue to create their own grid IDs. The main reason for this is that TACC is part of several statewide and national grid projects (including the SURAgrid project), and often has the need to create IDs for users who are not part of the campus community. It is more expedient to create these accounts for collaborators by using a locally managed ID system. In the future, TACC may provide users with short-term accounts that are based on “external” (to TACC’s immediate processes) authentication (such as providing a UT ID, having a X.509 certificate from a trusted CA, or having a ticket from a trusted Kerberos realm) and authorization (which individuals or organizations may access which resources).

A slightly different model is used at TACC when users authenticate to a grid portal. A grid portal can be a separate authentication/authorization domain from accounts on the compute resources, for instance. TACC is exploring authentication to a grid portal using multiple methods. One is through use of a campus-wide ID; another is to have a proxy stored in a MyProxy[16] server that is associated with a portal, essentially delegating authorization decisions to the MyProxy server).

Campus ID Management – Certification Authority

In a Globus-based grid, authentication is done using PKI and, therefore, relies on interaction with a Certification Authority (CA). If a campus CA already exists, use of this CA can provide authoritative user identification and also facilitate inter-institutional sharing through eventual connection of the campus grid to grids at other institutions. Even if inter-institutional sharing isn’t a priority for designing the campus grid initially, it is a recommended consideration in the planning process, given the increased emphasis on inter-institutional collaboration in both research and education. If there isn’t a campus CA in place, or if integration is not immediately possible due to policy, organizational, or other issues, grid design should still proceed on the assumption that a campus CA will become available.

While there is no single approach to translate the design criterion of relying on a campus CA into implementation, the methods used by SURAgrid participants follow a common progression. Listed in the

---

[16] http://grid.ncsa.uiuc.edu/myproxy/

order of most desirable first, these approaches either rely on the campus CA initially or assume that integration will be possible in the near future. More detail on each is available in Appendix A: Detail on Author's Campus Grid Initiatives:

- If there is a campus CA available, it should be used as the certification authority required by Globus. The University of Michigan, TACC, Texas Tech University and the University of Virginia are using this approach.
- If the campus has no existing CA, the grid builder should request that the central IT department or those responsible for centralized identity services create one.
- If central ID services cannot or will not provide the CA, the grid-building team can assemble and mirror centralized identification in a campus CA that they create, which will at least be in alignment with authoritative ID management if not eventually becoming an integral component of it. Georgia State University and the University of Alabama are using this approach. The University of Southern California uses a somewhat hybrid approach in that, as "central IT," the USC grid building team (from the Information Services Division, or ISD) has control over Unix user IDs, which are used as de facto enterprise IDs on the computer network. They created a CA to issue both host and persistent user certificates. User certificates are issued by the KCA (Kerberos Certificate Authority) through KX.509[17]. Having control over both the campus network and the Unix logins enabled ISD to enforce a single namespace, which has proven to be very useful for system and process integration.

A number of on-line resources provide information regarding the establishment and use of a campus CA. Some sites providing information to get started are:

- http://www.openca.org/
- http://middleware.internet2.edu/hepki-tag/opensrc.html
- http://middleware.internet2.edu/hepki-tag/#PKI_Lite

The links below provide additional detail towards the general use of PKI certificates and their use in conjunction with a Bridge CA in a dynamic or inter-institutional grid environment. The Bridge CA documentation includes best practices in the use of CA profiles as developed by institutions implementing inter-institutional resource sharing through SURAgrid:

- https://www.pki.virginia.edu/nmi-bridge/
- http://www1.sura.org/3000/NMI-Testbed/SURA-AuthNauthZ.pdf

Campus File System

One of the first issues campus grid implementers encounter, after a basic grid infrastructure is put in place, is how to get access to data and applications on grid nodes. Grid users typically have to stage-in any needed data, libraries and executables along with their job to be able to run their applications on a grid. If a campus grid has access to some type of distributed file system, the users job is much easier. Typically NFS and AFS are used to provide a network (shared) file system. A basic installation of a shared file system provides a common space accessible from both a grid gatekeeper and its worker nodes. Thus jobs can stage input directly to the gatekeeper with the expectation that the staged data will be accessible when the job begins execution. Both grid job inputs and outputs can be passed via this type of configuration. At the next level, the shared file system can span larger domains - between two or more grid clusters or possibly a whole campus. If users' grid and campus identities are coordinated, they can directly access any needed inputs from, and store output directly to, the campus network file system. Such a configuration can make grid job definition as well as execution much simpler and more effective.

The authors have different experiences with campus file systems and outline some of them here:

- *Texas Advanced Computing Center - TACC currently uses NFS and has evaluated, but are not currently using, Avaki. TACC production clusters use GPFS and Ibrix (parallel file systems, used only within a single cluster).*
- *Texas Tech University – TTU uses NFS Unix, Avaki Data Grid, and Windows CIFS. The Avaki Data Grid is used to expand grid storage capabilities as needed and scales nicely. To add more space to the data grid, one simply has to join another storage server to the grid via the Avaki Grid middleware.*

---

[17] http://www.citi.umich.edu/projects/kerb_pki/

- *University of Michigan – The University of Michigan utilizes an "institutional" file system to support their users. Currently this campus-wide file system is AFS, housed in the umich.edu AFS cell. All users are given a "home" directory space in AFS. Many individual departments and groups also deploy NFS (V2, V3 and now V4) to support their local data access and storage needs. Both NFS and AFS have distinct advantages and disadvantages in comparison with one another. NFS performs better than AFS but lacks the security and global namespace provided by AFS. AFS, while secure and integrated with our campus user authentication, doesn't provide very high IO bandwidth and requires special modifications to allow batch system or grid users to obtain their AFS tokens to allow their jobs access to the file system. Work is underway to leverage NFS V4 and create a "GridNFS" which is an NMI project funded by NSF and will help to address the shortcomings of both NFS and AFS (see http://www.citi.umich.edu/projects/GridNFS).*
- *University of Southern California – USC uses NFS because it enables user home directories available on the Linux cluster. Home directories are all on QFS. The temp directory on the Sunfire 15k to be is NFS, whereas the Beowulf cluster uses PxFS for scratch and temp. There is a discussion related to this at: http://www.phptr.com/articles/article.asp?p=169687&seqNum=5&rl=1.*
- *University of Virginia – UVA has used NFS on campus since the 1980's. Once the campus grid gets where it needs to be (production-ready), the plan is to utilize a "grid file system".*

Motivation for a "Grid File System"

One of the first tasks any grid user must undertake is to provide the inputs for their grid applications. This may include configuration (run-control) files, input data files as well as the actual executables themselves. In the absence of a common file system, such inputs must be staged from some remote site to the job execution node. Following the completion of the application the job outputs must also be moved to some specified remote storage system. In both cases users must move files so that they are accessible, either as inputs for the task or as outputs for analysis or use. This is typically done with an application like gridftp which can securely move files between gridftp capable nodes. If the job execution node is not also a gridftp node, the assumption is that the gridftp area used to stage grid files will be "visible" to the job execution node. If not, an additional step must be taken using the job scheduling environment to move the files to/from the job execution node.

How much simpler it would be if there was a file system visible from any job execution node on the grid (along the lines of the more pervasive network file system discussed above), which utilized grid security and was able to deliver high-performance across the wide-area network. Having such a system would obviate the need for any data staging since inputs could just be "pointed" to by their path in the file system and outputs could just be put where they were intended to reside. File system semantics would be available as well, allowing users to open, seek and extract just the portions of the input files relevant for their task. In many cases this could result in significantly reduced bandwidth usage and a corresponding speed up on job execution. For example, consider the case where only 10% of a 100 Gigabyte file is needed for a given job. Without some form of shared file system, the user would need to first move 100% of 100 Gigabytes before the job could even start. By having a shared file system in place, the user could immediately begin execution and would only have to access 10 Gigabytes thru the network to complete their job. Another advantage is that this type of capability allows access to datasets, which are much larger than the available local storage—multi-terabyte datasets could be accessed from clusters with minimal local storage. The bottom line is that some kind of campus file system is helpful in making campus grids easier to deploy and use.

Identifying & Adding Resources to the Grid

The type and order of resources to be added to the campus grid should ideally support the most ready or strategically visible applications. Compute resources are typically added first since compute-intensive applications are more obviously positioned to benefit from the multiplication of capability made possible by grid technology. However, on any given campus, other types of applications may be more compelling as initial application targets due to more obvious need, increased readiness to deploy, or the importance of deployment to institutional goals. Resources to support these applications may include visualization, high-capacity storage, data services, or access to unique or distributed instruments. It is also inevitable that some resources will be easier to deploy on a grid than others, either because the technical requirements for implementation are well-known and easier to meet, or because the policies surrounding the use of the resource are clearer and easier to implement given current technology.

Examples of the variety of resources and applications that can be supported include:

- Southern California Earthquake Center, http://www.scec.org/
- NEESgrid, www.neesgrid.org/
- TTU VIZ, http://www.hpcc.ttu.edu/visualization.html
- University of Michigan Life Sciences Institute, http://www.lifesciences.umich.edu/institute/index.html

If you can't find existing campus-based resources to support initial applications, be creative in seeking resources elsewhere. In several cases, the authors' grid teams have purchased new equipment that is under their direct control in order to expedite the deployment of resources to support target applications. Most have also carefully surveyed their campus to uncover resources that are under-utilized (e.g. public labs) or even surplus (abandoned or retired from previous projects, for instance) and given these resources new utility through their availability on the grid. In addition, the authors are all gaining use of resources at other institutions through their participation in SURAgrid.

The actual process for adding resources to a grid varies according to the grid technology in use and the type of resource being added. The following sections discuss the addition of compute resources within the context of a Globus-based grid. Adding resources other than compute resources (e.g. data services, visualization, instruments) will vary more depending on what the resource is. Many of these resources - telescopes or microscopes, for instance – aren't as common or well defined and may take longer to grid-enable or the grid-enabling process may be proprietary or highly dependent on the technical specifications of the device.

Adding Compute Resources to the Grid

A compute resource on a grid must provide a set of grid services such as user authentication and authorization, job and data management, and resource monitoring and management. Compute resources might be added to a grid in two ways: – as *dedicated* compute clusters running High Performance Computing applications, or as *non-dedicated* resources such as desktops, workstations or shared clusters. There are different options for adding each type of resource to a grid.

