OSG Campus Grid Working Meeting Notes

Brian Bockelman (Nebraska), Dan Bradley (Wisconsin), Keith Chadwick (Fermilab), Steve Gallo (Buffalo), Sebastien Goasguen (Clemson), Rob Gardner (Chicago), Sam Hoover (Clemson), John McGee (RENCI), Doug Olson (LBNL), Preston Smith (Purdue), Ben Cotton (Purdue), Prakashan Korambath (UCLA)

January 25, 2010

1 Executive Summary

We summarize here discussions from an OSG working meeting held at Fermilab, January 19-20, 2010, focusing on Campus Grids (CG). The goal was to create a technical summary document identifying and comparing current CG implementations, best practices and patterns, as well as address technical issues moving forward in a number of topical areas.

An important outcome of this effort is that we believe OSG may be a good forum for building a community for establishing best-practices and knowledge sharing about building and evolving such infrastructure within the campus. Conversely, in terms of connecting campuses to national infrastructure it seems obvious OSG should develop a strategy across all existing work areas to support these connections; indeed this may be uniquely the most important strategic area of focus for future generations of OSG.

The topical areas identified during phone discussions held prior to the meeting, including:

- Global file systems and storage systems in use or needed at the CG
- Creation and use of seamless user environments
- Role and use of virtual machines and cloud computing technologies in CG
- Implementation challenges resulting from provincial campus issues, local constraints and priorities

During the meeting we heard of progress or plans from the following CG efforts: GLOW (Wisconsin), Purdue, FermiGrid, University of California Grid, Nebraska, and New York State Grid (a regional CG), each project choosing to describe a number of activities, challenges, and strategic approaches taken in creating their infrastructure. There quickly emerged a number of interesting questions and points we believe are relevant to OSG, and that OSG should have well thought out answers to providing clear project and consortium-wide guidance and vision. We list some of them below, starting from the obvious:

1. Why build campus grid infrastructure and what role can or should OSG play?
2. What incentives are in place to convince faculty, departmental IT managers, CIOs, provosts (and others who may be unfamiliar with campus grids and OSG) to join their resources: a) together across the campus, and b) to national Cyberinfrastructure facilities using for example the services of the OSG?
3. What lessons can be drawn from successful efforts in pooling resources within a CG that can be useful for stimulating opportunistic sharing across OSG? For example, we've seen sharing agreements between contributed "pool" resources among campus researchers far more dynamic than what occurs in the wide-area on OSG.

4. What missing pieces could OSG provide in catalyzing opportunistic sharing or trading of compute (and ultimately storage) resources beyond and between campus grids, within the national Cyberinfrastructure context? How can potential joiners to OSG readily realize and monitor the potential benefit to their organizations?
5. Are there additional principles needed making up OSG's core mission and architecture to catalyze sharing within and across campus grids? For example, resource sharing so far within OSG has been governed within the context of the VO (virtual organization) with its associated required infrastructure and organizational frameworks. However, the diversity and size and opportunity of resources at the CG is more complex and often less VO-centric, especially those without well-established communities already within OSG but have sizable excess resources.

In this document it has been our intention to not offer any solutions or specific, prescriptive implementations or architectures to the complex issues touched by CG, but to point to key principles and issues that pose significant challenges.

2 Opportunities

We see several areas where the development of campus grids can create opportunities for our community:

- Harboring local HTC/HPC experts to help scientists get started (perhaps like TeraGrid's "Campus Champions"). An individual research group may simply not have the continuity of funds and demand for computing to support such people, but making the case for such support at the campus level has succeeded in Nebraska, Purdue, GLOW, Clemson and probably most or all of the successful campus grids. The OSG Engage group has demonstrated how effective and essential this sort of activity is. The more the better.
- Encourage more efficient and professional system administration. This could lead to less downtime, less manual configuration, better system tuning, reduced costs, and more efficient use of the scientist's time.
- Benefit from collective buying power and decision making. Examples where this has been very beneficial have been reported by FermiGrid, Purdue, Clemson and GLOW.
- Achieve more efficient usage of machines and licenses. For example, Purdue and GLOW get about 15% more out of their clusters by sharing them.
- Testing ground for new Grid technologies. If something has wild success at a campus, it might be a breakthrough for OSG as well.
- Participation in a national community which may help leverage local investments, increase revenue and diversify funding sources.

