to be published in Grid Computing: Making the Global Infrastructure a Reality
Fran Berman, Geoffry Fox and Tony Hey Editors
John Wiley

Implementing Production Grids

William E. Johnston[a],
The NASA IPG Engineering Team[b], *and*
The DOE Science Grid Team[c]

Contents

[a] DOE, Lawrence Berkeley National Laboratory and NASA Ames Research Center – wejohnston@lbl.gov
[b] www.ipg.nasa.gov
[c] doesciencegrid.org

Abstract[a]

Starting from Section 2, "The Grid Context," we lay out our view of a Grid architecture, and this definition provides a structure for the subsequent detailed description. In particular we identify what it is that differentiates a Grid from other structures for distributed computing, e.g. hierarchical clusters. The question of what is a minimum set of Grid services – the Grid Common Services, the neck of the hourglass model – and what needs to be added to make the Grid usable for particular communities is stated. Issues of interoperability and heterogeneity are addressed, and these are perhaps the most important distinguishing features of a Grid.

Section 3, "The Anticipated Grid Usage Model Will Determine What Gets Deployed, and When," addresses the question of Grid building from the points of view of various different types of Grid usage. This is an important point because differing usage patterns require different middleware, this is why the distinction of a minimal common set of Grid services and tools is so important. The underlying case studies have a supercomputing background, and so attention is given to the problems of coupling and synchronicity of resources that are not required in other sorts of Grids, e.g. Data Grids and Grids based on the seti@home concept (e.g. Entropia). This is why interoperability is so important, different usages of Grids will result in different middleware, scheduling strategies and tools for collaboration. The work of the Global Grid Forum is vital in ensuring that standards are defined so that these can interoperate. Nobody is going to be able to produce a commercial product or a Grid-in-a-box that can address the requirements of all Grid usage patterns, indeed much of the strength of the Grid concept is that it clearly recognizes this. The Globus team, who are the basis for the Grid building work described here, understood this very well and have produced a toolkit of sufficient flexibility and robustness to allow building of many different types of Grid. Section 3 also analyses different data usage patterns in Data Grids, and this highlights the realization in Grid computing that the distribution of data is even more important than the distribution of computing resources, since the curation and storage of data is becoming a key issue in tera- and peta- scale computing. The importance of workflow management has also come to the fore. The integration of message passing with the Grid is discussed primarily in the context of MPICH-G2, which provides access to both highly optimized vendor supplied MPI for intra-machine communication and socket based communication for the inter-machine communication. It is important that Globus, as core essential middleware, can interoperate with the best tools from anywhere in the world and a few examples of this are given.

Section 4, "Grid Support for Collaboration," describes how the Grid Common Services promote collaboration via the mechanisms for enabling secure resource sharing in Virtual Organizations. The Access Grid has an important role in enabling the human side of such collaboration, and in the building of trust and working relationships in a VO, and is mentioned as an aside.

[a] The author is indebted to one of the reviewers whose extensive and useful review formed the basis of this Abstract.

Sections 5, "Building an Initial Multi-site, Computational and Data Grid," and 6, "Cross-Site Trust Management," provide an account of the detail of Grid building. The interaction of the sociology and working practices of the administrators and users of a Grid is integrated with the technical details of Grid deployment and certificate management. Some detail is provided on the building of an identity Certification Authority and the issues of interoperability that are raised here.

Section 7, "Transition to a Prototype-Production Grid," fill in the essential steps necessary for Grid building. Section 7.3, The Model for the Grid Information System, describes the issue of Grid Information Service mechanisms. The strengths of the Globus model for GIS, which has been built on top of extensive practical experimentation, are set out. The tools described here give the ability to build functional and large scale Grids for particular communities. Whether tools such as X500 naming will enable the very complex Grids which can cross national borders and multiple administrative systems and lead to genuinely Global Grids, is not yet clear, but with the tools described here many very useful Grids can be built.

Section 7.4, "Local Authorization," provides an account of the features of Globus mapfiles. Section 7.5, "Site Security Issues," highlights a serious issue with Globus and firewalls, namely the necessity to keep a range of ports open.

Sections 7.6 - 7.9 give advice on moving towards getting real users onto the Grid, including issues such as high performance networking and batch schedulers. Section 7.11, Data Management and Your Grid Service Model, provides insights where large scale data management is an important issue (likely to be the majority of Grids).

Section 7.10, "Grid Systems Administration Tools," discusses a little progress in Grid administration.