Compute clusters are the most prevalent type of compute resource available on a campus today. In addition to clusters belonging to IT Services and Supercomputing Centers, a typical academic institute has several clusters owned by individual researchers and departments. These clusters are usually already configured with local resource managers such as PBS, LSF, or LoadLeveller that provide basic job queuing and parallel job execution support for the cluster. If a cluster is not already configured to run HPC jobs, there are several free cluster deployment and management packages available that are simple to install and configure such as:

- Rocks Cluster Distribution[18]
- eXtreme Cluster Management Toolkit (xCAT)[19]

The simplest and most non-intrusive method of adding a cluster compute resource to a grid is to install and configure the Globus Toolkit to provide the basic grid services to enable researchers to remotely submit jobs for execution. There are four basic services that are required for a compute resource in a grid:

- Security services to provide authentication and authorization of users requesting access
- Information services to provide discovery and monitoring of grid compute resources
- Data services to access and transfer data to and from a compute resource
- Execution management services to enable a user to launch jobs remotely on the resource

The Globus Toolkit supports robust, high performance, secure data movement services and has job managers to support job submission to remote clusters that are managed by a variety of local resource managers including PBS, LSF, LoadLeveller, Sun Grid Engine, and Condor. There are several packaged solutions to help install Globus Toolkit services on a compute resource. The NSF Middleware Initiative Grids package (NMI-Grids, http://www.nsf-middleware.org) contains a set of tools including the Globus

---

[18] http://www.rocksclusters.org/Rocks/

[19] http://www.alphaworks.ibm.com/tech/xCAT

Toolkit, and provides support for installation and configuration of these grid services. The Virtual Data Toolkit (VDT, http://www.cs.wisc.edu/vdt//index.html) is another widely used collection of grid middleware that is easy to install and configure. The IBM Grid Toolbox V3 is a free software package from IBM (http://www-1.ibm.com/grid/solutions/grid_toolbox.shtml) and includes a wizard-based installation that makes it easy for the grid administrator to install and configure the Globus Toolkit. These grid packages can often be installed as part of the cluster software installation. For example, the NMI grids components can be installed and configured along with a Rocks install by using the “Grid Roll” option available in Rocks.

If a campus grid has specific requirements, it might become necessary to diverge from a simple package installation in order to address the needs. In most cases, however, these packages are adequate to be used as is. In all cases, the grid packages tend to evolve significantly approximately every six months, thus requiring an upgrade on all grid resources. Some solutions like VDT address this issue by providing support for automatic software upgrades.

Universities usually also have a large set of desktop and workstation resources available in labs, student dorms, and even faculty/staff offices that are typically used for activities such as Internet browsing or user email. These machines are typically idle much of the time and may have significant untapped compute capacity even when they are in use. There are several grid software solutions available that harness the unused cycles on such machines and make them available to users to run their computations. Grids built from these spare cycles are usually suited to running specific types of applications that do not require high-bandwidth, low-latency communication between the application processes. Further, the applications must be tolerant to node failure, since these non-dedicated nodes could be shut down at any time by the desktop owner. Examples of desktop grid solutions include Condor (http://www.cs.wisc.edu/condor/), United Devices (http://www.ud.com), and BOINC (http://boinc.berkeley.edu/). These desktop grid packages are usually simple to install and make operational, and offer very low cost alternative for entry into grid computing.

Resource Registration, Scheduling & Monitoring

An accounting or scheduling package shouldn't be considered an absolute necessity for making resources available on a campus grid in the early stages. Keep in mind that a scheduler, in this context, is an internal resource management tool and is largely transparent to all the other elements of a campus grid deployment. The specific scheduler used will ultimately depend on what technology is used in a specific compute resource, e.g. Condor for opportunistic desktop-clusters, PBS and SGE for homogeneous clusters. If the anticipated usage of resources is intermittent or low and it is unlikely that jobs will be competing for resources at the same time, consider directing efforts to other parts of the infrastructure for the campus grid. Focus on the issues of making test resources available via Globus, identity management via PKI, and job submission interfaces for end users in designing the campus grid infrastructure.

Most Grids include a mechanism to monitor all grid resources to provide the user or application with the ability to choose the best resources for their job, based on the availability, load, and type of the resource. Monitoring packages also provide information on the status of jobs submitted to the grid, to help the user manage their jobs. Monitoring becomes significantly more important as the grid grows to contain larger numbers of resources and supports more users and jobs. There are many monitoring packages such as Ganglia (http://ganglia.sourceforge.net/), MonALISA (http://monalisa.cacr.caltech.edu/), Gridcat (http://www.ivdgl.org/gridcat/home/), and the ACDC monitoring dashboard (http://osg.ccr.buffalo.edu/). These packages are often included in the grid middleware packages such as VDT and NMI-Grids. Commercial products are also available, but they are typically costly or are included in a vendor grid package and may only work with that vendor’s grid products.

For large campus grids, it is very useful to have a metascheduler to provide resource brokering services for user jobs rather than place the onus on the user to select the “best” resource for a particular job. There are several resource brokers available today but none is really robust and general enough to be deployed in a large campus setting. This is a growing area of interest, however, and it is useful to stay aware of the current state of the art in order to evaluate and plan for future deployment. Some examples of metaschedulers include:

- Condor Matchmaker
- Gridlab Resource Management (GRMS, http://www.gridlab.org/WorkPackages/wp-9/)
- Community Scheduling Framework (CSF, http://www.globus.org/grid_software/computation/csf.php)
- Nimrod/G (http://www.csse.monash.edu.au/~davida/nimrod/nimrodg.htm)
- MARS (http://www-personal.engin.umich.edu/~abose/website/mars.htm)

Most sites, including the University of Michigan, TACC, and Texas Tech University, use home grown accounting packages to track resource usage by campus users. The UMich package used in their MGRID campus grid is publicly available in the NMI package as a superset of PBS/XML. Other sites such as USC use packages such as Qbank for Solaris resources and NWCH for Linux resources. With the proliferation of grids, these accounting packages must evolve to support special usage scenarios in some grids. For example, the Open Sciences Grid (http://www.opensciencegrid.org) maps all users from the same virtual organization into a single user-id at a local site. In this usage scenario, user-ids no longer uniquely identify a user, and the accounting system must be modified to use additional information to identify the user.

Building Critical Mass

While campus grid initiatives may begin in different ways, all need to build a critical mass of users, applications and resources in order to become a sustainable institutional asset. To accomplish this, design and planning should be coordinated among relevant stakeholders as early as is practical.

For successful and efficient alignment with other campus information tools and processes, some key functional areas need to be represented right from the start (also see section above, “Integration with Central Campus Components”). These include central ID management, campus network management, and staff that provide computing support for researchers on campus.

Current and planned grid users should also be actively involved to insure that the resulting grid resource meets the broad needs of the campus. Formalizing a collaborative forum such as a grid users group, or grid user representation added to existing IT decision-making processes, can help increase interest and build critical mass by demonstrating to users that the grid is intended to evolve in response to their needs. Forums of this nature are part of the grid initiative on the campuses at Georgia State[20], Texas Tech, and TACC[21]. As grid staff and end-users gain experience together in deploying and using applications on the campus grid, more real-world examples of the benefits can be seen and the recruiting of new grid users becomes easier over time.

Outreach Techniques

The ultimate goal of outreach in a campus grid initiative isn’t merely to increase the number of grid users but to ensure that the resulting resource is broadly useful and provides a return on investment – across research domains and various user groups, e.g. students as well as faculty. A program that supports multiple opportunities for collaboration and the open exchange of ideas and experiences among these groups and with the grid-building team should be a foundation of the outreach effort. Georgia State University’s multi-faceted outreach strategy has proven successful in promoting and growing their campus grid initiative, with the grid-building team acting on the following principles:

- Strive to meet mutual objectives of faculty, researchers and IT staff. Common objectives often center on securing project funding, publication of work, and a desire to expand student learning opportunities by including them in projects whenever possible.
- Both grid builders and potential grid users should reach beyond the campus to find peers benefiting from grids in order to increase their opportunities for, and exposure to, new ideas.
- Actively participate in collaborative environments - volunteer, be enthusiastic and work to develop cooperative approaches to common problems.
- Grow relationships through personal contact, using both one-on-one and group discussion formats.

---

[20] The GRID Group @ GSU: http://www2.gsu.edu/~wwwacs/Grid_Group2.htm

[21] UT Grid Group: http://utgrid.utexas.edu

- Use existing campus communication tools (e.g., newsletters, committees, websites), and develop new tools if existing methods are insufficient (e.g., a grid users or steering group).

It's important to utilize the organizational and communication structure when reaching out to researchers and other campus community members to understand of the value of the campus grid. For instance, outreach work by the staff of the campus grid building effort at TTU was more effective because they worked within the campus political infrastructure, which included strong support from the CIO. Jerry Perez, Senior Administrator in TTU's High Performance Computing Center, was able to leverage this backing as he approached resource owners and potential users to join the campus grid-building effort.

Anticipating and Addressing Common User Concerns

Conversations with potential grid users about the benefits a campus grid could have for their applications and research may not produce immediate results. By developing an understanding of the issues involved, staff working on the campus grid initiative can better prepare for discussions they are likely to have with various campus community members.

While grid middleware has been rapidly maturing and is in production use in many organizations, the grid builder is likely to encounter those who consider grid technology to be immature or too cutting edge, or feel that changing their current approaches would be too time-consuming or intimidating. The value of grid technology for use with particular applications may need to be analyzed or clarified to a more specific degree, in terms that can be understood by the owner or users of the application.

Some potential users may feel that the independent clusters of workstations they currently use are, in fact, grids themselves and, moreover, "all the grid" that the applications need. This view is often derived from an earlier view of grids, where individual computers are linked together in clusters to act in concert on the same problem. In more recent usage, a grid is commonly considered to be a strategic infrastructure that provides similar coordination for more widely deployed resources that are shared across departmental or even organizational boundaries. Once this definition is understood, the details of how the campus grid might provide additional capability for the applications in question can be discussed, including any necessary technology or policy-based details for deployment.

It's common for campus researchers to be involved in funded projects where funds have been allocated in the project budget for computing hardware. When these researchers acquire equipment they deem sufficient for their needs (including perhaps their own isolated grid), the job of convincing them to put their resources onto the campus grid and share them is particularly challenging. The best arguments in this case relate to scaling – that the model of "each acquiring his own" is expensive and difficult for any one researcher or team to justify or sustain. As importantly, it can be argued that the expanding community of users with access to grid-based resources and the knowledge and expertise gained can be a strategic and competitive advantage for a campus today.

Additional points to help address user concerns or highlight grid benefits include:

- Grid users gain access to peak, high-burst mode computing, which may not be available otherwise.
- Resource owners who are also grid users can be provided with accounting and scheduling services that enable them to track the usage of grid resources. In addition to monitoring usage of their own resources, they'll be able to see the value they've gained from the use of others', hopefully promoting the perspective that they have "gained a grid" versus losing some portion of an individual resource.
- Tools can be added to a grid to assure users that their expected access (e.g., priority scheduling, a specific time or amount of time/processing capability) was indeed provided. Tools such as backend schedulers and the use of job pre-emption can monitor and enforce usage policies.

Prototyping and Demonstration

No matter who your audience is, there are few better ways to champion a new concept than letting potential users see and, ideally, interact with, a working example. Developing a grid demonstration project for presentation within various user forums can be a valuable tool for promoting the grid to potential users. The steps below outline a common grid demonstration scenario that the authors have found to be understandable and convincing for a range of audiences. Except for the fact that a grid is providing the assemblage of resources, the work of scientists in many disciplines resembles this process. Therefore, the scenario can be customized to illustrate a likely grid experience within a variety of scientific fields.