Some opportunities that OSG can create for campus grids are

- Making cross-organizational collaborations practical (sharing data and compute power, authenticating users).
- Bootstrapping campus grid activity in cases where connection to OSG happens first.
- Sharing expertise, best practices, Grid software stacks, documentation and training modules.
- Our community sometimes claims or implicitly assumes that linking resources to OSG will benefit sites by making them part of a collective pool that provides greater computing power to its members than they would get out of their own isolated resources. Therefore, it seems like an item that would

normally be added to this list. However, providing greater total computing power to the users of OSG does not necessarily mean that resource providers are the ones who benefit. If it *is* true that providing resources to OSG is strongly motivated by the expectation of increased computing power, wouldn't this lead to discussions or estimations of value exchange by those who pay the bills? Most concretely, my site expects to spend X on power and system administration to support your VO, but we expect to get Y in return, which is better than simply hibernating our idle machines. At the campus grid scale, part of this discussion is implicit in the university's typical willingness to provide free power to researchers, because the institution hopes to achieve more as a result. So the resource owners mostly just need to agree that the administrative costs of sharing are worth it (and an argument can be made that the administrative costs of a CG are actually less, not more than isolated clusters). So for a campus, sharing seems to support both the resource owners' and the institution's desires for more computing power, without considering other benefits such as increased ease of collaboration. At the inter-campus OSG level it is not obvious that any accounting or expectation of increased computing power is really taking place in the thinking of the resource providers. Resource sharing between VOs for the most part appears to be motivated more along the lines of volunteer computing. We mention this because it seems like a noteworthy difference between the dynamics of a campus grid and the dynamics of the present incarnation of OSG.

3 Difficulties on the Street

The shared cluster model (or condominium computing model) of resource sharing has been very successful on many university campuses (Wisconsin, Duke, Rice, Purdue, Clemson, UCLA among others). In this model, the benefit for a researcher to buy into the program and contribute research dollars for nodes in a community owned cluster is clear: the overall cost of the resource is shared among participating researchers and the IT organization hosting the resource; resource administration and management is typically more professional, moved out of the research lab and into an IT organization; the overall utilization of the system is greater than the alternative of researchers purchasing and managing their own individual systems; researchers maintain an important confidence level similar to individually owned resources as there is a guaranteed level of service commensurate with their level of contribution to the shared system; opportunistic access to more resources than their direct contribution.

One could argue that the OSG Campus Grids initiative, or OSG Campus Shared Resources initiative, is a scaled up version of the shared cluster model that crosses new boundaries and borders such as resources and even campuses. Given the fundamental OSG principles of local autonomy and control, OSG resource owners maintain control of and therefore a guaranteed level of service to their own resources, yet provide unused cycles out to a broader community. By joining the OSG community, researchers gain opportunistic access to significantly more resources than their local system via sharing couched in the framework of the VO model. The overall utilization of resources is greater in this shared model and the overall cost of the infrastructure is shared among the resource owners, and the OSG project.

Given these similarities between the successful shared cluster model and the OSG Campus Resource Sharing initiative, there are some important differences that affect OSG's ability to gain traction and catalyze change at the campus level on a larger scale:

- Infrastructure know-how: Cluster management has been around for a number of years, and is a reasonably well understood challenge. Enterprise wide resource sharing is significantly more complex, not as well understood, and the necessary tooling is less mature. For example configuration, identity, and data management. For a research team that owns a resource, their IT admin burden decreases (typically to zero) in a shift to a shared cluster, but increases non-trivially in a shift to CG style resource sharing.