Section 7.12, "Take Good Care of the Users as Early as Possible," describes some of the things that can be done to ease the transition of users in a Grid environment. This includes some detail on a proxy certificate management service (MyProxy), since experience has shown that certificate handling is one of the big barriers to consumer acceptance of Grids. MyProxy also allows the flexibility of the Globus proxy delegation model to be exploited via advanced programming models and problem solving portals.

Section 8 provides some concluding remarks, and section 9 attempts to acknowledge the many people who have helped to make IPG and the DOE Science Girds successes.

Section 10 is an annotated bibliography that is intended to provide pointers to a lot of additional information, and to acknowledge that there is a lot of other work going on in Grids that is only mentioned in passing in this article.

1 Introduction: Lessons Learned for Building Large-Scale Grids

Over the past several years there have been a number of projects aimed at building "production" Grids. These Grids are intended to provide identified user communities with a rich, stable, and standard distributed computing environment. By "standard" and "Grids" we specifically mean Grids based on the common practice and standards coming out of the Global Grid Forum (GGF) (www.gridforum.org).

There are a number of projects around the world that are in various stages of putting together production Grids that are intended to provide this sort of persistent cyber infrastructure for science. Among these are the UK's e-Science program [1], the European DataGrid [2], NASA's

Information Power Grid [3], several Grids under the umbrella of the DOE Science Grid [4], and (at a somewhat earlier sage of development) the Asia-Pacific Grid [5].

In addition to these basic Grid infrastructure projects, there are a number of well advanced projects aimed at providing the sorts of higher-level Grid services that will be used directly by the scientific community. These include, for example, Ninf (A Network based Information Library for Global World-Wide Computing Infrastructure - [6, 7]) and GridLab [8].

This paper, however, addresses the specific and actual experiences gained in building NASA's IPG and DOE's Science Grids, both of which are targeted at infrastructure for large-scale, collaborative science, and access to large-scale computing and storage facilities.

The IPG project at NASA Ames [3] has integrated the operation of Grids into the NASA Advanced Supercomputing (NAS) production supercomputing environment and the computing environments at several other NASA Centers, and, together with some NASA "Grand Challenge" application projects, has been identifying and resolving issues that impede application use of Grids.

The DOE Science Grid [4] is implementing a prototype production environment at four DOE Labs and at the DOE Office of Science supercomputer center, NERSC [9]. It is addressing Grid issues for supporting large-scale, international, scientific collaborations.

This paper only describes the experience gained from deploying a specific set of software: Globus [10], Condor [11], SRB/MCAT [12], PBSPro [13], and a PKI authentication substrate [14-16]. That is, these suites of software have provided the implementation of the Grid functions used in the IPG and DOE Science Grids.

The Globus package was chose for several reasons:

- A clear, strong, and standards based security model
- Modular functions (not an all or nothing approach) providing all of the Grid Common Services, except general events
- A clear model for maintaining local control of resources that are incorporated into a Globus Grid
- A general design approach that allows a decentralized control and deployment of the software
- A demonstrated ability to accomplish large-scale Metacomputing (in particular, the SF-Express application in the Gusto testbed – see [17])
- Presence in supercomputing environments
- A clear commitment to open source
- Today, one would also have to add "market share"

Initially, Legion [18] and UNICORE [19] were also considered as starting points, but both of these failed to meet one or more of the selection criteria given above.

SRB and Condor were added because they provided specific, required functionality to the IPG Grid, and because we had the opportunity to promote their integration with Globus (which has happened over the course of the IPG project).

PBS was chosen because it was actively being developed in the NAS environment along with the IPG. Several functions were added to PBS over the course of the IPG project in order to support Grids.

Grid software beyond those provided by these suites are being defined by many organizations, most of which are involved in the GGF. Implementations are becoming available, and are being experimented with in the Grids being described here (e.g. the Grid monitoring and event framework of the Grid Monitoring Architecture Working Group [20]), and some of these projects will be mentioned in this paper. Never-the-less the software of the prototype-production Grids described in this paper is provided primarily by the aforementioned packages, and these provide the context of this discussion.

This paper recounts some of the lessons learned in the process of deploying these Grids, and provides an outline of the steps that have proven useful / necessary in order to deploy these sorts of Grids. This reflects the work of a substantial number of people, representatives of whom are acknowledged below.

The lessons fall into four general areas – deploying operational infrastructure (what has to be managed operationally to make Grids work), establishing cross site trust, dealing with Grid technology scaling issues, and listening to the users – and all of these will be discussed.

