FERMILAB-PUB-07-323-CD

Challenges Facing Production Grids

Ruth PORDES
Fermi National Accelerator Laboratory[1],
PO Box 500, Batavia, Illinois, USA

Abstract: Today's global communities of users expect quality of service from distributed Grid systems equivalent to that their local data centers. This must be coupled to ubiquitous access to the ensemble of processing and storage resources across multiple Grid infrastructures. We are still facing significant challenges in meeting these expectations, especially in the underlying security, a sustainable and successful economic model, and smoothing the boundaries between administrative and technical domains. Using the Open Science Grid as an example, I examine the status and challenges of Grids operating in production today.

Keywords: Global Communities, Grids, Distributed Computing, High Throughput Computing, Opportunistic Computing

Introduction

Modern Grid infrastructures transfer up to a terabyte of data and run tens of thousands of processing jobs a day supporting not only "early adopter" physical science applications but also a broad range of research including nanotechnology, drugs and disease research, and the social sciences.

There are several large national and international Grids, for example, Enabling Grids for EsciencE (EGEE)[1], GridPP[2], Nordic DataGrid Facility[3], Open Science Grid[4], and TeraGrid[5]. These Grids provide access to and sharing of as much as five hundred Teraflops of processing power. They include as many as two hundred independent physical sites. They provide data storage using tape and disk caches of as much as ten PetaBytes. They support data movement across a range of production and research networks from a hundred Mbit to several tens of GBits. Each Grid serves from ten to a hundred independent user communities each consisting from a single to a hundred users, and many of which act across multiple infrastructures.

In 2008, the largest user communities are expected to run more than twenty thousand jobs a day across local and remote resources. This will involve moving data at sustained rates of up to ten Gigabit/second with latencies for the transfer of Terabyte datasets of less than a day. Each community will support hundreds of users doing data analysis in simultaneously. As these distributed facilities are relied on more and more, there is an accompanying expectation that they will be as robust, capable and available as local data centers.

[1] Fermilab is supported by the U.S. Department of Energy under contract No. DE-AC02-76CH03000.

Modern researchers depend on a rapidly increasing amount of computation and electronic data to make progress. Individuals, small research teams, and university groups, need to gain access to and learn to use remote resources. Usable common distributed infrastructures lower the barriers for these groups to benefit from distributed facilities as well as increase local productivity by enabling sharing of resources across the campus or local region.

1. The Context

The challenges facing production Grids today are threefold: First to provide high reliability, high throughput, scalable, multi-user, distributed data centers which operate around the clock and around the world; second to provide the security, technologies and infrastructure to serve an increasingly large and demanding community of researchers, educators, commercial companies and the general public; and third provide usable services that facilitate the entry of new participants in the use of distributed computational infrastructures.

Present experience where the Grid is driven by and embedded with the end user communities is encouraging in the success of matching expectation to realization and in the effectiveness of the operating infrastructure. This inclusion of all the actors – the users, the facility owners, and the technology providers, is a key component of the success. One of the larger distributed facilities, the Open Science Grid (OSG) Consortium, provides such an example. The scientists from the internal stakeholder communities – especially the Large Hadron Collider (LHC)[6] ATLAS and CMS experiments and the Laser Interferometer Gravitational Wave Observatory (LIGO)[7] – are immersed into the project in all aspects of the management and technical program. Similarly the computer science researchers and information technologists, who provide the software and services on which the infrastructure relies, participate fully in the leadership and all activities of the Consortium.

1.1. The Open Science Grid Distributed Facility

The Open Science Grid distributed facility, shown in Figure 1, includes: sites which provide shared storage and processing resources; communities, or Virtual Organizations[8], which include the users of the facility working in collaborating groups; and the common infrastructure which provides services to integrate and operate the facility as a whole. The infrastructure also provides gateways to other Grids.

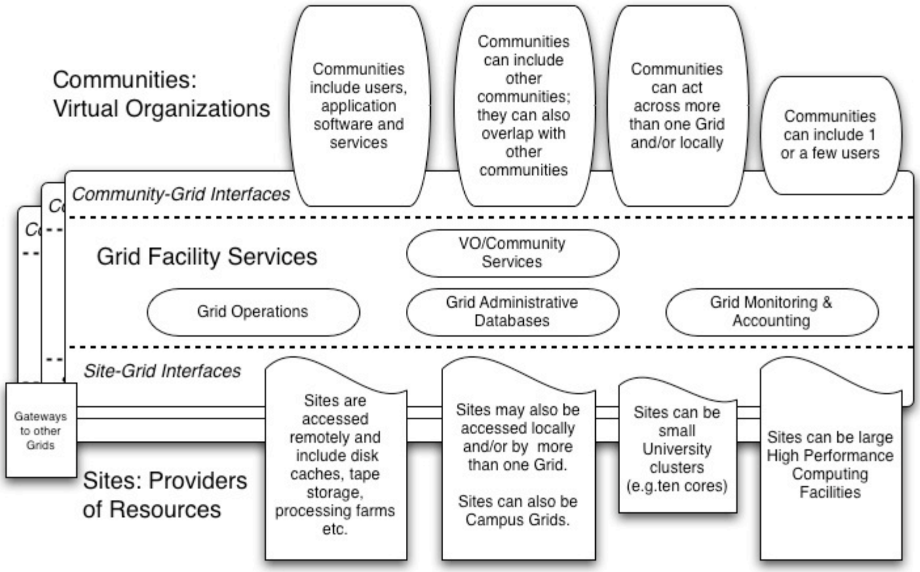

Figure 1: The OSG Facility

The following principles[9] guide the Open Science Grid's implementation:

- The owners of a site are responsible for, control, and manage access to, their resources.
- The community is responsible for its users and applications and their access to and use of the resources.
- The OSG staff is responsible for the common services and for federating with other distributed facilities including campus, regional, and international infrastructures.

Such a facility can only thrive in an environment and with a value proposition where:

- The economic model provides effective sharing so that the benefit experienced from the ensemble of resources is sufficiently greater than that delivered by the sum of each individual site.
- The security, policy and trust infrastructures demonstrably protect all participants from unintended as well as deliberate misuse and attack.

OSG today is a production infrastructure providing access to more than fifty processing sites and ten grid accessible storage sites, where space can be managed remotely, used by more than twenty VOs. The processing sites provide access to about thirty thousand cores providing over a hundred teraflops peak usage, with the size of the sites ranging from a few tens to a few thousand cores each. While the majority of the processing and storage resources are provided for the owners of the resource, on average more than ten percent are provided on a regular basis for other users – and on many days the "opportunistic" usage is significantly greater. In parallel with the production system is an integration or test infrastructure of about fifteen processing and two storage sites. The processing sites internally have a significant amount of both shared and local storage, with the total amount of disk accessible storage being of the order of a few petabytes.

1.2. Implementation Architecture

A production grid enables multiple VOs to use and share resources, software and services. The definition of an OSG VO includes not only the people but also the services and the resources used by the organization. And a VO may include sub-VOs within it. In the OSG context each VO acts as a "community grid" overlaid over one of more grids – there being a well-defined scope for the services, policies and procedures which make up the organization.

Within a grid there are a set of well-defined common interfaces between heterogeneous implementations for each component. These interfaces allow the distributed data center to act as a system and provide a manageable and smooth process for adding resources and new applications. Our architecture, which is typical for production grids today enables a resource to be accessed and used through multiple interfaces—locally through existing "non-grid' means and/or through other grids such as local shared campus infrastructures etc. Similarly, the implementation architecture is cognizant that any VO may be using multiple infrastructures simultaneously and may have a deep set of (sometimes complex) shared software and services that are specific to the VO and operate across these infrastructures. Other architectural aspects include:

- Shared resources provide not only agreed levels of use but also provide "opportunistic" access to otherwise unused resources.
- Each service or resource manages their own interfaces.
- The production facility supports multiple versions of the infrastructure simultaneously.
- There are no single points of failure.
- The distributed facility supports incremental upgrade and extensions.
- All components adhere to the facility security requirements and can trace their use and access to a responsible person. Access may be denied (to a VO or user) based on security as well as contract and policy requirements.
- Latency as well as performance needs influence the implementation.

1.3. Lacks in Implementation

Current implementations of the architecture are lacking in several dimensions. Extending the current practices can solve some of these, but others need research into new methods and techniques:

- The ability of the services and components to defend themselves against overload and unintentional misuse by the application software.
- The impossibility of testing all configurations in such a heterogeneous environment before putting new software into production.
- The lack of software to support dynamic sharing of storage on a distributed set of resources.
- The lack of the infrastructure to support recursive sub-VOs in a manner equivalent to VOs, including the OSG VO itself.
- End-to-end communication of, response to, and diagnosis of downtimes, failures and errors.

2. The Value Proposition

Large scientific collaborations control and own their vertically integrated data handling and analysis systems. This is essential to give them the ability to manage and make trade offs in the performance, functionality and scale of their systems to meet their scientific priorities and goals. And the nature of scientific collaboration today is intrinsically collaborative and widely distributed. Reliance on common infrastructures such as the OSG only succeed when there is a clear value which results in a reduction of effort or an increase in total throughput without compromising the end goals

An example benefit that has been widely acknowledged by the communities is the integration, testing and monitoring of the suite of software and the deployed system. Another is the sharing of expertise and gradual acceptance of common needs and software that can be shared. This latter is a cultural change, which can only come gradually. It is an example of one of the longer-term values of common and collaborative grids that is difficult to quantify but should be included in the tally.

Below we list some the value and benefits through deploying production grids for the science and research community:

- Distribution of peak demands across a larger ensemble of resources and on average increase the % usage of available compute and storage cycles;
- Reduction in the effort needed to adopt and use distributed systems;
- Reuse of software and reduction of effort applied to duplication;
- On the ground communication of knowledge and experience;
- Reduction in risk of failure due through a broader range of available experts and systems.