- Usage know-how (and end-user environments): In all of the OSG connected CG deployments that were examined, it was noted that there are separate submission mechanisms for users to run on the CG vs. OSG. The infrastructure is not yet robust enough to allow for a CG to easily “spill over” jobs from the CG to OSG as the CG reaches capacity. Because of this missing piece, a researcher must make a decision before job submission whether to send it to the CG or to the OSG. Our experience in the campus engagement activity indicates that this lack of integration significantly hinders CG adoption.
- Benefits: In the shared cluster model, there is a more easily quantified *potential value* that can be gained from the opportunistic access. The researcher likely has a sense of the scientific computing landscape on their campus and the other researchers participating in the shared cluster. Thus, they can make a somewhat reasonable informed guess as to how much of the opportunistic cycles they are likely to be successful in acquiring. Given the small size of this community of sharers (relative to a national infrastructure), they can also horse trade and make informal agreements about usage time windows, or simply gain insight into likely times of high availability. Another way of thinking about this is that scope and locality in this model enables a simple marketplace where members can exchange value and benefits from the community resource. This is not possible in OSG today, and the community is too inaccessible for making arrangements for future availability in this way. This also relates to concerns that HEP VOs will overwhelm the system leaving little to no opportunistic availability.
- Incentives: In the shared cluster model, the incentive for increased overall utilization of resources is in fact a fiduciary responsibility of the campus CIO: to provide capable and cost effective research computing to faculty and staff on campus. In OSG, the incentive for increased overall resource utilization is rooted in the VO’s, groups of like-minded researchers working for the betterment of a specific science community. Each of these individually works very well. However, mixing the two is like mixing oil and water. Without some form of tangible and quantifiable value exchange, CIO’s simply cannot justify the sharing of campus resources beyond the borders of his/her administrative purview. In the cases where this has been successful (Purdue, Clemson, Fermilab, NY State Grid), there were trail blazing thought leaders in place who were willing to contribute campus resources and take a leadership position in the national Cyberinfrastructure. In doing so, they likely enjoyed important benefits that are difficult to quantify, such as help in winning future awards, attracting faculty, etc. However, this level of incentive quantification is not sufficient for broad adoption of CG infrastructures.

4 Outreach and Engagement

As we have been discussing, without CIO-level support a campus grid project is not likely to gain any traction. Support from the CIO is important for making resources and personnel available to the project. In some cases, for example Purdue and Clemson, it is the CIO who drives the creation of the campus grid. However, some institutions may find the CIO indifferent or even opposed to the idea of creating a campus grid (some have reported that campus resources are dedicated to students and should not be used by external users). The convincing argument will depend on the specific objections of the CIO, but it is worth discussing some of the more compelling arguments here: namely that a campus grid can be set up to take advantage of existing underutilized resources. In 2009, Purdue provided 17 million hours of compute time on it's DiaGrid, i.e. 17% of the total HPC hours that year. For a campus without a traditional cluster, the case is even stronger as scientists can spend more time analyzing data instead of waiting for computations to finish.

While having an engaged CIO is beneficial, this of course does not preclude campus grid creation within academic divisions on campus; however this usually involves senior involvement of institute heads or Deans with fiscal authority over the required campus resources (space, power, cooling, networking) and limits the size of the contribution that the CG can make globally, even though it can still prove to be a valuable local resource.

It is in fact the research scientist who is the ultimate driver of a successful campus grid. The best technical setup in the world is of little use if there is no work being done on it. As a result, reaching out to researchers is a critical part of establishing a campus grid. Other researchers who have their own success stories with a campus grid can be the best advocates, not only to their peers, but also to the CIO. Indeed, a clamor from the research faculty may be the compelling argument that gets the CIO support when all other approaches have failed. Indeed if faculty do not ask for it why would a CG be created, often other projects have higher priorities.

By approaching the researchers with real-world cases showing how a campus grid can help meet the research and instructional missions of an institution, the necessary support can be achieved. However, the outreach to and engagement of researchers does not end when the campus grid is declared operational. In order to provide ongoing success the researchers must continue to be involved. This means modifying the setup to provide the necessary environment as well as providing support for understanding how to best make use of the grid. The end result should be that researchers could focus on their area of interest, and not have to worry about maintaining their research-computing environment. While OSG can help startup the engagement and outreach efforts, long-term sustainability will be based on local support structure.

5 Strategies for Resource Aggregation

At the root of campus grids is the idea of sharing and aggregating the resources on a campus. At institutions with successful campus grid programs, several different strategies exist for aggregating resources into a grid.