This paper is addressed to those that are setting up science oriented Grids, or who are considering doing so.

2 The Grid Context

"Grids" ([21, 22]) are an approach for building dynamically constructed problem solving environments using geographically and organizationally dispersed, high performance computing and data handling resources. Grids also provide important infrastructure supporting multi-institutional collaboration.

The overall motivation for most current large-scale, multi-institutional Grid projects is to enable the resource and human interactions that facilitate large-scale science and engineering such as aerospace systems design, high energy physics data analysis [23], climate research, large-scale remote instrument operation [9], collaborative astrophysics based on virtual observatories [24], etc. In this context, Grids are providing significant new capabilities to scientists and engineers by facilitating routine construction of information and collaboration based problem solving environments that are built on-demand from large pools of resources.

Functionally, Grids are tools, middleware, and services for:

- building the application frameworks that allow discipline scientists to express and manage the simulation, analysis, and data management aspects of overall problem solving
- providing a uniform and secure access to a wide variety of distributed computing and data resources
- supporting construction, management, and use of widely distributed application systems
- facilitating human collaboration through common security services, and resource and data sharing
- providing support for remote access to, and operation of, scientific and engineering instrumentation systems
- managing and operating this computing and data infrastructure as a persistent service

This is accomplished through two aspects: 1)A set of uniform software services that manage and provide access to heterogeneous, distributed resources; and 2) a widely deployed infrastructure. The software architecture of a Grid is depicted in Figure 1.

Grid software is not a single, monolithic package, but rather a collection of interoperating software packages. This is increasingly so as the Globus software is modularized and distributed as a collection of independent packages, and as other systems are integrated with basic Grid services.