Grid demonstration scenario:

1. Access the campus grid via a web-based portal
2. Show a list of available resources and the ability to monitor activity and performance
3. Log-in with an authoritative ID
4. Set the grid environment for the specific use or application (e.g., home and library directories)
5. Submit, execute, and run a job
6. Move job results to a visualization machine (e.g., via GridFTP)
7. Visualize the data

To better convey the value of grids to researchers, a grid demonstration should ideally feature “the researcher’s science” (e.g., molecular manipulations for chemists or biologists; particle simulations for physicists; computation fluid dynamics for mechanical engineers…). Expanding the research applications in the grid builder’s demonstration portfolio is something that should become easier over time, as more users join the grid and enable their applications (see section “Identify Application & User Requirements”) for use in the grid environment.

When deciding which applications in particular to use in a domain-specific demonstration, it’s also useful to keep in mind that some researchers’ work predominantly involves use of existing applications within their scientific domain whereas others are focused on developing or improving those applications, with frequent modifications and testing. Well-known and relatively “static” applications are best to use as demonstrations initially to reach the broadest audience possible and to avoid the need to re-verify and debug the application repeatedly over time.

Finally, though it is crucial to establish a test implementation when building a campus grid initially, it is also very valuable to maintain a test grid going forward. This test grid can be used to harden the set of technologies and processes supporting growth of the campus grid, to evaluate or develop new features and services before introducing them to the broad user base, and to provide a means for potential grid users to learn about grid technology before deploying their applications. A test grid can also be used to provide more dynamic or open “guest access” to demonstration applications, avoiding the more formal and rigorous accounting and security set-up of the operational campus grid.

The Production Campus Grid

When to Call it “Production”

For the purposes of this paper, an appropriate definition of a “Production Grid” is a grid that meets the functional expectations of both the grid builders and the potential grid users. According to Peter Kunszt, Gavin McCance, Krzysztof Nienartowicz, and Akos Frohner[22], these expectations are met through clearly identifying grid partner sites with a high level of commitment, adhering to the support of mutually acceptable standards, and using a clear process of educating and accepting new Virtual Organizations as partners who will add more resources to the grid and users who will use these aggregate resources.

Production grids primarily possess the following characteristics:

- Fault-handling: error-recovery and error reporting mechanisms in place
- Reliability: possessing a high level of stability and availability
    - Provide a high availability of grid resources
    - Insure that grid components are very robust, especially with respect to network outages
- Security: standards defined and in place, satisfying all grid partner sites
- Interoperability: flexible enough to ensure data exchange and integrity across all grid resources

---

[22] Peter Kunszt, Gavin McCance, Krzysztof Nienartowicz, Akos Frohner,
Grid Data Services for Production Grids
<http://www.nesc.ac.uk/events/GGF10-DA/programme/%20papers/DataAreaArticle-GGF10-final.pdf> Global Grid Forum 10, 2004
http://www.nesc.ac.uk/events/GGF10-DA/programme/papers/DataAreaArticle-GGF10-final.pdf

    - Able to support a wide range of executable requirements
- Usability: have a set of one or more highly developed grid application
- Performance: have an acceptable level of performance
    - Able to provide an acceptable Quality of Service across the grid-system
- Flexibility: have the ability to change and evolve with grid standards and software

Keeping these requirements in mind, once initial services are running reliably on the campus grid and support for these services can be met at a production level (see following), then grid organizations may begin offering their campus grid as a production service. Given the current state of grid technology, below are services that a production campus grid should be able to offer initially and in the near future.

Current vs. Future Services

Many grid services are developed and stable to the point where they can be offered as part of a campus grid in a production environment. What is required versus optional for one campus, however, may be different for another, and it is possible to go into "production" with some services while monitoring the maturation of available technology for others. The following are examples of services our authors are offering with their production campus grids:

Texas Tech University - Texas Tech's production grid services are the use of lab machines for short-running cycle scavenging and the use of clusters for scheduled, long running job submissions. Another grid service is the use of the Sybase Avaki Data Grid to share information between researchers, their compute resources, their executables, and their collaborators. TTU has various independent vendor grids and in-house developed grids with Condor and Globus running as well.

University of Michigan - MGRID is charged with prototyping a possible production grid environment for the University of Michigan. MGRID's role is to research grid technology. Once tools and services are developed to a certain level of reliability then these are turned over to central IT. There are a number of areas MGRID has focused on to enable such a prototype infrastructure:

- *MGRID portal services – http://www.mgrid.umich.edu/projects/portal.html. Secure web-based portal using CHEF interface and authenticating with kx509, supporting the following services:*
    - *Job submission*
    - *Accounting information reporting*
    - *Upload and download to/from user's machine*
    - *Grid FTP between clusters*
    - *Bioinformatics interface providing BLAST search capability*
- *MARS - a inter-cluster metascheduler; publicly viewable web interface for viewing MARS status; secure web access for viewing your jobs*
- *Collection of accounting data following GGF URWG standards (PBS XML based accounting package)*
- *Identity mapping via Walden, fine-grained authorization via Walden, implemented with XACML. Also includes Solon - a GUI for editing policy files as input to Walden*
- *NTAP - Network Testing and Performance secure diagnostic system based upon grid technologies.*
- *Common file system using NFSv3 (migrating to NFSv4 soon), 1/2 terabyte grid-wide accessible storage, and "gridNFS" to leverage NFSv4 for the grid.*
- *NetLogo simulator language*
- *g98 - Gaussian quantum chemistry package*
- *GAMESS - General Atomic and Molecular Electronic Structure System is a general ab initio quantum chemistry package; more MAC nodes; and upgrade from CHEF to Sakai*
- *Providing pre-built gatekeeper nodes to enable university departments to easily join the grid*
- *MGRID CA (certificate Authority) for making gatekeeper/host certificates*
- *Consulting services for grid-enabling custom applications*
- *Documentation for using MGRID services*

Texas Advanced Computing Center - TACC offers a set of production level services that include customized user guides, user training workshops, consulting and support for users and resource providers, and an allocations process for users to request accounts and access to resources. TACC *production compute services provide access to large compute clusters for traditional parallel computing jobs, as well as high throughput desktop grids, Rodeo (Condor-managed desktop grid, and Roundup (managed by the United Devices Grid MP software). The UT Grid website (http://utgrid.utexas.edu) has detailed documentation for users, resource providers, and developers. TACC offers a user portal and currently offers its users:*

Web site with project information
- *Account and allocation request processes available via user portal*
- *Site Certificate Authority to issue grid certificates*
- *User support (system status information, problem reporting system)*
- *Online documentation (about all resources; tools such as the user portal)*
- *Education/outreach: training courses on using grid resources; user forum meetings*
- *Grid user portal supporting:*
    - *job submission and management*
    - *reliable data transfer*
    - *single sign-on & proxy management*
    - *resource monitoring*
    - *application portals for specific applications of such as NAMD, POV-Ra*

University of Alabama at Birmingham - UAB is planning for production services and offers a perspective on when a campus grid is production ready: The campus grid is ready to be in production as a whole when the primary interface for job submission is via the grid standards. This is distinct from certain HPC resources, which may already be production resources when they are made available on the grid. In effect, production state for the campus grid will be when it is "the" interface for interaction with the campus resources.

There are several additional services that would be valuable to include on a production campus grid but are considered by our authors to be fairly far in the future for implementation. These include:
- Metaschedulers, with accounting attuned to an overall view of the grid as well as granular views by resource, virtual organization, or user
- NFSv4 file system
- Resource advertising and discovery
- Verification testing for authorization policies
- Improved cycle-scavenging to support long-running jobs (better job suspension)-
- Portlets customized to support specific applications
- Interoperability across grid products to realize a “grid of grids,” where one can submit grid jobs via one type of grid middleware and the job may be submitted transparently to another grid using different grid middleware, if resources are available

User Interfaces

From the perspective of a Globus “fresh-out-of-the-box” user experience, user grid interfaces are minimal. Currently, a user has to determine from among available resources where their job should run. This can be a difficult and error-prone process, particularly if the user is not knowledgeable about various machine or performance characteristics. (Indeed, this is why metaschedulers and portals are being built. Metaschedulers in particular are also good for coordinating resources on disparate machine configurations.) To reduce the technical expertise that is required from the grid user, and help insure that jobs run as effectively as possible, most grid implementers end up developing a more user-friendly grid interface – or grid portal – to serve their user community. Several grid portal approaches are in use on the authors’ campus grids:

- *University of Southern California – For most users it's all command-line with nothing pretty to look at. However, the SCEC project is developing a web-based portal that will be available only for their project.*
- *University of Michigan – Portal interfaces are provided. Appendix B shows several - all from real, running programs (except for MARS which is a mock-up.)*
- *Texas Advanced Computing Center – The main web sites (http://www.utgrid.utexas.edu and UTGrid user portal https://portal.tacc.utexas.edu) have links to resources, user guides, news, training,*

consulting, reservations & allocations. Note that a user may have to be logged into the portal for some of these features to be enabled.

Maintenance & Support

Maintenance of a grid includes "care & feeding" as well as understanding of failures experienced in creating and/or running support structure. Maintenance of grids is, of course, a necessity for offering production services. The approaches vary across the authors' sites, though the trend is clearly to integrate grid support services with the overall IT infrastructure support of a campus.

The University of Southern California has 2-1/2 FT employees who are responsible for supporting not just the grid but all of the high-performance resources. Grid maintenance and support responsibilities have simply been added to this existing support infrastructure.

The support model for the Texas Advanced Computing Center begins with hardware resources to host basic grid services such as the user portal, certificate authority, resource broker/scheduler, etc. (in addition to the actual compute, data, visualization and storage resources that will be part of the grid). Staffing depends on the size and complexity of the grid (number of users, number and type of resources, frequency of jobs). Assuming that normal maintenance and support of compute resources, etc. are accounted for separately (since they would already be part of the IT services), TACC has found that it should not take more than 1 to 2 full time employees to support and maintain a small-to-mid size campus grid. The expected duties of grid support staff include installation and maintenance of the grid software components; monitoring the grid to see if there are problems; handling requests for new grid accounts and certificates; and handling user problem tickets. This does not account for additional support that might be required to help users get new applications deployed on the grid, develop additional grid or grid application tools as required, etc. TACC's approach also underscores that it is very desirable to integrate grid services into the existing services that support high performance research computing on campus.