5.1 Aggregation approaches in use today

GLOW

GLOW, at the University of Wisconsin, aggregates physically distributed collections of systems into a single Condor pool. For example, the engineering group acquires, houses, powers, and cools its own collection of machines in its own space, as does the Physics department. Software and operating systems are managed centrally.

A site can join GLOW with a minimum contribution of about one rack of machines. The University and the Condor Team adds value to the campus grid with additional opportunistic resources.

GLOW provides access to the AFS software repositories of its members. Other than this, there is no shared file system.

Purdue

Purdue aggregates resources both via its "community cluster" program, where faculty research dollars are pooled together to build a single large cluster, with professional system administration, support, and facilities; and the use of Condor to tie together otherwise idle machines in student labs and around the campus with idle cluster nodes. A group can join the community clusters with a single node.

Distributed Condor resources at Purdue are managed with the "CycleServer" management console - useful for maintaining Condor configuration on machines with distributed ownership. A tool for configuring, monitoring, etc is useful to administrators operating a grid of systems owned by multiple groups.

Purdue provides pre-configured packages to aid departments with adding resources to the campus grid.

Purdue clusters share centralized NFS servers.

FermiGrid

Rather than combining individual cluster node systems into a central grid or cluster, FermiGrid aggregates many distinct, previously independently owned and operated clusters together with middleware - placing all clusters behind a single point of entry. All FermiGrid clusters share site-level services (GUMS, SAZ, MyProxy, VOMS, etc)

FermiGrid clusters share centralized NFS servers (using BlueArc).

NYSGrid

The New York State Grid initially aggregated together Blue Gene systems at New York state universities for others to use. Additionally, Buffalo is aggregating campus resources with Condor, and backfilling HPC clusters with Condor, much like Purdue's model, with one or two gatekeepers functioning as an entry point.

Other Institutions

- Clemson aggregates resources in a model very similar to Purdue: condominium cluster and large condor pools centrally managed. It is important to note that Purdue former CIO is now Clemson's CIO.
- Nebraska leadership is supportive of sharing and combining resources and is taking advantage of opportunistic funding opportunities (gifts) for broad benefit to campus researchers.
- UC (California) Grid presents distributed resources (at many UC campuses) through a single portal, providing easy access for job submission and monitoring, and data transfers.

6 Connecting Researchers to Resources

During our discussions it became clear that (somewhat paradoxically) it is helpful if the focus is less on "building Campus Grid infrastructure" and more on connecting researchers to any resources available to them. We decided to dissect this approach somewhat. Thus for campus grids, there are three parts to the subject of "Connecting Researchers to Resources":

1) Resources: Some sufficiently enticing and easy to use computational or storage resource on campus that can be used for scientific computing.
2) Researchers: Finding and maintaining the interest of campus researchers who have a need for computing in order to get their science done.
3) Connecting: Selecting the right level of engagement for researcher based on the present and future need of the science. In this section, we are not going to tackle the acquisition of resources but rather finding and growing the needs of researchers and connecting them to the types of scientific computing that suit them best.

We consider two CG efforts in this section – experience garnered by doing these activities at the University of Nebraska-Lincoln and at the University of California Grid. Both of these examples provide useful insights into the thinking and considerations that are required from the resource sharing, user interface, and application / workflow porting perspectives.

6.1 Connecting Nebraska Researchers

In Nebraska, new researchers are using the following means:

1) Top-down approach: UNL enjoys strong support from the Office of Research, which encourages scientists to partner with the Holland Computing Center when additional computing is needed and does not allow new grants to purchase their own computing resources outside HCC.
2) "User Recommendation": Users often are recommended by their peers, whether they are new research groups recommended by colleagues who use HCC or by new students who are joining a research group that is already utilizing HCC.
3) Active Engagement: Some research groups have joined after having directly talked to heads of departments, deans, or research group leads. UNL feels this is one method to "break in" to a new campus or department where we have no active users.
4) Education: HCC staff members teach CSE classes almost every semester. Topics in the past have included system administration, parallel programming, cluster computing, and grid computing. Students who believe they need scientific computing (or whose advisers believe this) often take these classes and get involved with HCC through their class project. These classes are used as a recruitment tool for student workers. A local workshop is offered approximately once a year. UNL has never examined the "retention rate" for active users, or thoroughly examined the reasons why active users become inactive.