Using OSG as the example, the benefits indeed come from reducing the risk in and sharing support for large complex systems, which must be run for many years with a workforce whose availability is significantly shorter than the lifetime of the application projects. Leverage of the expertise and support for such systems to enable new communities to more easily participate in distributed science is an important additional benefit. This includes:

- Savings in effort for integration, system and software support,
- Opportunity and flexibility to distribute load and address peak needs.
- Maintenance of an experienced workforce in a common system.
- Lowering the cost of entry to new contributors.
- Enabling of new computational opportunities to communities that would not otherwise have access to such resources

Clearly, the existence of a shared virtual facility, aka grid, does not obviate the need for the purchase of hardware and building of computational facilities themselves. In fact when one calculates the basic costs of purchasing and maintaining hardware locally and compare it to the cost to interface and share say 10% of remote resources that one does not own, the results do not favour the latter. A more relevant calculation is the total lifetime cost of building and maintaining complex systems that scientists and researchers can effectively use.

2.1. Benefits from a Common, Integrated Software Stack

Most production grids provide a reference software stack for use by resource owners and application communities to not only ease their participation with the infrastructure but also to ensure a robust, reliable foundation to which the resources interface and on which the applications can act. This testing, large-scale use and support for this software that is demonstrably "used under fire" is one of the values of a recognizable common infrastructure, which is widely deployed.

Most scientific and research grids today rely on the security and execution management technologies from Condor[11] and Globus[12]. In addition, several rely on the additional extensions and integration offered by the OSG software releases – the Virtual Data Toolkit (VDT) [10]. The VDT includes about thirty additional modules, including components from other computer science groups, the Enabling Grids for EScience (EGEE), DOE Laboratory facilities (Brookhaven, Fermilab and LBNL), the application communities themselves, as well as standard open source software components such as Apache and Tomcat. And, while TeraGrid does not rely on VDT per se, OSG and TeraGrid align their versions of the Condor and Globus software to ensure ongoing interoperability. The EGEE gLite and TeraGrid CTSS software stacks are similar to the VDT in goals and scope, and provide an integrated set for those grids that interoperate with the VDT as well as other software systems.

Once installed on a site the VDT software enables support for remote job submission and local execution, access to remote storage and sharing of local storage, and data transfer into and out of the site. The site also has services to help the administrators manage priorities and access between VOs, support policies within the VOs, and can participate in the OSG monitoring, validation and accounting services.

The VDT also provides client libraries and tools for the applications to use to access the distributed resources and services.

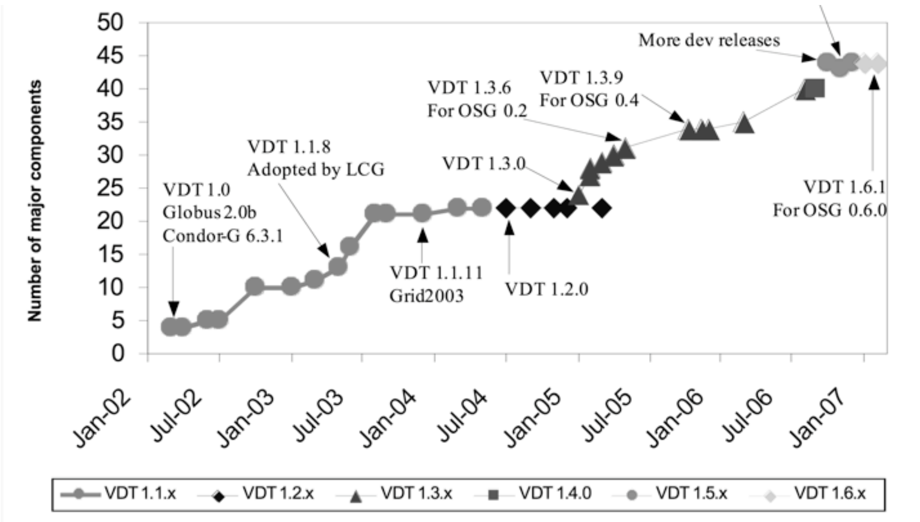

Figure 2: Timeline of VDT Releases

It is essential to maintain the operation and robustness of the production grid infrastructure to changes in software versions and capabilities. Production grids invariably institute a careful process in deploying new software and OSG is a prime example. Before a new release of the

VDT, all components are built up to ten Linux platforms (as well as some limited builds on AIX and MacOSX) using the NSF Middleware Initiative build and test facility [13] and, by re-using builds, the VDT tested on up to fifteen platforms. The validity of the integrated release is checked carefully. For example many components use Apache web server and we check that usage is compatible. The integrated release is validated on a small number of well-controlled sites to check it works in a distributed environment. It is then in at scale on a parallel grid to the production facility. In the OSG case this is a >15 site infrastructure, the Integration Testbed (ITB). The system tests only the individual services and components but also that the system and VO applications work reliably for long periods.