The production model at Texas Tech University requires people resources dedicated to the project and compute resources to monitor and manage the grid. When people fail to commit to take care of their grid zone, then nodes in those zones can be unreliable and affect grid performance. TTU's TechGrid involves a centralized IT grid authority, such as a Grid manager, and local Zone Administrators to manage and maintain grid resources in remote regions. TTU seeks to integrate grid support with campus IT infrastructure support. Indeed, local "zone" grid administrators will know well before most people that their resources may be having problems and cannot participate in the grid for whatever reason. The Zone Administrators cooperate with the central Grid Administrator as a key to offering a truly campus-wide grid. TTU has also created web-based grid monitoring tools to allow users to view and select grid resources that meet their computing needs. Specific responsibilities of TTU grid administrators are:
Zone Administrators duties (2 hours a week):

- *Installing nodes: A script has been provided.*
- *Uninstalling nodes: A script has been provided.*
- *Configuring nodes: A script has been provided.*
- *Testing nodes: A script has been provided.*
- *Re-imaging: This is a standard ATLC function; however nodes need to be uninstalled before re-imaging takes place.*

Emergency procedures:

- *Grid Failures: If the grid goes down without warning, then the next step will be to disable grid system services on each machine (This is a rare occurrence). It is the zone administrator's duty to inform the Campus Grid Administrator when issues like these arise so that a remedy can be applied immediately.*

Campus Grid Administrators duties (20 hours a week):

- *Maintain the Bootstrap server: create scripts to monitor grid usage and failures.*
- *Train Zone Administrators.*
- *Write scripts to add functionality to the grid.*
- *Train users.*
- *Find more resources to add to the grid.*
- *Help Zone Administrators with grid related issues.*

- *Help students/researchers grid-enable their code.*
- *Installing nodes: Install nodes into new grid zones.*
- *Uninstalling nodes: Help new grid zones uninstall at first re-imaging.*
- *Configuring nodes: Help new grid zones configure their nodes.*
- *Testing nodes: Create scripts that test the availability of a node.*

Emergency procedures:

- *Grid Maintenance: If the Campus Administrator knows when the grid will go down with enough warning, then the grid can be gracefully unmounted using an elegant shutdown script that will unmount each individual node in the grid without affecting quality of services for the end users. Zone Administrators will be told of the grid shutdown in advance. Grid Zone Administrators will be asked to reset their compute nodes at the end of the day to reactivate the grid on those nodes.*
- *If an emergency shutdown is required, then the grid will be shutdown without dismounting worker nodes. This case is rare since this type of failure is caused by circumstances beyond Grid Administration control such as power or chiller failure at Reese Center. Emails will be sent to Zone Administrators.*

The University of Michigan notes that the day-to-day maintenance is very small, though there can still be “scare” events at times, such as a gatekeeper going down and not having been not properly backed up, requiring a couple of days to rebuild. They note the following “gotchas and experiences” for added perspective on what grid maintenance and support can entail:

- *NFSv3 is not perfect but it was the best choice given the alternatives. Currently we do not have a way to restrict authorized people from doing bad things with storage. We recently had a user accidentally fill the entire disk on a gatekeeper and it took the system down. There does not seem to be a problem with a user deleting another's data, but there is nothing to stop it at this point.*
- *Initial installations were difficult, aggravated by security requirements. Getting the correct list of certificates to provide the desired chain of authentication was hard. These problems are now solved.*
- *The kx509 portal interface requires installation on the user's system. Keeping pace with different operating system versions (Linux, Windows, and Mac) and the variety of ways they can be configured is a problem. We have sometimes resorted to enlisting MGRID personnel to make the installations for the users.*
- *Researchers often develop their own code, and want to modify, build, and run it. We don't have a "grid" way of providing this capability. Currently, we do this by providing them an account on the head node and let them build their software via ssh.*
- *Static programs (such as BLAST) are easiest to make available. It also can be easier for all involved to make a customized portlet rather than having users use the more generic Job Submission portlet.*
- *Some users have programs that have the potential to be implemented in parallel, but doing so can be a lot of work. If users wrote the program themselves then they are often open to modifying it, but will not necessarily put forth the effort or have the expertise to do so. If it is written by a third party then no one wants to take up the task, or cannot because of lack of source code or licensing restrictions. The best success here has been situations where the program is "embarrassingly parallel", which means that the problem can be divided into sub-tasks that have little or no interdependencies.*
- *It is important to have a detailed technical discussion with potential users (of new applications) before trying to move their application to the grid to determine if the effort would be worth it. There have been cases where a user can simply run their program on a PC overnight and get the results they need rather than putting lots of work into making it run in parallel.*
- *Some programs cannot be made to run in parallel and must run a long time (up to a week). Our cycle scavenging clusters (running PBS) do not properly restart applications, so we can only run them on dedicated processors.*
- *We have not yet tried scavenging from Windows nodes. There has been very little interest. Conversations with Altair (PBS company) indicate that it is less stable.*
- *There are a number of systems on campus that are dual boot Windows/Linux, which makes cycle scavenging more difficult. We have not even tried to incorporate these as worker nodes.*

Conclusion

This paper describes the work of participants within the SURAgrid initiative who have been instrumental in the development of campus grids at their universities. SURAgrid is continuing to expand from the collaborative exploration of grid technology begun in the NMI Testbed Grid, a sub-project originating from SURA’s management of the NMI Integration Testbed Program. Campuses working within SURAgrid are leveraging existing authentication/authorization methods, deploying portals, and developing components that enable campus researchers to more easily realize the benefits of grid and cooperative computing. Grid technology is being developed and deployed from the perspective of a higher education research campus where grids are integrated components of existing campus infrastructure. Rather than a homogeneous environment, SURAgrid participants represent a range of grid maturity, various infrastructures and different computing resources, and are benefiting by collaboratively building a scaleable, interoperable, shared grid. This paper will hopefully provide other organizations that are contemplating or proceeding with the deployment of grid technology with added insight and perspective towards building an effective and efficient campus grid.

Appendix A: Detail on Authors' Campus Grid Initiatives

Georgia State University
Shifting the Grid Paradigm with NMI, http://www1.sura.org/3000/NMI-Testbed/GSU-GridParadigmShift.pdf, part of the NSF Middleware Initiative (NMI) Integration Testbed Case Study Series*, May 2005.

University of Michigan
How Grid Science and MGRID are Changing Research and Education, http://www1.sura.org/3000/NMI-Testbed/UMich-MGRID.pdf, part of the NSF Middleware Initiative (NMI) Integration Testbed Case Study Series*, January 2005.

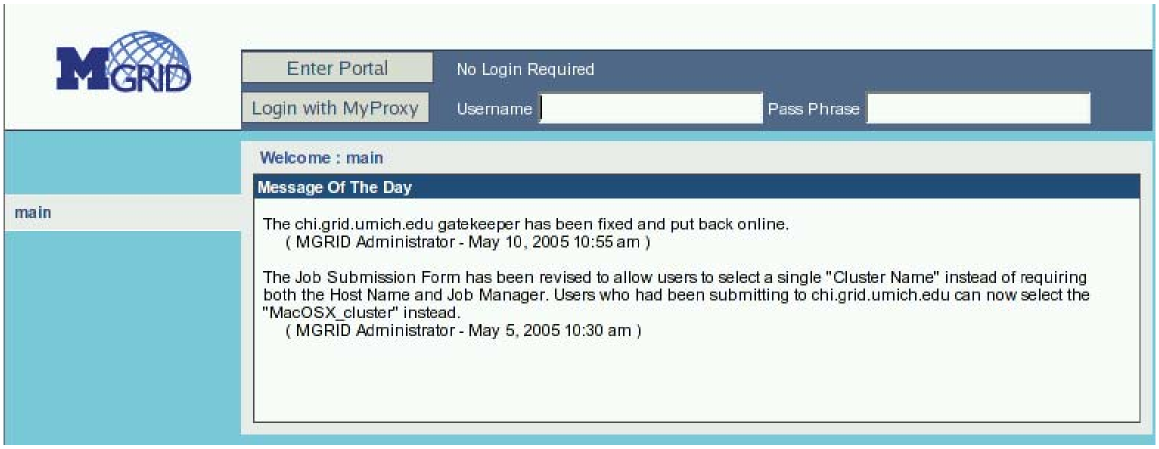
The Texas Advanced Computing Center at The University of Texas at Austin
UTGrid Design and Implementation, http://www1.sura.org/3000/NMI-Testbed/TACC-UTGrid.pdf, part of the NSF Middleware Initiative (NMI) Integration Testbed Case Study Series*, November 2004.

Texas Tech University
TechGrid: Designing and Building a Campus-wide Compute Grid in a Heterogeneous Operating System Environment At Texas Tech University, http://129.118.51.86/zlin/class/TechGrid.pdf, author: Jerry Perez, April 2003.

**The NMI Integration Testbed Program provided practical evaluation of NMI components within the context of real projects and application scenarios from June 2002 through November 2004. For more information or to view the entire NMI Integration Testbed Case Study series, see: http://www1.sura.org/3000/NMI-Testbed.html.*

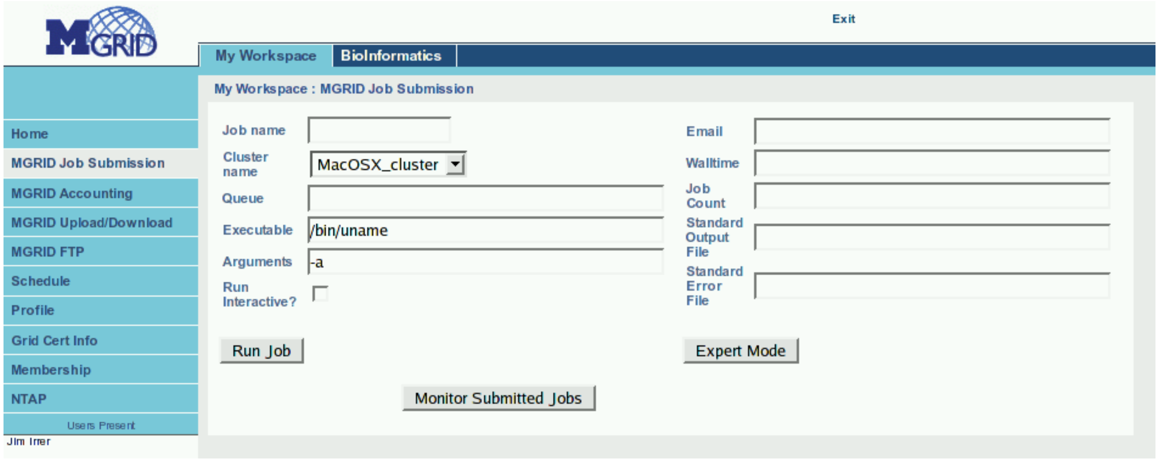
Appendix B: User Interface Examples

University of Michigan



The chi.grid.umich.edu gatekeeper has been fixed and put back online.
(MGRID Administrator - May 10, 2005 10:55 am)

The Job Submission Form has been revised to allow users to select a single "Cluster Name" instead of requiring both the Host Name and Job Manager. Users who had been submitting to chi.grid.umich.edu can now select the "MacOSX_cluster" instead.
(MGRID Administrator - May 5, 2005 10:30 am)