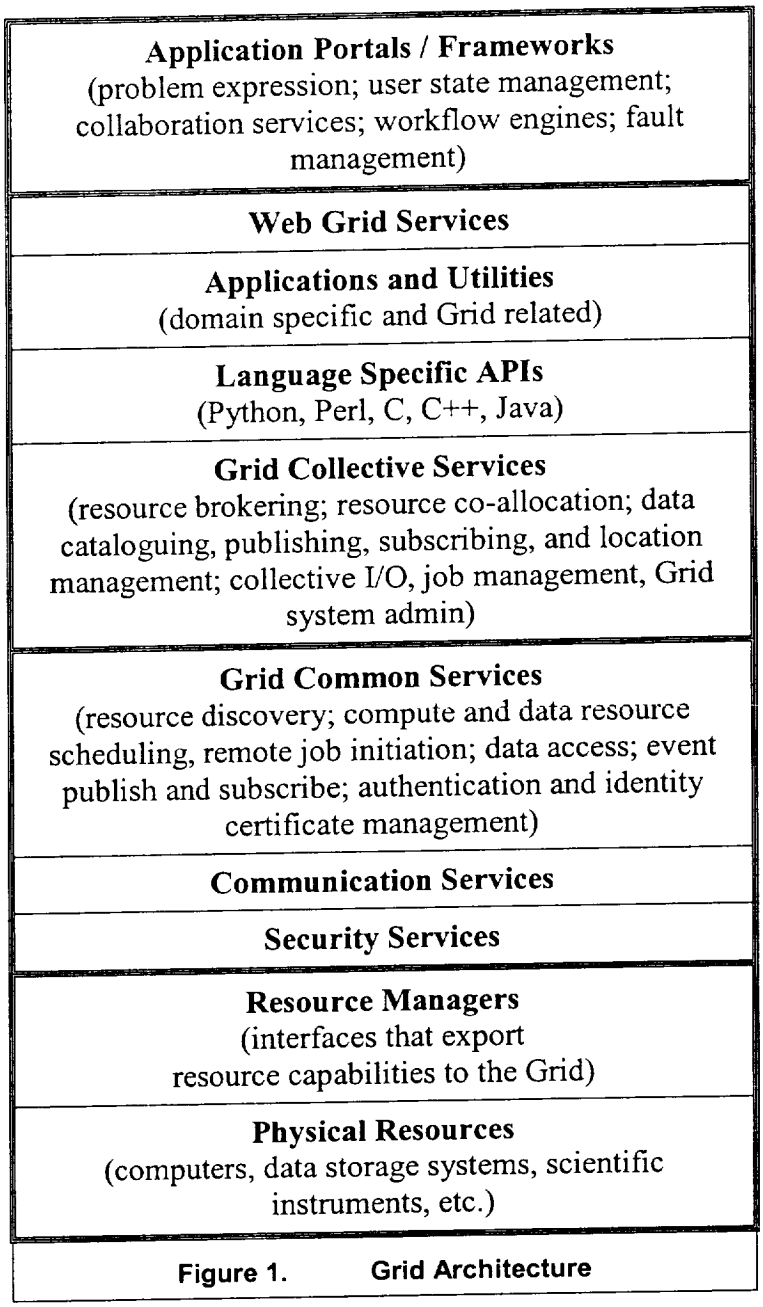

Figure 1. Grid Architecture

In the opinion of the author, there is a set of basic functions that all Grids must have in order to be called a Grid: The Grid Common Services. These constitute the "neck of the hourglass" of Grids, and include the Grid Information Service ("GIS" – the basic resource discovery mechanism) [25], the Grid Security Infrastructure ("GSI" – the tools and libraries that provide Grid security) [26], the Grid job initiator mechanism (e.g., Globus GRAM [27]), a Grid scheduling function, and a basic data management mechanism such as GridFTP [28]. It is almost certainly the case that to complete this set we need a Grid event mechanism. The Grid Forum's Grid Monitor Architecture [29] addresses one approach to Grid events, and there are several prototype implementations of the GMA (e.g. [30] and [31]). A communications abstraction (e.g., Globus I/O [32]) that incorporates Grid security is also in this set.

At the resource management level – which is typically provided by the individual computing system, data system, instrument, etc. – important Grid functionality is provided as part of the resource capabilities. For example, job management systems (e.g., PBSPro [13], Maui [33], and under some circumstances the Condor Glide-in [34] – see section 3.1.5) that support advance reservation of resource functions (e.g. CPU sets) are needed to support co-scheduling of administratively independent systems. This is because, in general, the Grid scheduler can request such service in a standard way, but cannot provide these services unless that are supported on the resources.

Beyond this basic set of capabilities (provided by the Globus toolkit [10] in this discussion) are associated client-side libraries and tools, and other high level capabilities such as Condor-G [35] for job management, SRB/MCAT [12] for federating and cataloguing tertiary data storage systems, and the new Data Grid [10, 36] tools for Grid data management.

In this paper, while we focus on the issues of building a Grid through deploying and managing the Grid Common Services (provided mostly by Globus), we also point out along the way other software suites that may be required for a functional Grid, and some of the production issues of these other suites.

3 The Anticipated Grid Usage Model Will Determine What Gets Deployed, and When

As noted, Grids are not built from a single piece of software, but from suites of increasingly interoperable software. Having some idea of the primary, or at least initial, uses of your Grid will help identify where you should focus you early deployment efforts. Considering the various models for computing and data management that might be used on your Grid is one way to select what software to install.

3.1 Grid Computing Models

There are a number of identifiable computing models in Grids that range from single resource to tightly coupled resources, and each requires some variations in Grid services. That is, while the basic Grid services provide all of the support needed to execute a distributed program, things like coordinated execution of multiple programs (as in high throughput computing) across multiple computing systems, or management of many thousands of parameter study or data analysis jobs, will require additional services.

3.1.1 Export Existing Services

Grids provide a uniform set of services to export the capabilities of existing computing facilities such as supercomputer centers to existing user communities, and this is accomplished by the Globus software. The primary advantage of this form of Grids is to provide a uniform view of several related computing systems, or to prepare for other types of uses. This sort of Grid also facilitates / encourages the incorporation of the supercomputers into user constructed systems.

By "user constructed systems" we mean, e.g., various sorts of portals or frameworks that run on user systems and provide for creating and managing related suites of Grid jobs. See, e.g., The GridPort Toolkit [37], Cactus [38, 39], JiPANG (A Jini-based Portal Augmenting Grids) [40], GridRPC [41], and in the future, NetSolve [42].

User constructed systems may also involve data collections that are generated and maintained on the user systems and that are used as input, e.g., to supercomputer processes running on the Grid, or are added to by these processes. The primary issue here is that a Grid compatible data service such as GridFTP must be installed and maintained on the user system in order to accommodate this use. The deployment and operational implications of this are discussed below, in section 7.11 "Data Management and Your Grid Service Model."

3.1.2 Loosely Coupled Processes

By loosely coupled processes we mean collections of logically related jobs that never-the-less do not have much in common once they are executing. That is, these jobs are given some input data that might, e.g., be a small piece of a single large dataset, and they generate some output data that may have to integrated with the output of other such jobs, however their execution is largely independent of the other jobs in the collection.