Most, if not all, the research communities' domain specific software and services layered above the production grid software stack. The goal is for them to install their software dynamically using Grid methods and either compile and leave the executables on each site, or move the executable for each job submission. Integration, or pre-production, grid goals are invariably to test the complete matrix of horizontally integrated grid software and vertically integrated VO software.

Software infrastructures provided by production grids also provide tools for incremental installation of new releases and patches, verification of installed configurations, and for functional testing of the sites. Security and robustness of the software as deployed is of paramount concern and an ongoing area of activity on alls production infrastructure. While progress is being made the following areas still merit research and development to make the support and use of these grids more secure, usable and dependable:

- Tools to audit the use of and probe the software interfaces to determine overload and "denial of service" conditions.

2.1.1. Challenges with Software Evolution and Support

The following needs for software support have been shown to be ongoing challenges:

- Quick and efficient patches of the software and redeployment in response to security notifications.
- Balancing the amount and effort spent on testing with the need to get new services "to the user" quickly. The stability of the resulting infrastructure becomes at risk since testing invariably does not cover the full set of usage patterns.
- Adding functionality (driven by the user communities) balanced with the need to make the software stack minimal and low impact.
- Integration of diverse software components from multiple software suppliers with different levels of development maturity and different release cycles.

3. An Economic Model

Computing and storage resources owners make their resources accessible to the OSG shared distributed facility and retain control of the management, use and policies of these resources (except of course in the response to security incidents) The OSG Consortium makes no policy on their use except that each resource contributor is expected to support at least one OSG application which is not the one owning the resources -- otherwise this would obviate the need or interest in being part of the OSG - as well as supporting the OSG administrative VOs for monitoring, accounting, validation.

Requests for production running are brought to the Executive Board to assess the technical needs and to the Council for an understanding of resource availability and policy. The

Consortium members define policies that allow users of the OSG to take advantage of available and otherwise unused cycles. The economic model of OSG has the goal that the ensemble offers an overall higher peak and average throughput than would come from the sum of individual and separate use of each of the resources; that sharing of resources is a win-win situation; and that we encourage sites to define policies for opportunistic use by a broad set of OSG member VOs wherever possible. In the long run OSG can only work if people give as well as receive. At the moment there are sufficient resources and the sum of the VO needs and ability to run on sites is such that OSG Council members have not yet had to take the hard decisions that we know are coming when there is oversubscription and there are many people knocking on the door.

3.1. A Practical example: Opportunistic Use

D0 is one of the two currently running experiments at the Tevatron in Batavia, Illinois. The experiment currently has more than two and a half petabytes of raw and processed data on archival tape. D0's own resources are committed to the processing of newly acquired data and analysis of these datasets as they are processed. In November 2006 D0 asked to use fifteen to twenty thousand CPUs for up to four months for re-processing of an existing dataset (~500 million events) to deliver science results for the summer conferences in July 2007. The OSG Executive Board estimated there were currently sufficient opportunistically available resources on OSG to meet the request; we also reviewed that the local storage and I/O needs could be met. The Council members agreed to contribute resources to meet this request. D0 had several months of smooth production running using more than a thousand CPUs and met their goal by the end of May.

The steps to achieving this included: D0 testing of the integrated software system from December to February; OSG staff and D0 working closely together as a team to reach the needed throughput goals during February and March; working through detailed problems at more than twelve OSG sites as well as sites on the EGEE, Canada and UK grids; support for sustaining the throughput during April and May. The goal was achieved by the end of May. OSG contributed over half of the total events processed (286 million events) using more than two million CPUhours (see Figure 4) of opportunistically available cycles by three hundred thousand jobs, the average job length being between six and eight hours. During the reprocessing 48TeraBytes of data were moved from the central archive at Fermilab to the OSG sites and 22TeraBytes was moved from the OSG reprocessing sites back to the central archive.