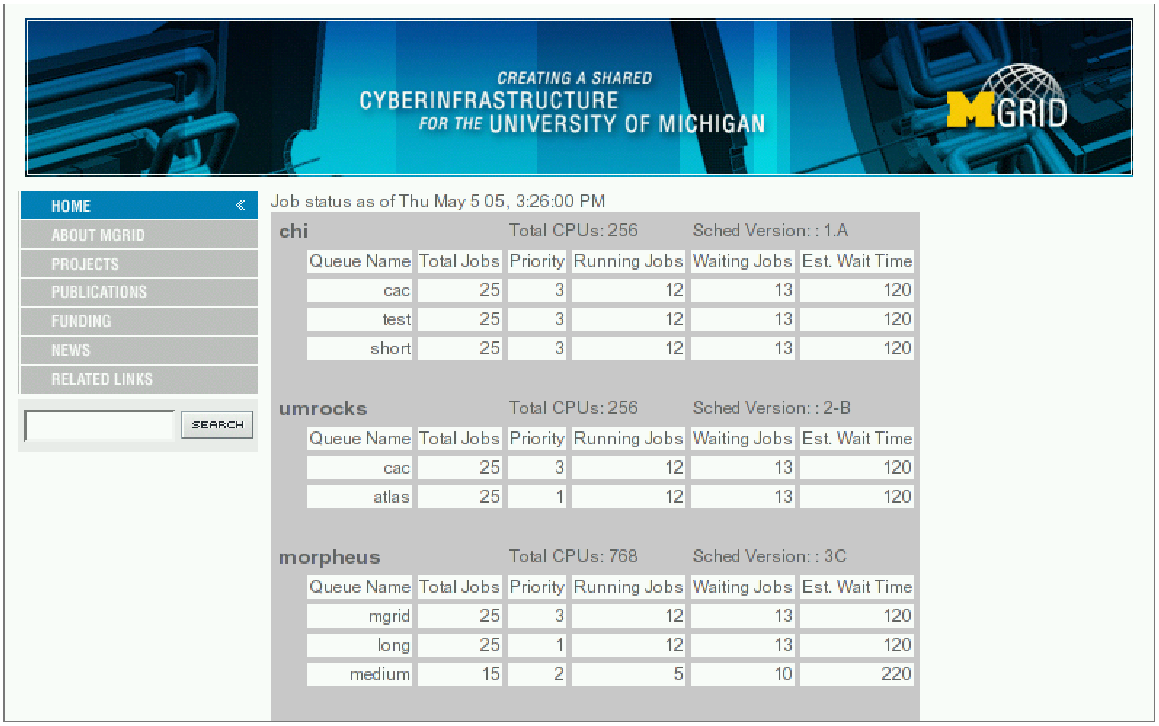


Cluster name: MacOSX_cluster

Executable: /bin/uname

Arguments: -a

Jim Irrer



Job status as of Thu May 5 05, 3:26:00 PM

chi Total CPUs: 256 Sched Version: : 1.A

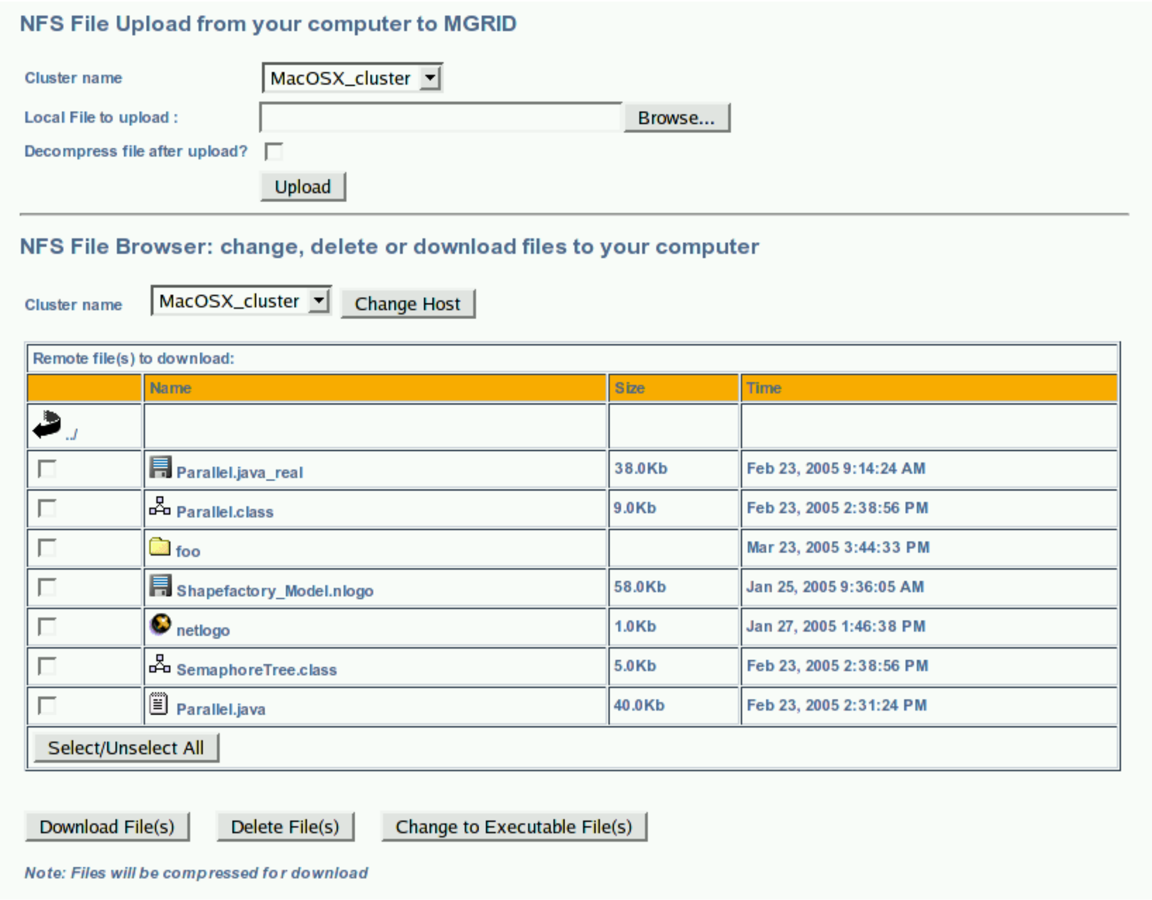
Queue Name	Total Jobs	Priority	Running Jobs	Waiting Jobs	Est. Wait Time
cac	25	3	12	13	120
test	25	3	12	13	120
short	25	3	12	13	120

umrocks Total CPUs: 256 Sched Version: : 2-B

Queue Name	Total Jobs	Priority	Running Jobs	Waiting Jobs	Est. Wait Time
cac	25	3	12	13	120
atlas	25	1	12	13	120

morpheus Total CPUs: 768 Sched Version: : 3C

Queue Name	Total Jobs	Priority	Running Jobs	Waiting Jobs	Est. Wait Time
mgrid	25	3	12	13	120
long	25	1	12	13	120
medium	15	2	5	10	220



NFS File Upload from your computer to MGRID

Cluster name: MacOSX_cluster

Local File to upload :

Decompress file after upload? ☐

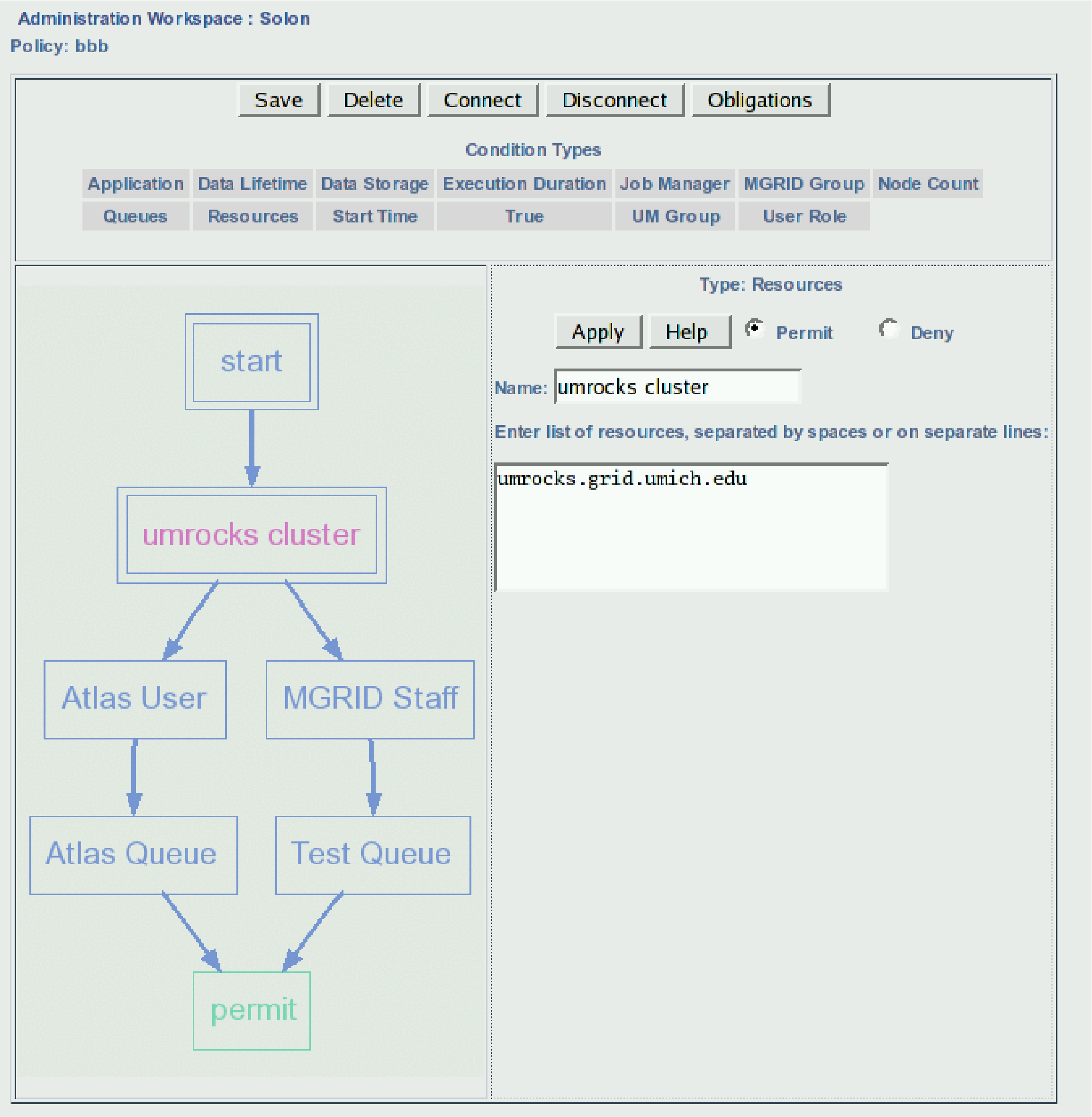
NFS File Browser: change, delete or download files to your computer

Cluster name: MacOSX_cluster

Remote file(s) to download:

	Name	Size	Time
	../		
☐	Parallel.java_real	38.0Kb	Feb 23, 2005 9:14:24 AM
☐	Parallel.class	9.0Kb	Feb 23, 2005 2:38:56 PM
☐	foo		Mar 23, 2005 3:44:33 PM
☐	Shapefactory_Model.nlogo	58.0Kb	Jan 25, 2005 9:36:05 AM
☐	netlogo	1.0Kb	Jan 27, 2005 1:46:38 PM
☐	SemaphoreTree.class	5.0Kb	Feb 23, 2005 2:38:56 PM
☐	Parallel.java	40.0Kb	Feb 23, 2005 2:31:24 PM