Before we going into detail of how researchers are connected to resources, a few definitions are given (one can skip this section if they are familiar with all the keywords).

Primary types of resources:

1) Commodity Linux clusters: Clusters composed of low-to-mid-range server hardware; commodity Ethernet network; small number (<=16) of cores per node; 1-3 GB RAM per core.
2) Tightly coupled clusters: Commodity Linux clusters with a low-latency network.
3) Specialty resources: Machines serving a specific niche purpose not well suited for general usage or non-dedicated applications. Examples include GPU-equipped machines, SGI Altix / large memory single-system-image machines, and possibly machines with non-x86 architectures.

We also divide the jobs up into general classes:

1) High throughput: A large number of single-core jobs; usually a large number of jobs (hundreds to tens of thousands) form a single workflow, which might have trivial or complex interdependencies (a large number of jobs should be able to be run simultaneously). Parallelism is achieved through running additional jobs.
2) High performance: MPI or other massively parallel jobs. These tend to take up significant amounts of computing resources - many, perhaps hundreds, of nodes. Usually, a small number (<10) batch system jobs per workflow
3) High throughput, high performance: Workflows that mix the characteristics of high performance and throughput; usually multi-core jobs running on a single machine.
4) Specialty - jobs that can only run on specialty resources.

At UNL the following pairings have made the most sense:

1) Specialty jobs can only be run on specialty resources. The users with specialty jobs often have more computing expertise; they know what they want and they can take care of themselves if they have access to the resource. No interest in distributed or grid computing. Care must be taken to take the added cost for hardware support and purchase into consideration.
2) HPC jobs: Generally, HPC jobs can only run on HPC resources. The portability of these jobs is low, as the researcher is often interested in intimately tuning the jobs to the machine (we've had experts state that it takes 1-2 months to “break in” a new machine for their code); it can take many recompiles per machine to get the desired performance. These workflows may not scale well by adding additional cores to the jobs, which is why compiler settings are so important. Amongst these jobs, there is usually low interest in distributed computing, and hence a low probability of success for Engagement with campus grids effort. The only potential successes we foresee are from converting those users whose jobs are really HTC to use HTC methods; for example, there are embarrassingly parallel workflows ideal for HTC that are implemented in MPI because that was the only tool the researcher was familiar with.
3) HTPC jobs are only run on tightly-coupled clusters at Nebraska. These have the potential to run on commodity cluster and a core stakeholder (CMS) may express interest in this. Nebraska will probably wait for guidance and leadership from OSG Satellite for running HTPC jobs on the grid. There are a small number of local users whose jobs might fit this description; they may eventually be a target for the campus grid. Currently, the cost for porting is too high and the potential for increased resources for these users is too low.
4) HTC jobs: These jobs can and are run almost anywhere - HPC, HTC, or even specialty resources. These workflows scale by adding additional jobs - there is less focus to highly tune the job to each machine they run on. These users are more interested in distributed computing and their jobs have a higher probability to be successfully ported to grids.

At Nebraska, the hope is to offload as many researchers off to a campus grid or the OSG as possible. Porting a job to the campus grid increases utilization of all local resources and increases the resources available to a single research group. UNL currently **does not** attempt to port a job to the campus grid if they meet any of the following criteria:

1) Software requires licensing or license server.
2) Software requires multiple cores per job.
3) Workflow can be done within the desired timeframe regardless of how busy the cluster is. I.e., almost any local user should be able to finish a 1,000 compute hour workflow overnight through fair-share; if the turn-around of 8 hours is acceptable for the user, there will likely never be a need to use distributed clusters.
4) Workflows which are highly data-intensive (more than 1GB of input per job)