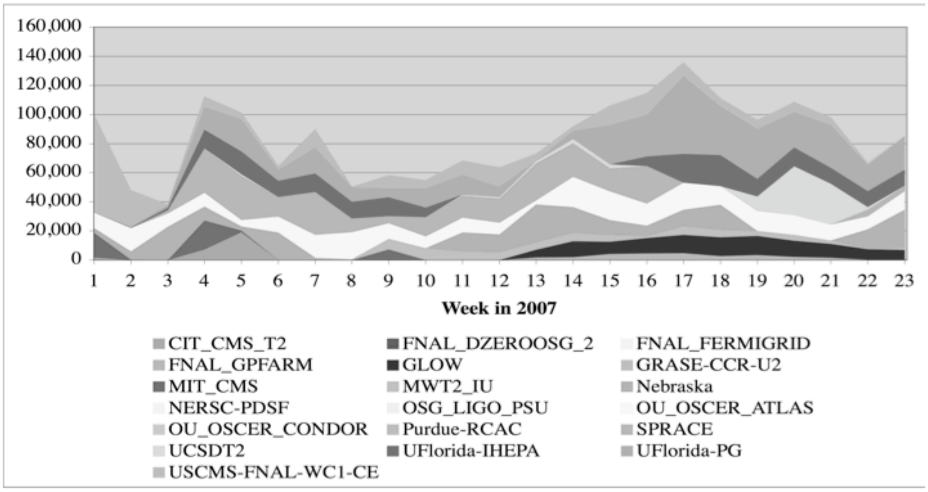

Figure 3: D0 CPUHours/Week on OSG in 2007

3.1.1. Lessons Learned

This production activity was a very instructive experience in learning how OSG as an organization, as well as an operating infrastructure, can handle such requests. The ramp up time, as well as the unique problems experienced at each site as the VO's application was commissioned, can clearly benefit from improvement. Additional problems and overheads were experienced scaling the whole system up to the throughput needed. As explained above we faced two main problems: Overall root cause analysis and troubleshooting of the quite different hardware configurations at the sites—especially data storage architectures—was a challenge; and during the steady state operations phase we found inefficiencies in our processes for notification and responding to site downtimes and problems.

While the activity was a success there are several issues that we are taking under consideration for the future:

- Sites do provide substantial sharing of their resources and VOs can rely on significant effective throughput from resources they do not own.
- Ongoing teamwork between the application and infrastructure groups is essential in such a complex software and hardware environment.
- We need to start preparing now for the future where the resources on OSG are oversubscribed and prioritization across the infrastructure is needed.

4. Production Grid Services

4.1. Security

The security challenges faced by distributed data centers extend beyond those for a fire walled single site. Most grids define the site and VO management as responsible for the security of the resources and services they own. The VO management is responsible for ensuring their individual users follow best practice and are aware of security and its needs.

The grid organization is responsible for the risk analysis, assessment and auditing of the assets it owns – the grid-wide services and the software it provides. Also the grid organization monitors and acts on risk and identified incidents across the whole infrastructure. In general, production grids do have active security activities that include security management, operational and technical components, agreed upon scope and authorities [14]. Communication of and timely response to security alerts is essential and the possibility of a major incident causes most grid managers loss of sleep.

As an indication of scale, OSG has responded to more than fifteen security alerts that involve software vulnerabilities over the past year. These invariably result in new software releases and significant effort is spent developing and distributing these patches in a timely fashion. Communication across the software development teams early in the process is crucial for fast in depth analysis of the problem, validation of proposed solutions and non-perturbative deployment.

To identify its users, resources and services the largest production grids in action today, (including EGEE and OSG) support X509 user, host and service certificates allocated through one of the International Grid Trust Federation (IGTF) [15] accredited Certificate Authorities. The grids provide services that extend these certificates by adding information about the VO the use of the certificate will be associated with for a particular set of transactions, and in some cases a finer granularity (or role) within the VO which will be used to determine the access policies and priorities that will be applied to the transactions themselves [16]. The grid infrastructure components map such certificate attributes to accounts, ACLs, and other access controls on sites and at the boundaries to services, as well as apply policy - including black- and white-listing.

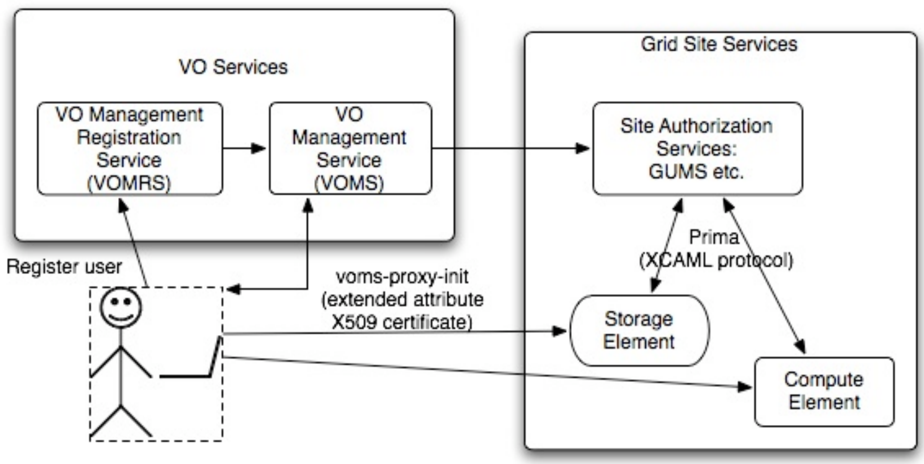

Figure 5: The OSG Authorization Infrastructure. The VO maintains user membership information, while Sites control access and privileges to resources.

This security architecture implements a model of trust between sites and VOs with the grid organization. It relies on delegated trust between the VO and the end users, which must also be audited and assessed. Integration of newly emerging campus identification and authorization framework, Shibboleth [17] is underway as an extension to the existing X509 based infrastructures.