Note: Files will be compressed for download



Administration Workspace : Solon

Policy: bbb

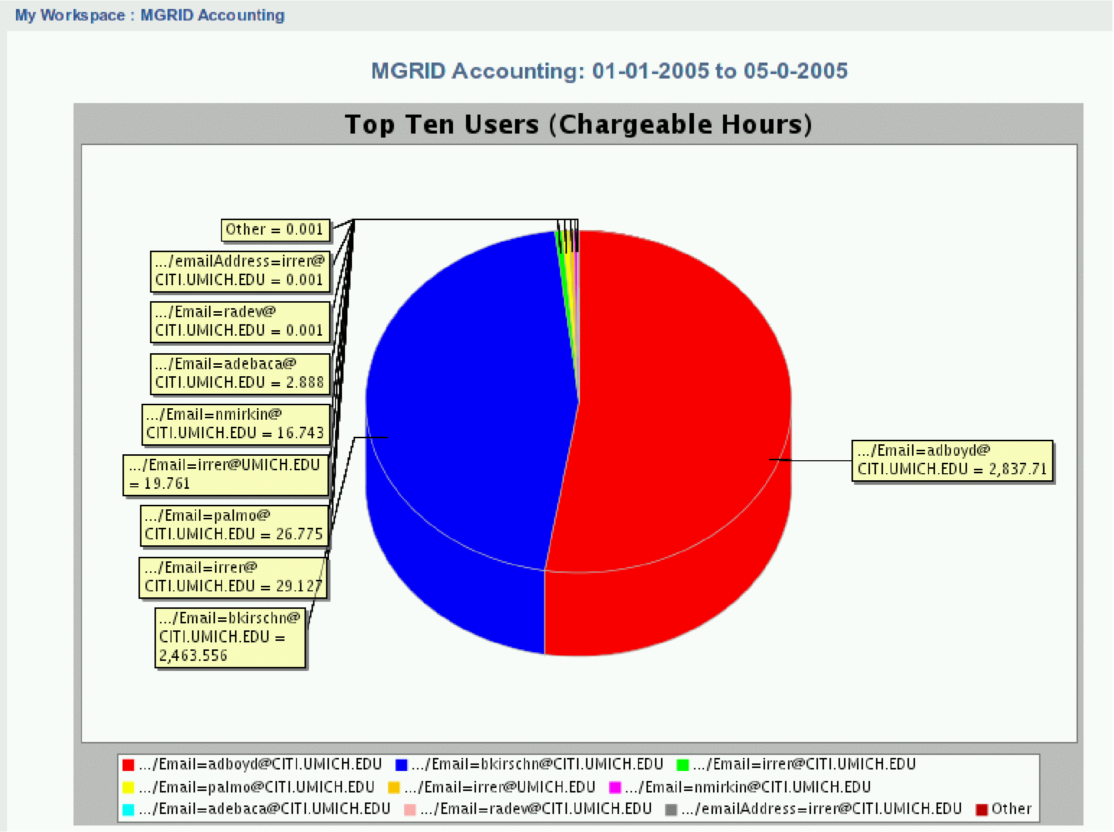
Type: Resources

● Permit ○ Deny

Name: umrocks cluster

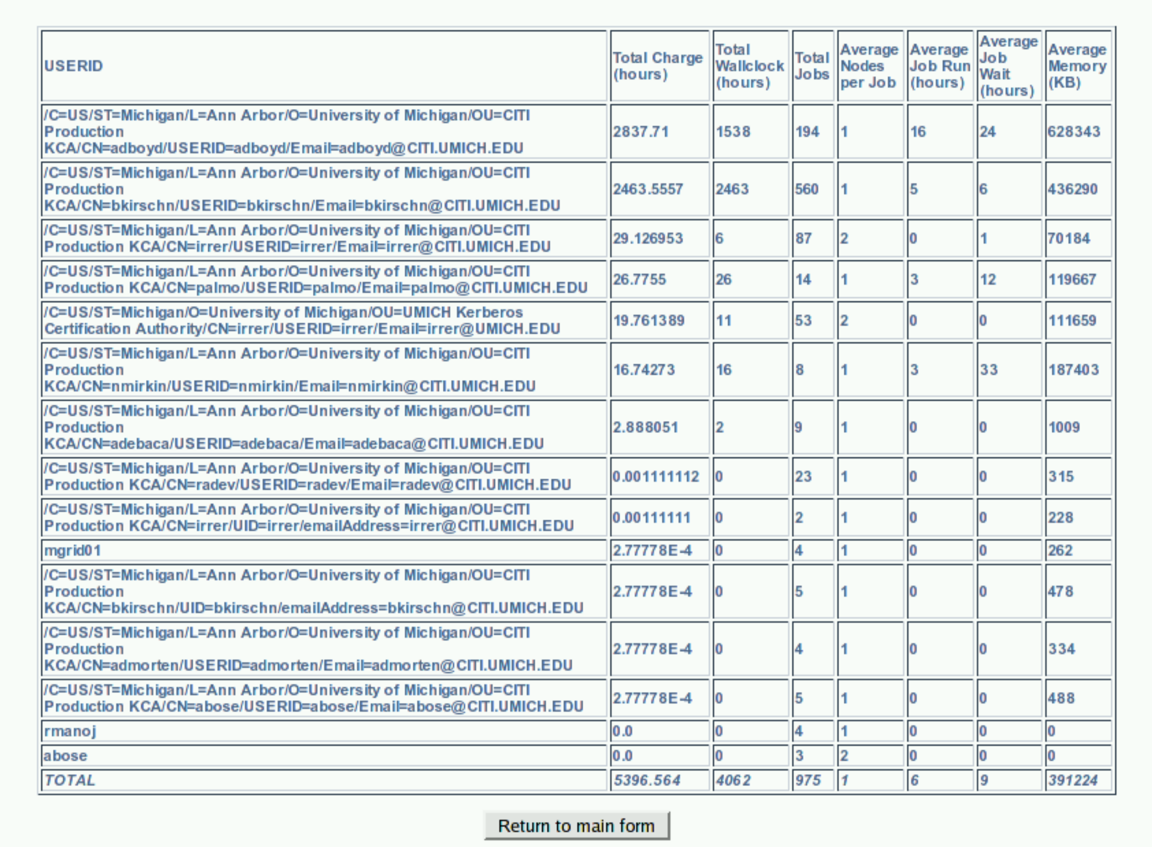
Enter list of resources, separated by spaces or on separate lines:

```
umrocks.grid.umich.edu
```



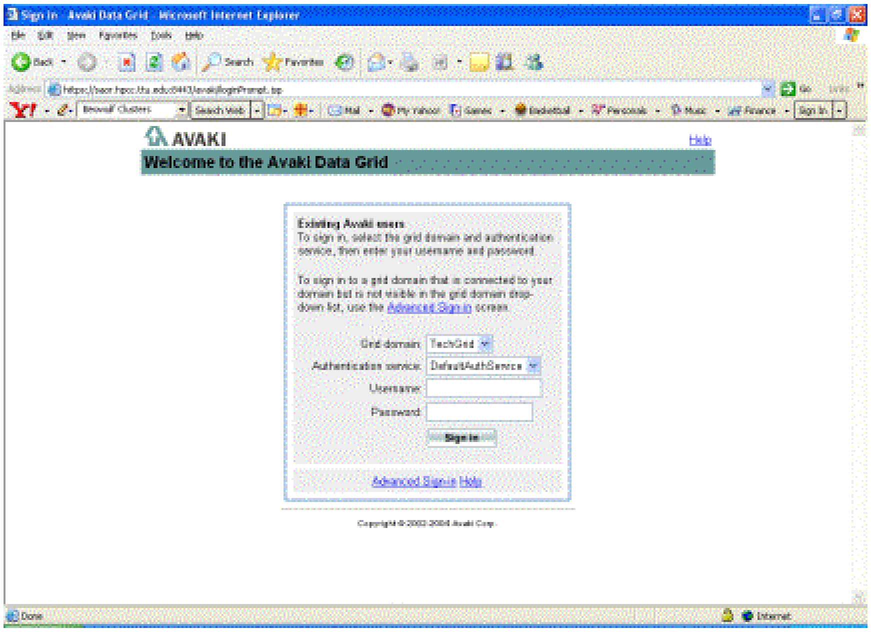
My Workspace : MGRID Accounting

MGRID Accounting: 01-01-2005 to 05-0-2005



USERID	Total Charge (hours)	Total Wallclock (hours)	Total Jobs	Average Nodes per Job	Average Job Run (hours)	Average Job Wait (hours)	Average Memory (KB)
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=adboyd/USERID=adboyd/Email=adboyd@CITI.UMICH.EDU	2837.71	1538	194	1	16	24	628343
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=bkirschn/USERID=bkirschn/Email=bkirschn@CITI.UMICH.EDU	2463.5557	2463	560	1	5	6	436290
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=irrer/USERID=irrer/Email=irrer@CITI.UMICH.EDU	29.126953	6	87	2	0	1	70184
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=palmo/USERID=palmo/Email=palmo@CITI.UMICH.EDU	26.7755	26	14	1	3	12	119667
/C=US/ST=Michigan/O=University of Michigan/OU=UMICH Kerberos Certification Authority/CN=irrer/USERID=irrer/Email=irrer@UMICH.EDU	19.761389	11	53	2	0	0	111659
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=nmirkin/USERID=nmirkin/Email=nmirkin@CITI.UMICH.EDU	16.74273	16	8	1	3	33	187403
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=adebaca/USERID=adebaca/Email=adebaca@CITI.UMICH.EDU	2.888051	2	9	1	0	0	1009
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=radev/USERID=radev/Email=radev@CITI.UMICH.EDU	0.001111112	0	23	1	0	0	315
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=irrer/UID=irrer/emailAddress=irrer@CITI.UMICH.EDU	0.00111111	0	2	1	0	0	228
mgrid01	2.77778E-4	0	4	1	0	0	262
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=bkirschn/UID=bkirschn/emailAddress=bkirschn@CITI.UMICH.EDU	2.77778E-4	0	5	1	0	0	478
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=admorten/USERID=admorten/Email=admorten@CITI.UMICH.EDU	2.77778E-4	0	4	1	0	0	334
/C=US/ST=Michigan/L=Ann Arbor/O=University of Michigan/OU=CITI Production KCA/CN=abose/USERID=abose/Email=abose@CITI.UMICH.EDU	2.77778E-4	0	5	1	0	0	488
rmanoj	0.0	0	4	1	0	0	0
abose	0.0	0	3	2	0	0	0
TOTAL	*5396.564*	*4062*	*975*	*1*	*6*	*9*	*391224*