Two common types of such jobs are data analysis, where a large dataset is divided into units that can be analyzed independently, and parameter studies, where a design space of many parameters is explored, usually at low model resolution, across many different parameter values (e.g. [43] and [44]).

In the data analysis case, the output data must be collected and integrated into a single analysis, and this is sometimes done as part of the analysis job, and sometimes by collecting the data at the submitting site where the integration is dealt with. In the case of parameter studies, the situation is similar. The results of each run are typically used of fill in some sort of parameter matrix.

In both cases, in addition to the basic Grid services, a job manager is required to track these (typically numerous) related jobs in order to ensure either that they have all run exactly once, or that an accurate record is provided of those that ran and those that failed. (Whether the job manager can restart failed jobs typically depends on how the job is assigned work units or how it updates the results dataset at the end.)

The Condor-G job manager [35, 45] is a Grid task broker that provides this sort of service, as well as managing certain types of job dependencies.

Condor-G is a client-side service, and must be installed on the submitting systems. A *Condor_manager* server is started by the user, and then jobs are submitted to this user job manager. This manager deals with refreshing the proxy[a] that the Grid resource must have in order to run the user's jobs, but the user must supply new proxies to the Condor manager (typically once every 12 hours). The manager must stay alive while the jobs are running on the remote Grid resource in order to keep track of the jobs as they complete. There is also a Globus GASS server on the client side that manages the default data movement (binaries, stdin/out/err, etc.) for the job. Condor-G can recover from both server-side and client-side crashes, but not from long-term client-side outages. (That is, e.g., the client-side machine cannot be shutdown over the weekend while a lot of Grid jobs are being managed.)

This is also the job model being addressed by "peer-to-peer" systems. Establishing the relationship between peer-to-peer and Grids is a new work area at the GGF. See [46].

3.1.3 Workflow Managed Processes

The general problem of workflow management is a long way from being solved in the Grid environment, however it is quite common for existing application system frameworks to have ad hoc workflow management elements as part of the framework. (The "framework" runs the gamut from a collection of shell scripts to elaborate Web portals.)

One thing that most workflow managers have in common is the need to manage events of all sorts. By "event" we mean essentially any asynchronous message that is used for decision making purposes. Typical Grid events include:

- normal application occurrences that are used, e.g., to trigger computational steering or semi-interactive graphical analysis
- abnormal application occurrences, such as numerical convergence failure, that are used to trigger corrective action

[a] A proxy certificate is the indirect representation of the user that is derived from the Grid identity credential. The proxy is used to represent the authenticated user in interactions with remote systems where the user does not have a direct presence. That is, the user authenticates to the Grid once, and this authenticated identity is carried forward as needed to obtain authorization to use remote resources. This is called "single sign-on."

- messages that certain data files have been written and closed so that they may be used in some other processing step

Events can also be generated by the Grid remote job management system signaling various sorts of things that might happen in the control scripts of the Grid jobs, etc.

The Grid Forum, Grid Monitoring Architecture [29] defines an event model and management system that can provide this sort of functionality. Several prototype systems have been implemented and tested to the point where they could be useful prototype in a Grid. See, e.g., [30] and [31]. The GMA involves a server where the sources and sinks of events register, and these establish event channels directly between producer and consumer – i.e., it provides the event publish/subscribe service. This server has to be managed as a persistent service, however, in the future, it may be possible to use the GIS/MDS for this purpose.

3.1.4 Distributed-Pipelined / Coupled processes

In application systems that involve multidisciplinary or other multi-component simulations, it is very likely that the processes will need to be executed in a "pipeline" fashion. That is, there will be a set of interdependent processes that communicate data back and forth throughout the entire execution of the each process.

In this case co-scheduling is likely to be essential, as is good network bandwidth between the computing systems involved.

Co-scheduling for the Grid involves scheduling multiple individual, potentially architecturally and administratively heterogeneous, computing resources so that multiple processes are guaranteed to execute at the same time in order that they may communicate and coordinate with each other. This is quite different than co-scheduling within a "single" resource, such as a cluster, or within a set of (typically administratively homogeneous) machines all of which run one type of batch scheduler that can talk among themselves to co-schedule.