(1) and (2) are software limitations that can be solved, but the solutions are complex enough that the costs outweigh the benefits. (3) is difficult as it is anticipated that clusters will become increasingly over-subscribed; if in doubt, UNL doesn’t apply it. It is important to consider the potential science benefit versus the cost of HCC support time; there are cases where a HTC workflow meets all the criteria for running on the OSG except actually being “large enough”. When a researcher has HTC jobs that don't meet any of the exclusion criteria above, UNL envisions the following steps for a successful campus grid user:

1) Run application interactively on any HCC cluster, and run the workflow from start to finish for one path (i.e., if the workflow is a sweep through 3,000 input parameter sets, verify they can run it on 1 parameter set). The HCC effort involved is usually the systems administrators installing new software dependencies or any HCC employee helping with Unix basics.
2) Port the application to the Condor cluster. Express the workflow dependencies are expressed as a Condor DAGMan. Each individual Condor job should encompass a "reasonable" amount of work (i.e., should be between 30 minutes and 8 hours long, with an average of 2 hours). Data dependencies should be well defined and expressed in the condor job; the user should not depend on a shared file system. An HCC application expert or integration expert will be able to help here. It is possible that, after this step is completed, the user is satisfied for quite awhile. It is possible they may not continue to step 3 until their computing needs increase or usage of the condor cluster increases.
3) Port the application to the campus grid. Currently, this is done by individual engagement with the integration expert. Usually, this involves:
    a. Getting the user a grid certificate and adding them to the GPN (or Engage?) VO. Training with OSG grid basics.
    b. Deploying their application code to all sites supporting GPN.
    c. Modifying their Condor submit script slightly to use OSG-MM (match-maker) instead of "normal" Condor. UNL already has implemented a mostly automated tool to do this. UNL evaluating GlideInWMS that should make this step even easier.

UNL believes (2) is a crucial step in order to allow a user to fully debug their application locally (where it is easier to separate condor errors from application errors than separate condor-g errors from application errors). It also provides the cleanest transition from a single local resource to an OSG-like resource, especially when the data dependencies are correctly expressed.

6.2 Connecting University of California Researchers

We now turn to experience from the UC Grid project that has focused principally on web-portal based designs. The architecture and the web interface for the UC Grid Portal was evolved from experience as an organization giving extensive consulting help to users who run high performance computing applications. Some observations are:

1) Majority of the CPU time-consuming users are using commercial or precompiled applications such as Gaussian, NWChem, Matlab, Mathematica, R, Q-Chem, Amber etc.
2) UC Grid found while most of the young researchers have extensive backgrounds in browsing and using the web, they lack experience in command line computing interfaces.
3) They also have to learn basic data management commands such as those that group files into a single tar file, transfer files between their local machine and the cluster(s) they are using, etc.
4) Different clusters use different job schedulers such as condor, SGE or Torque. This often confuses users who already don't know much about Linux or Unix.
5) Some researchers do collaborative research with researchers at other UC campuses and/or with other university campus researchers. So there is a need to authenticate them within their campus as well as outside their campus.
6) There are a lot of idle resources on many of the clusters but the cluster owners are hesitant to share those resources to others due to lack of secure transfer of those resources to unknown users.

7) License fee for some of the commercial applications are very expensive and often times a lot of unused licenses are seen.

As a solution to this UC Grid designed a Job submission service and a Data Manager Service with a web interface to upload input file or specify arguments. In order to use the job submission service the users only need to upload the input file and choose the number of processors and duration of the job. The Grid portal system will submit appropriate GRAM jobs calling suitable commands because the Grid Portal maintains a database that knows exactly how to run a specific application on each participating cluster, where the executables reside, how they are invoked, default arguments, etc. For generic job submission users will have to choose their executable and other parameters. Data Manager services uses GridFTP to transfer files. UC Grid chose Globus Toolkit as the underlying grid software because of its wide usage in Teragrid and other national grids such as OSG. This also provides a common user authentication mechanism through the use of X-509 based certificates.

UC Grid also wanted anybody in any of the UC campuses to have a unique certificate so that he or she can be uniquely identified from anywhere. This led to the creation of a single certifying authority for all campuses. As all of the ten UC campuses use Shibboleth, it was decided to use the campus Shibboleth authentication as the basic service to authenticate and issue the certificate. As of now a UC Grid certificate is used only for the services from the UC Grid portal. It was therefore decided to keep the user certificate with the portal and allow users lease only the short-lived credentials as and when they login at the portal from a MyProxy server.