Texas Tech University



https://saor.hpcc.ttu.edu:8443/avaki/loginPrompt.jsp

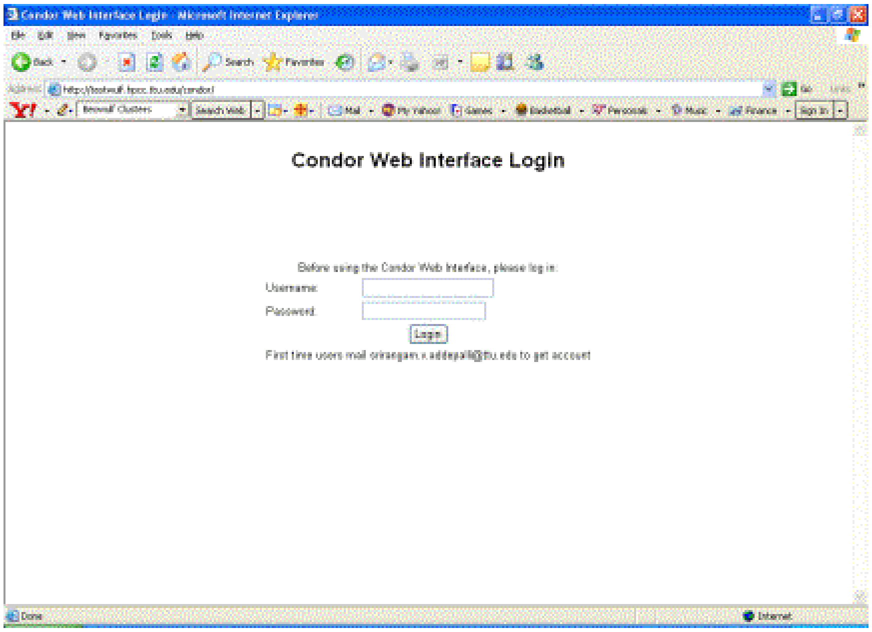
Welcome to the Avaki Data Grid

Existing Avaki users
To sign in, select the grid domain and authentication service, then enter your username and password.

To sign in to a grid domain that is connected to your domain but is not visible in the grid domain drop-down list, use the Advanced Sign in screen.

Grid domain: TechGrid
Authentication service: DefaultAuthService
Username:
Password:

Copyright © 2002-2004 Avaki Corp.

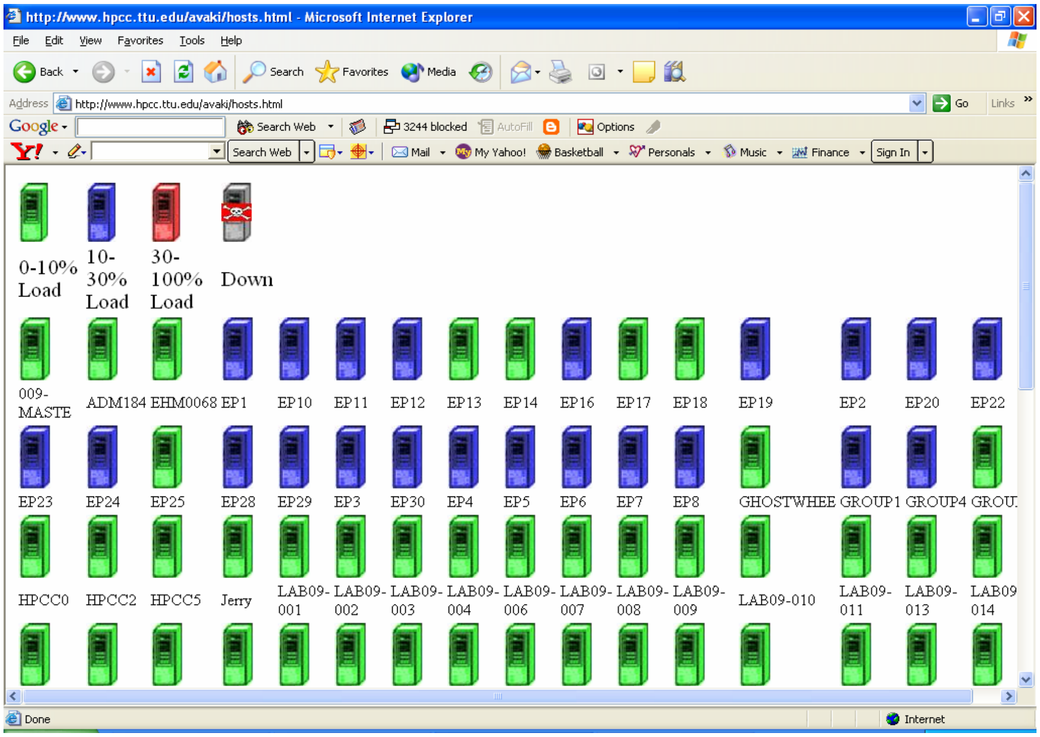


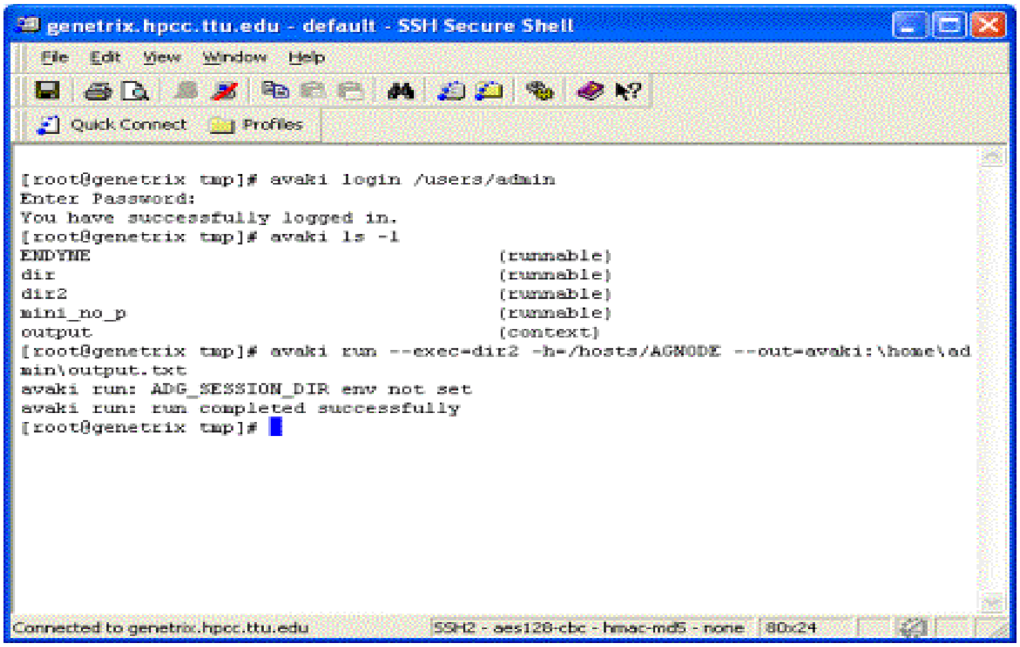
http://testwulf.hpcc.ttu.edu/condor/

Condor Web Interface Login

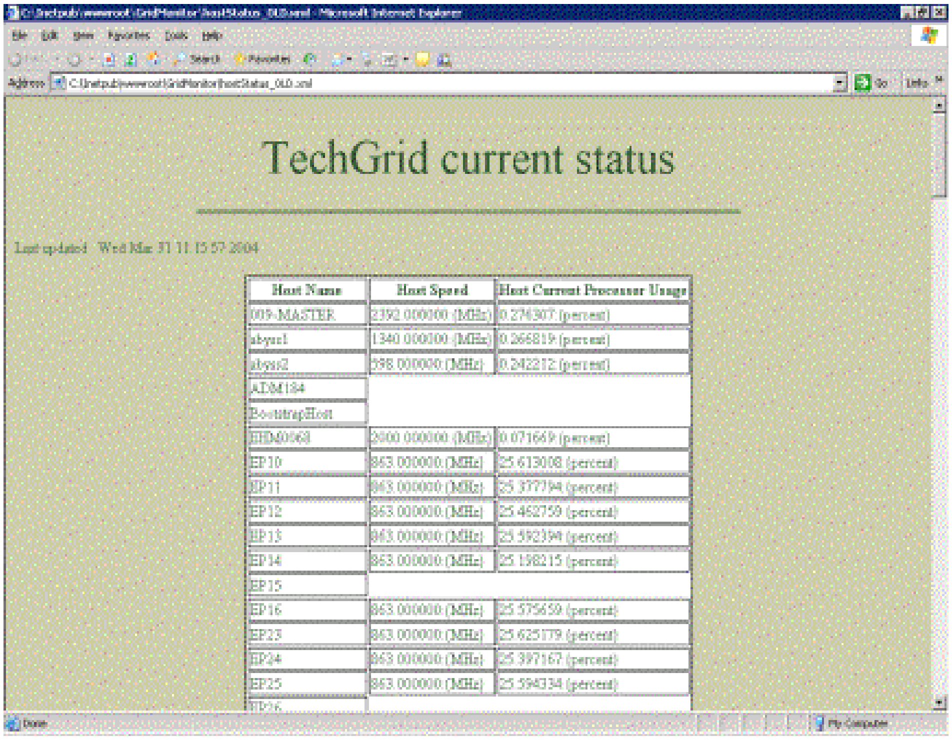
Before using the Condor Web Interface, please log in:
Username:
Password:

First time users mail srirangam.v.addepalli@ttu.edu to get account





```
[root@genetrix tmp]# avaki login /users/admin
Enter Password:
You have successfully logged in.
[root@genetrix tmp]# avaki ls -l
ENDYNE                                     (runnable)
dir                                        (runnable)
dir2                                       (runnable)
mini_no_p                                  (runnable)
output                                     (context)
[root@genetrix tmp]# avaki run --exec=dir2 -h=/hosts/AGNODE --out=avaki:\home\ad
min\output.txt
avaki run: ADG_SESSION_DIR env not set
avaki run: run completed successfully
[root@genetrix tmp]# 
```