This coordinated scheduling typically accomplished by fixed time, or advance reservation scheduling in the underlying resources so that the Grid scheduling service can arrange for simultaneous execution of jobs on independent systems. There are currently a few batch scheduling systems that can provide for Grid co-scheduling, and this is typically accomplished by scheduling to a time-of-day. Both the PBSPro [13] and Maui Silver [33] schedulers provide time-of-day scheduling (see section 7.7). Other schedulers are slated to provide this capability in the future.

The Globus job initiator can pass through the information requesting a time-of-day reservation, however it does not currently include any automated mechanisms to establish communication among the processes once they are running. That must be handled in the higher level framework that initiates the co-scheduled jobs.

In this Grid computing model, network performance will also likely be a critical issue. See section 7.6 "High Performance Communications Issues."

3.1.5 Tightly Coupled Processes

MPI and PVM support a distributed memory programming model.

MPICH-G2 (the Globus enabled MPI) [47] provides for MPI style interprocess communication between Grid computing resources. It handles data conversion, communication establishment,

etc. Co-scheduling is essential for this to be a generally useful capability since different "parts" of the same program are running on different systems.

PVM [48] is another distributed memory programming system that can be used in conjunction with Condor and Globus to provide Grid functionality for running tightly coupled processes.

In the case of MPICH-G2, it can use Globus directly to co-schedule (assuming the underlying computing resource supports the capability) and coordinate communication among a set of tightly coupled processes. The MPICH-G2 libraries must be installed and tested on the Grid compute resources where they will be used. MPICH-G2 will use the manufacturer's MPI for local communication if one is available, and currently will not operate correctly if other versions of MPICH are installed. (Note that there was a significant change in the MPICH implementation between Globus 1.1.3 and 1.1.4 in that the use of the Nexus communication libraries was replaced by the Globus I/O libraries, and there is no compatibility between programs using Globus 1.1.3 and below, and 1.1.4 and above.) Note also that there are WAN version of MPI that are more mature that MPICH-G2 (e.g. PAXC-MPI [49], [50]), however, to the author's knowledge, these implementations are not Grid services because they do not make use of the Common Grid Services. In particular, the MIPCH-G2 use of the Globus I/O library that, e.g., automatically provides access to the Grid security services, since the I/O library incorporates GSI below the I/O interface.

In the case of PVM, one can use Condor to manage the communication and coordination. In Grids this can be accomplished using the Personal Condor Glide-In [34]. This is essentially an approach that has Condor using the Globus job initiator (GRAM) to start the Condor job manager on a Grid system (a "Glide-In"). Once the Condor Glide-In is started, then Condor can provide the communication management needed by PVM. PVM can also use Condor for co-scheduling (see the Condor User's Manual [51]), and then Condor, in turn, can use Globus job management. (The Condor Glide-In can provide co-scheduling within a Condor flock if it is running when the scheduling is needed. That is, it could drive a distributed simulation where some of the computational resources are under control of the user – e.g. a local cluster – and some (the Glide-in) are scheduled by a batch queuing system. However, if the Glide-in is not the "master" and co-scheduling is required, then the Glide-in itself must be co-scheduled using, e.g., PBS.) This, then, can provide a platform for running tightly coupled PVM jobs in Grid environments. (Note, however, that PVM does not use the has no mechanism to make use of the Grid Security Services, and so its communication cannot be authenticated within the context of the GSI.)

This same Condor Glide-In approach will work for MPI jobs.

The Condor Glide-In is essentially self installing: As part of the user initiating a Glide-In job, all of the required supporting pieces of Condor are copied to the remote system an installed in user-space.

3.2 Grid Data Models

Many of the current production Grids are focused around communities whose interest in wide area data management is at least as great as their interest in Grid based computing. These include, for example, Particle Physics Data Grid (PPDG) [52], Grid Physics Network (GriPhyN) [23], and the European Union DataGrid [36].

Like computing, there are several styles of data management in Grids, and these styles result in different requirements for the software of a Grid.

3.2.1 Occasional Access to Multiple Tertiary Storage Systems

Data mining, as, e.g., in [53], can require access to metadata and uniform access to multiple data archives.

SRB/MCAT provides capabilities that include uniform remote access to data, and local caching of the data for fast and/or multiple accesses. Through its metadata catalogue, SRB provides the ability to federate multiple tertiary storage systems (which is how it is used in the data mining system described in [53]). SRB provides a uniform interface by placing a server in front of (or as part of) the tertiary storage system. This server must directly access the tertiary storage system, so there are several variations depending on the particular storage system (e.g. HPSS, UniTree, DMF, etc.) The server should also have some local disk storage that it can manage for caching, etc. Access control in SRB is treated as an attribute of the dataset, and the equivalent of a Globus mapfile is stored in the dataset metadata in MCAT. See below for the operational issues of MCAT.