4.1.1. Lacks in the Security Infrastructure

- Support for sub-VOs with inherited attributes and delegated trust is not fully available. Support for the extended attributes is restricted to a few services - notably processing, and soon storage.
- Mechanisms for auditing and tracing of the end-to-end grid access and transactions are lacking.
- Tools for tracing and assessing access across the whole ensemble of sites, VOs and services are not fully realised.
- The safe storage and integrity of the certificates is a significant concern.
- Delegation and use of proxy certificates is not fully supported by the end-to-end infrastructure.

4.2. Job Management and Execution

Sites present interfaces allowing remotely submitted jobs to be accepted, queued and executed locally. The priority and policies of execution are controllable by and at the site. Science driven infrastructures such as OSG and EGEE provide mechanisms for each research community to internally define priorities across particular groups and users and X509 certificate attribute extensions are used to support this.

Site policies and priorities are defined through mapping the user and attributes associated with their current transaction (see above) to specific accounts used to submit the job to a standard batch system. For example, OSG supports the Condor-G job submission client, which interfaces to either the pre-web service or web services GRAM Globus interface at the executing site. Job managers at the backend of the GRAM gatekeeper support job execution by local Condor, LSF, PBS, or SGE batch systems.

Of course, we expect that our distributed data center won't make the user have to select which remote system to submit their job to. Grid infrastructures offer a selection of automated resource selection services and meta-schedulers. The usefulness of such a service of course depends on the timeliness and accuracy of information about the state of the resources and their ability to successfully execute work that is sent to them. This is where the main challenges lie.

Many different such resource selectors exist. Using OSG as an example, there is support for stand-alone Condor match-making, a generalized Resource Selection Service from D0 and also interoperable support for the EGEE resource broker (RB) which provide user controllable mechanisms to automatically select to which site jobs are sent. Both RESS and the RB depend on the OSG site information services, which present information about the resources using the Glue Schema [17] attributes and providers and optionally converting the information to Condor Classads.

The user communities have realized that providing their own job management within the remote environment gives them more control over the prioritization and overall throughput of their applications. The largest science communities on OSG for example now use "pilot job" mechanisms. Pilot jobs are submitted through the normal job execution services. When the VO pilot job starts execution, using a standard batch slot, it interacts with its partner "VO job scheduler" to download the application executable to be run. The VO job scheduler controls the scheduling of jobs between the users in the VO and schedules jobs to run only on those resources that are immediately ready to execute them. These user jobs execute under the identity of the pilot job submitter—, which can break the policy that sites must be able to identify the end user of their resources. OSG has integrated an Apache suexec [19] derivative, the EGEE glexec module, that enables the pilot to run jobs under the identity of the originating user.

Typical job submission and workflow usage is through user or community portals provided by the VOs themselves. These are increasingly sophisticated to hide the complexity and increase the automated throughput of the infrastructure for the indiviual user.

Scalability and response to overload are two challenges of the current job execution infrastructures. One major challenge that has yet to be well addressed is the reporting of well described and identified error conditions, and associated tools to allow easy user debugging of problems.

4.3. Data Transport, Storage and Access

Many of the early adopters using production grids have large file based data transport and application level high data I/O needs. The data transport, access, and storage implementations must take account of these needs. Implementation of file transport that follow the Globus GridFTP protocol for the raw transport of the data is ubiquitous. Invariably Globus GridFTP itself is used except where interfaces to storage management systems (rather than file systems) dictate individual implementations.

Several of the large infrastructures, including EGEE and OSG support the Storage Resource Management (SRM) [20] interface to storage resources that enables management of space and data transfers to prevent unexpected errors due to running out of space, overload of the GridFTP services, and to provide capabilities for pre-staging, pinning and retention of the data files. While there are several implementations of SRM OSG, for example, currently provides reference implementations of two storage systems - the LBNL Disk Resource Manager and dCache [21]. In addition, because functionalities to support space reservation and sharing are not yet available through grid interfaces, OSG defines a set of environmental variables (see Figure 6)that a site must implement and a VO can rely on to point them to available space, space shared between all nodes on a compute cluster, and for the use of high-performance I/O disk caches. These environment variables are used to distinguish between smaller, more long-lived storage for applications, and large, higher throughput, more transient areas for placement of data.

4.3.1. Challenges in Data Storage and Access

The major challenges for data storage and access are to provide implementations of managed storage areas that:

- Are easy to install, configure and support;
- Support the full range of disk storage resource sizes from several to hundreds of TeraBytes;
- Provide guaranteed and opportunistic sharing of large amounts of space between and within VOs;
- Deliver hundred terabyte data sets on-time;
- Provide applications high throughput access to data at a local or remote site.