TechGrid current status

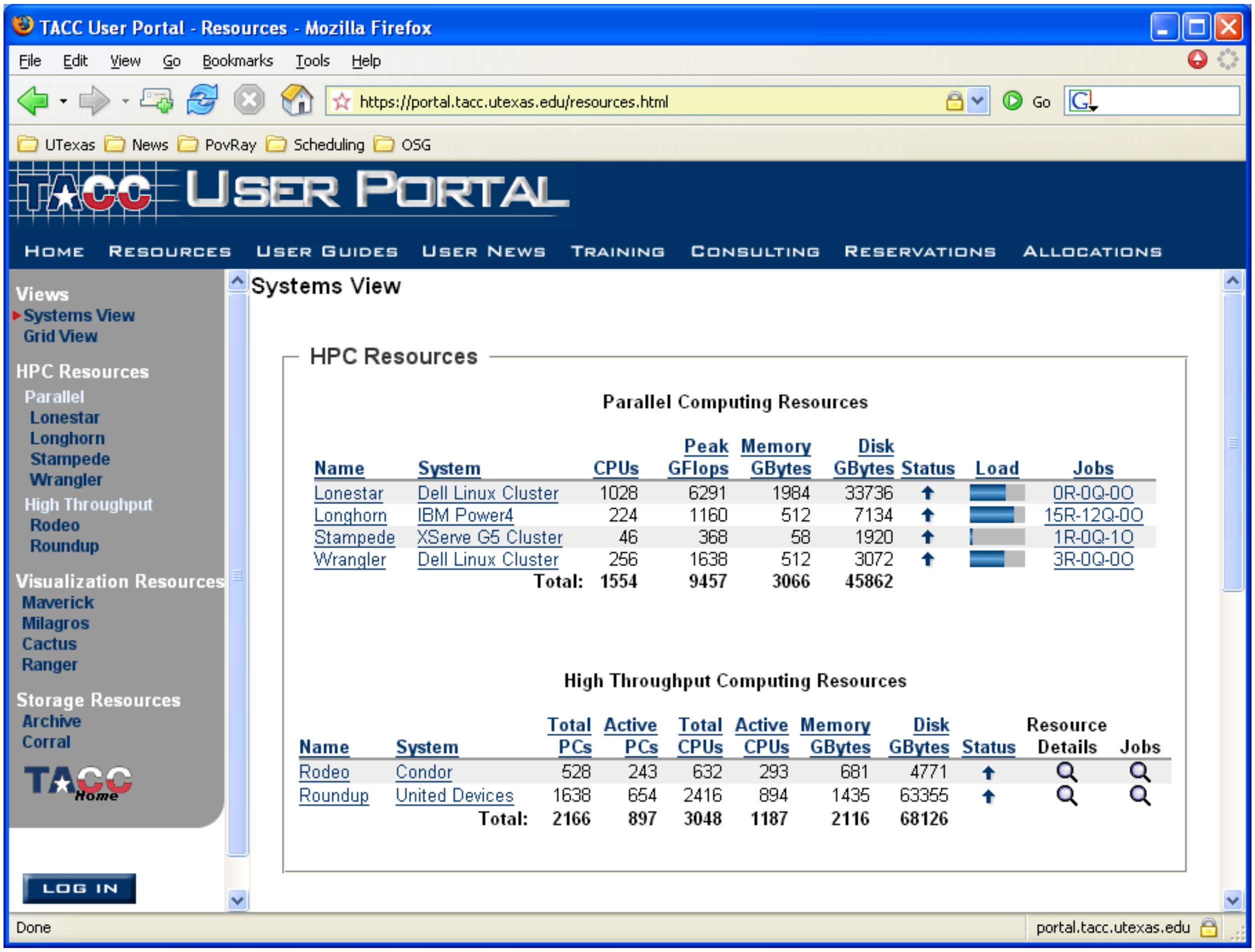
Last updated: Wed Mar 31 11:15:57 2004

Host Name	Host Speed	Host Current Processor Usage
009-MASTER	2392.000000 (MHz)	0.274307 (percent)
abyss1	1340.000000 (MHz)	0.266819 (percent)
abyss2	598.000000 (MHz)	0.242212 (percent)
ADM184		
BootstrapHost		
EHM0063	2000.000000 (MHz)	0.071669 (percent)
EP10	863.000000 (MHz)	25.613008 (percent)
EP11	863.000000 (MHz)	25.377794 (percent)
EP12	863.000000 (MHz)	25.462759 (percent)
EP13	863.000000 (MHz)	25.592394 (percent)
EP14	863.000000 (MHz)	25.198215 (percent)
EP15		
EP16	863.000000 (MHz)	25.575659 (percent)
EP23	863.000000 (MHz)	25.625179 (percent)
EP24	863.000000 (MHz)	25.397167 (percent)
EP25	863.000000 (MHz)	25.594334 (percent)
EP26		

Texas Advanced Computing Center

Project Overview

The UT Grid project vision is to integrate the numerous and diverse computational, visualization, storage, data, and instrument/device resources of The University of Texas at Austin (UT Austin) into a comprehensive campus cyberinfrastructure for research and education. The integration of these vast resources - from 'personal scale' to terascale - into UT Grid will enable resource access and sharing on an unprecedented scale, while new Web-based and command-line interfaces will facilitate new models for utilization and coordination. The development team goals of rapid deployment, adoption, and evolution of UT Grid will enable it to serve as production computing infrastructure in late 2004 (both research and education), while also providing a platform for grid computing R&D. UT Grid will thus present a unique campus environment for knowledge discovery and education.



Systems View

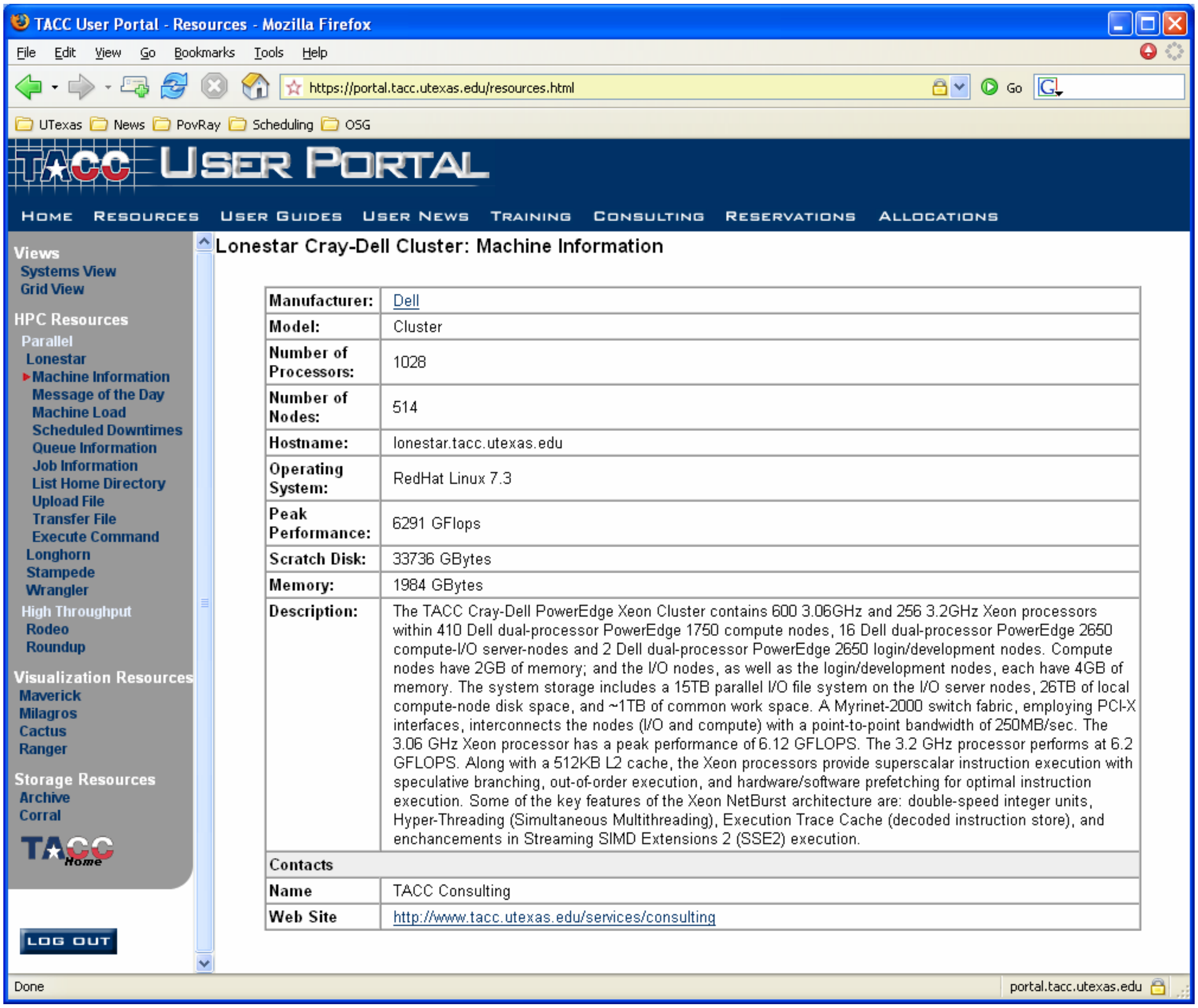
HPC Resources

Parallel Computing Resources

Name	System	CPUs	Peak GFlops	Memory GBytes	Disk GBytes	Status	Load	Jobs
Lonestar	Dell Linux Cluster	1028	6291	1984	33736	↑		0R-0Q-0O
Longhorn	IBM Power4	224	1160	512	7134	↑		15R-12Q-0O
Stampede	XServe G5 Cluster	46	368	58	1920	↑		1R-0Q-1O
Wrangler	Dell Linux Cluster	256	1638	512	3072	↑		3R-0Q-0O
	Total:	1554	9457	3066	45862			

High Throughput Computing Resources

Name	System	Total PCs	Active PCs	Total CPUs	Active CPUs	Memory GBytes	Disk GBytes	Status	Resource Details	Jobs
Rodeo	Condor	528	243	632	293	681	4771	↑	🔍	🔍
Roundup	United Devices	1638	654	2416	894	1435	63355	↑	🔍	🔍
	Total:	2166	897	3048	1187	2116	68126			



Lonestar Cray-Dell Cluster: Machine Information

Manufacturer:	Dell
Model:	Cluster
Number of Processors:	1028
Number of Nodes:	514
Hostname:	lonestar.tacc.utexas.edu
Operating System:	RedHat Linux 7.3
Peak Performance:	6291 GFlops
Scratch Disk:	33736 GBytes
Memory:	1984 GBytes
Description:	The TACC Cray-Dell PowerEdge Xeon Cluster contains 600 3.06GHz and 256 3.2GHz Xeon processors within 410 Dell dual-processor PowerEdge 1750 compute nodes, 16 Dell dual-processor PowerEdge 2650 compute-I/O server-nodes and 2 Dell dual-processor PowerEdge 2650 login/development nodes. Compute nodes have 2GB of memory; and the I/O nodes, as well as the login/development nodes, each have 4GB of memory. The system storage includes a 15TB parallel I/O file system on the I/O server nodes, 26TB of local compute-node disk space, and ~1TB of common work space. A Myrinet-2000 switch fabric, employing PCI-X interfaces, interconnects the nodes (I/O and compute) with a point-to-point bandwidth of 250MB/sec. The 3.06 GHz Xeon processor has a peak performance of 6.12 GFLOPS. The 3.2 GHz processor performs at 6.2 GFLOPS. Along with a 512KB L2 cache, the Xeon processors provide superscalar instruction execution with speculative branching, out-of-order execution, and hardware/software prefetching for optimal instruction execution. Some of the key features of the Xeon NetBurst architecture are: double-speed integer units, Hyper-Threading (Simultaneous Multithreading), Execution Trace Cache (decoded instruction store), and enchancements in Streaming SIMD Extensions 2 (SSE2) execution.
Contacts	
Name	TACC Consulting
Web Site	http://www.tacc.utexas.edu/services/consulting