GridFTP provides many of the same basic data access capabilities as SRB, however for a single data source. GridFTP is intended to provide a standard, low-level Grid data access service so that higher level services like SRB could be componentized. However, much of the emphasis in GridFTP has been WAN performance and the ability to manage huge files in the wide area for the reasons given in the next section. The capabilities of GridFTP (not all of which are available yet, and many of which are also found in SRB) are also described in the next section.

GridFTP provides uniform access to tertiary storage in the same way that SRB does, and so there are customized backends for different type of tertiary storage systems. Also like SRB, the GridFTP server usually has to be managed on the tertiary storage system, together with the configuration and access control information needed to support GSI. (Like most Grid services, the GridFTP control and data channels are separated, and the control channel is always secured using GSI. See [54].)

The Globus Access to Secondary Storage service (GASS, [55]) provides a Unix I/O style access to remote files (by copying the entire file to the local system on file open, and back on close). Operations supported include read, write and append. GASS also provides for local caching of file so that they may be staged and accessed locally, and reused during a job without re-copying. That is, GASS provides a common view of a file cache within a single Globus job.

A typical configuration of GASS is to put a GASS server on or near a tertiary storage system. A second typical use is to locate a GASS server on a user system where files (such as simulation input files) are managed so that Grid jobs can access data directly on those systems.

The GASS server must be managed as a persistent service, together with the auxiliary information for GSI authentication (host and service certificates, Globus mapfile, etc.)

3.2.2 Distributed Analysis of Massive Datasets Followed by Cataloguing and Archiving

In many scientific disciplines, a large community of users requires remote access to large datasets. An effective technique for improving access speeds and reducing network loads can be to replicate frequently accessed datasets at locations chosen to be "near" the eventual users. However, organizing such replication so that it is both reliable and efficient can be a challenging problem, for a variety of reasons. The datasets to be moved can be large, so issues of network performance and fault tolerance become important. The individual locations at which replicas may be placed can have different performance characteristics, in which case users (or higher-level tools) may want to be able to discover

these characteristics and use this information to guide replica selection. In addition, different locations may have different access control policies that need to be respected.

From **A Replica Management Service for High-Performance Data Grids**, The Globus Project [56].

This quote characterizes the situation in a number of data intensive science disciplines, including high energy physics and astronomy. These disciplines are driving the development of data management tools for the Grid that provide naming and location transparency, and replica management for very large data sets. The Globus Data Grid tools include a replica catalogue ([57]), a replica manager ([58]) and a high performance data movement tool (GridFTP, [28]). The Globus tools do not currently provide metadata catalogues. (Most of the aforementioned projects already maintain their own style of metadata catalogue.) The European Union DataGrid project provides a similar service for replica management that uses a different set of catalogue and replica management tools (GDMP [59]). It, however, also uses GridFTP as the low-level data service. The differences in the two approaches are currently being resolved in a join US – EU Data Grid services committee.

Providing an operational replica service will involve maintaining both the replica manager service and the replica catalogue. In the long term, the replica catalogue will probably just be data elements in the GIS/MDS, but today it is likely to be a separate directory service. Both the replica manager and catalogue will be critical services in the science environments that rely on them for data management.

The data-intensive science applications noted above that are international in their scope have motivated the GridFTP emphasis on providing WAN high performance and the ability to manage huge files in the wide area. To accomplish this, GridFTP provides:

- integrated GSI security and policy-based access control
- third-party transfers (between GridFTP servers)
- wide area network communication parameter optimization
- partial file access
- reliability/restart for large file transfers
- integrated performance monitoring instrumentation
- network parallel transfer streams
- server side data striping (cf. DPSS [60] and HPSS striped tapes)
- server-side computation
- proxies (to address firewall and load balancing)

Note that the operations groups that run tertiary storage systems typically have (an appropriately) conservative view of their stewardship of the archival data, and getting GridFTP (or SRB) integrated with the tertiary storage system will take a lot of careful planning and negotiating.