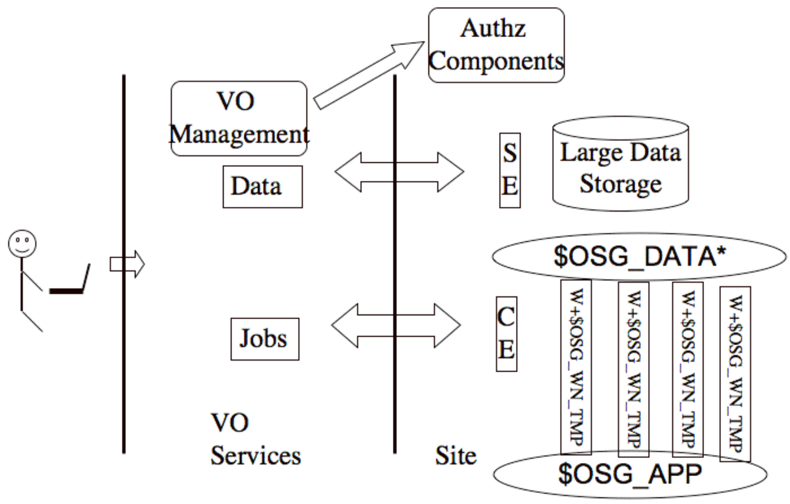

Figure 6: OSG Envrironment Variables for Storage Areas

4.4. Information Services

4.4.1. Service Status and Discovery

As mentioned above it must be accepted that information about the state and availability of the resources and services on a widely distributed and autonomously managed set of components will be incomplete and out of date. Reliance on this information must be cognizant of its quality and latency. A range of relatively static information can be used to dispatch work to sites, understand whether a configuration can support a particular application etc.

The idea of discovery of services and resources has long been a dream of transparent access to a distributed set of resources. While there have been some initial deployments a fully dynamic discovery service is not yet a reality. In general information is collected and checked against expected results and error conditions raised when the unexpected is noted.

4.4.2. Monitoring and Accounting

All production grids include, invariably many different, monitoring infrastructures. The VO , site and grid administrators and operations, the users and the management are all interested in many knowing the current and historical performance and load on the system. To enable discussion fo economics, accounting of the use of the resources is needed. This immediately brings us to an interesting question: If a resource is used it is clear whom to charge, but it may be that many different components have contributed to the value gained. If a job is scheduled through one grid infrastructure and executed through another – who accounts the job? Invariably, both do and this does indeed make sense.

One of the challenges of monitoring and accounting is to reduce the resource load due to the monitoring and accounting infrastructures themselves. The diversity and independence of the stakeholders in knowing the information, as well as the different views of to gather the information, make this a challenge in its own right.

4.5. Gateways to other Facilities and Grids

There is a rapid growth in the interest and deployment of shared computational infrastructures at the local and regional level. And in addition these is rapid growth in research communities' need to move data and jobs between heterogeneous facilities and build integrated community computational systems across high performance computing (HPC) facilities and more traditional computing clusters.

It was previously stated that the world is a "grid-of-grids" and interoperation and interfacing between them is part of the architecture and an operational goal. For the OSG the scope includes software and support for campus and regional organizations to form their own locally shared facilities and distributed infrastructures which gateway to the national grid. Three examples participating in OSG today are the Fermilab Facility Grid (FermiGrid) [22], the Grid Laboratory of Wisconsin (GLOW)[23] and the Purdue University campus-wide infrastructure. Each local organization contributes to and manages the gateway between the local facility and the OSG. The implementation principles of the gateways for the first two differs:

- On FermiGrid each VO interfaces their own resources directly to the OSG and then accesses them through their regular remote access mechanisms. FermiGrid interfaces a gateway to the OSG, which dispatches jobs from all other OSG VOs automatically across the complete set of resources (see Figure 7).
- GLOW provides a single interface to the OSG. Local job and data submissions are handled through the existing local mechanisms and then automatically "uploaded" to OSG resources under control of the GLOW gateway.

The campus infrastructures provide multiple access mechanisms to the processing nodes so that computing cluster appear as local Condor pools as well as being accessible by OSG and, in the case of Purdue, TeraGrid.

Also as stated above, the large production grids federate and supporting groups to submit jobs across and move data between them. For example, the OSG collects information from the resources and publishes them in the format needed by the EGEE. A VO "Resource Broker" or job dispatcher can then submit jobs transparently across resources on the two grids.

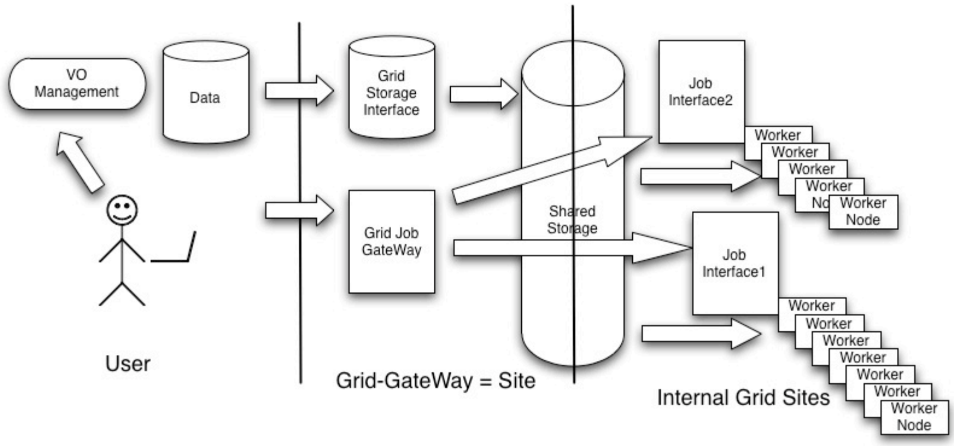

Figure 7: Access between Grids (example OSG -> FermiGrid).