Finally, UC Grid added some of the services such as interactive login through VNC due to user demands for real time code development and debugging.

7 High Availability Services at the Campus Grid Scale

In the evolution of a campus Grid, a large number of resources will eventually wind up being dependent on various services. The traditional set of these services includes:

- Space and power
- Environmental management (heating, cooling, humidification)
- Networking (physical network, DNS)
- Staff

There are various “traditional” methods to deal with these service dependencies:

- Generators and UPS for power
- Multiple CRAC units so that failure of a single unit does not impact the environment
- Use of switch and router capabilities to provide network fault tolerance
- Redundant DNS (primary and secondary)

The campus Grid may also introduce dependencies on additional Grid specific services, such as (using FermiGrid as a model):

- Virtual Organization Membership Service (VOMS)
- Grid User Mapping Service (GUMS)
- Site AuthoriZation (SAZ) Service
- Squid Web Cache
- MyProxy

For the initial building of the campus grid, these Grid specific services can be provisioned as non-redundant services, but as the campus Grid grows, there will come a point when the Grid specific services will need to be commissioned in a fault tolerant or highly available (HA) infrastructure (an outage of either GUMS or SAZ has the potential to impact >5K jobs/hour on a 20K job slot campus Grid). FermiGrid is addressing this vigorously.

In the case of a regional Grid, the need for redundant services is even greater, since a power or network outage in one administrative area of the regional Grid has the potential to impact the availability of resources across the entire regional Grid.

The need to deploy these Grid specific services in a redundant deployment may be viewed as requiring a large amount of hardware resources to accomplish. Fortunately, virtualization can be used to deploy these services on a minimal hardware footprint.

For the set of Grid specific services listed above, FermiGrid (the Fermilab Campus Grid) has developed configurations that can support in excess of 20K job slots on two (appropriately configured) “midrange” systems. These configurations are freely available, and the FermiGrid personnel are willing to consult on the necessary deployments for campus Grids that are considering joining the Open Science Grid.

8 Leveraging Emerging Technologies

There are a number of emerging technologies affecting the landscape of campus research computing and it is sensible that we consider that some or all of these will have an effect on the interfaces between the future OSG and campus research computing. Some of these technologies have been around for a while but are recently beginning to have a noticeable impact.

As just discussed, one technology is the Shibboleth identity management technology that is being used on an increasing number of campuses for campus-wide authentication; clearly it has an important role in the interface with users and newcomers to scientific computing. Coupled with other available technologies like GridShib and MyProxy one could potentially achieve a much simpler user experience around authentication than is used on OSG today. To be most effective there would need to be some policy work on the acceptance of credential stores at the IGTF level but one can consider if the international acceptance of credentials is really needed for all grid users. Clemson who is part of the InCommon federation has demonstrated peering with the NCSA Gridshib CA, which enabled Clemson faculty to obtain a short-lived NCSA proxy certificate which gave them access to TeraGrid resources.

Another technology that has been developing for years is virtual machines. These are used at a number of campuses in the “traditional” areas of server consolidation and resource management and are more recently being explored for more widespread scientific computing. From the end user perspective the key driver is being able to run the same application environment everywhere. From the resource providers perspective it is a way to provide application-specific resources on a temporary or shared basis without having to dedicate resources to individual applications. This is similar to the server consolidation concept but moving into broader application areas.

The third emerging technology, related to virtual machines, is cloud computing. The deployment of the Globus-Nimbus, Eucalyptus and Opennebula software as a means to provide a common interface across numerous distributed resources is showing to be amenable to many application deployment scenarios. A key driver is the relatively clean separation of resource management from application environment so the same resource can be quickly re-configured for different uses. In the scientific computing domain there are open questions and issues about the performance that can be achieved, primarily in the storage and network communications domains. The Magellan project at NERSC and ANL is focused on understanding the cost and performance issues of the cloud paradigm with respect to scientific computing and the relation of public and private clouds.