5. Managing the Production Grid

Treating a production grid as a data center brings new insight and requirements into what is required to manage and operate them. Configuration management, service level management, dynamic resource control and so forth together with the life cycles of and relationships between the components become important to sustain a robust and evolutionary infrastructure. There are some unique challenges that the remoteness and autonomy of the components of the infrastructure bring forth.

Attention must also be paid to the provisioning and lifecycle of the components and thought given to what information is needed from all components, how to deal with a system where some components are always non-operational at any one time, how to reason given the latency in the information available, how to reconcile the accounting by different parties accessing information at different layers of the system, and how a uniform management can be layered onto a set of disparate entities not under direct control of the system itself. Such management includes monitoring as well as managing the state transitions of the services (autonomous in the OSG model). Relationships and dependencies need to be understood also. These can be described as objectives for the level of service (Service Level Objectives) which defines the goals for the components attributes in terms of up time, defense from overload, replicability etc. Measurement, control and life-cycle management of grid service components then become part of an overall management infrastructure [24], with a set of defined states and interfaces of its own and into which can be plugged new services as they are deployed.

6. Challenges for the Future

Today's production research and science grids are charting new territories in several areas: Community driven production support and evolution of large complex systems with coordination but no control over the "end-points"; Security and operations across autonomously owned and managed facilities which scale from small university department clusters to large leadership class high performance computing facilities; Transparent support and failover during update of software and services; Robustness against failure, overload and unintentional and intentional misuse; Development of a sustainable economic model for use and growth of the infrastructure.

Not just the number of jobs executed and the amount of data transferred and stored must measure success. Success has to be measured by the impact on scientific productivity and maturity of computation as a cornerstone, together with experiment and simulation, of the research portfolio. While many measurements are made and much data is being collected to help us determine the impact, we do not yet really understand how to translate and analyze this information to quantify value and benefit.

Acknowledgments

Operational production quality grids are due to the intellectual commitment and hard work of many people. The OSG Consortium, staff and collaborators are a wonderful team who has made the OSG implementation possible. This work is supported by the Office of Science, U.S. Department of Energy, SciDAC program under Contract DE-FC02-06ER41436 and the National Science Foundation Cooperative Agreement, PHY-0621704.

References

[1] Enabling Grids for EsciencE (EGEE) web site at http://www.eu-egee.org/
[2] GridPP home page at http://map.gridpp.ac.uk/
[3] NDGF home page at http://www.ndgf.org/index.php?page=about
[4] Open Science Grid home page at http://www.opensciencegrid.org.
[5] TeraGrid home page at http://www.teragrid.org/
[6] LHC home page: http://lhc.web.cern.ch/lhc/
[7] LIGO http://www.ligo.caltech.edu/
[8] Definition of Virtual Organization: http://en.wikipedia.org/wiki/Virtual_organization
[9] OSG Blueprint http://osg-docdb.opensciencegrid.org/0000/000018/005/OSG-Blueprintv0.10.pdf
[10] The Virtual Data Toolkit web site: http://vdt.cs.wisc.edu
[11] Condor ™project web site: www.cs.wisc.edu/condor
[12] Globus Tookit™ web site: www.globus.org/toolkit
[13] Metronome (formerly NMI build and test): http://nmi.cs.wisc.edu/
[14] OSG Security plan: http://osg-docdb.opensciencegrid.org/0003/000389/017/OSGSecurityPlan.doc
[15] International Grid Trust Federation: http://www.gridpma.org/
[16] Virtual Organization Management systems for EGEE and OSG: R. Alfieri, R. Cecchini et al: Managing Dynamic User Communities in a Grid of Autonomous Resources, Talk from the 2003 Computing in High Energy and Nuclear Physics (CHEP03), http://osg-docdb.opensciencegrid.org/0006/000623/001/Certificate%20Attribute%201.0.doc
[17] Shibboleth federated Single-SignOn and attribute exchange framework: http://shibboleth.internet2.edu/
[18] Glue Schema Open Grid Forum Working Group: http://forge.gridforum.org/sf/projects/glue-wg
[19] Apache suexec: http://httpd.apache.org/docs/1.3/suexec.html
[20] Storage Resource Management Collaboration: http://sdm.lbl.gov/indexproj.php?ProjectID=SRM
[21] dCache Consortium web site: www.dcache.org
[22] FermiGrid web site: http://fermigrid.fnal.gov
[23] Grid Laboratory of Wisconsin web site: www.cs.wisc.edu/condor/glow.
[24] EGA reference model: http://tinyurl.com/3d942b.