3.2.3 Large Reference Data Sets

A common situation is that a whole set of simulations or data analysis programs will require the use of the same large, reference dataset. The management of such datasets, the originals of which almost always live in a tertiary storage system, could be handled by one of the replica managers. However, another service that is needed in this situation is a network cache: A unit of storage that can be accessed and allocated as a Grid resource, and that is located "close to" (in the network sense) the Grid computational resources that will run the codes that use the data. The Distributed Parallel Storage System (DPSS, [60]) can provide this functionality, however it is not currently well integrated with Globus.

3.2.4 Grid Metadata Management

The Metadata Catalogue of SRB/MCAT provides a powerful mechanism for managing all sorts of descriptive information about data: data content information, fine grained access control, physical storage device (which provides location independence for federating archives), etc.

The flip side of this is that the service is fairly heavy-weight to use (when its full capabilities are desired) and it requires considerable operational support. When the MCAT server is in a production environment where a lot of people will manage lots of data via SRB/MCAT, it requires a platform that typically consists of an Oracle DBMS running on a sizable multi-processor Sun system. This is a common installation in the commercial environment, however it is not typical in the science environment, and the cost and skills needed to support this in the scientific environment are non-trivial.

4 Grid Support for Collaboration

Currently, Grids support collaboration, in the form of Virtual Organizations (by which we mean human collaborators, together with the Grid environment that they share), in two very important ways.

The Grid Security Infrastructure provides a common authentication approach that is a basic and essential aspect of collaboration. It provides the authentication and communication mechanisms, and trust management (see section 6.1), that allow groups of remote collaborators to interact with each other in a trusted fashion, and it is the basis of policy based sharing of collaboration resources. GSI has the added advantage that it has been integrated with a number of tools that support collaboration, e.g. secure remote login and remote shell – GSISSH ([61] [62]), and secure ftp – GSI FTP ([62]), and GridFTP ([28]).

The second important contribution of Grids is that of supporting collaborations that are virtual organizations, and as such have to provide ways to preserve and share the organizational structure (e.g. the identities – as represented in X.509 certificates (see section 6) – of all of the participants and perhaps their roles), and share community information (e.g. the location and description of key data repositories, code repositories, etc.). For this to be effective over the long-term, there must be a persistent publication service where this information may be deposited and accessed by both humans and systems. The Grid Information Service can provide this service.

A third Grid collaboration service is the Access Grid [63] – a group-to-group audio and video conferencing facility that is based on Internet IP multicast, and can be managed by and out-of-band floor control service. The AG is currently being integrated with Globus directory and security services.

5 Building an Initial Multi-site, Computational and Data Grid

5.1 The Grid Building Team

Like networks, successful Grids involve almost as much sociology as technology, and therefore establishing good working relationships among all of the people involved is essential.

The concept of an Engineering Working Group ("WG") has proven successful as a mechanism for promoting cooperation and mutual technical support among those who will build and manage the Grid. The WG involves the Grid deployment teams at each site and meets weekly via

teleconference. There should be a designated WG lead responsible for the agenda and managing the discussions. If at all possible, involve some Globus experts at least during the first several months while people are coming up to speed. There should also be a WG mail list that is archived and indexed by thread. Notes from the WG meetings should be mailed to the list. This, then, provides a living archive of technical issues and the state of your Grid.

Grid software involves not only root owned processes on all of the resources, but also a trust model for authorizing users that is not typical. Local control of resources is maintained, but is managed a bit differently from current practice. It is therefore very important to set up liaisons with the system administrators for all systems that will provide computation and storage resources for your Grid. This is true whether or not these systems are within your organization.

5.2 Grid Resources

As early as possible in the process, identify the computing and storage resources to be incorporated into your Grid. In doing this be sensitive to the fact that opening up systems to Grid users may turn lightly or moderately loaded systems into heavily loaded systems. Batch schedulers may have to be installed on systems that previously did not use them in order to manage the increased load.

When choosing a batch scheduler, carefully consider the issue of co-scheduling! Many potential Grid applications need this, e.g., to use multiple Grid systems to run cross system MPI jobs or support pipelined applications as noted above, and only a few available schedulers currently provide the advance reservation mechanism that is used for Grid co-scheduling (e.g. PBSPro and Maui). If you plan to use some other scheduler be very careful to critically investigate any claims of supporting co-scheduling to make sure that they actually apply to heterogeneous Grid systems. (Several schedulers support co-scheduling only among schedulers of the same type and/or within administratively homogeneous domains.) See the discussion of the PBS scheduler in section 7.7, below.

5.3 Build the Initial Testbed

5.3.1 Grid Information Service

The Grid Information Service provides for locating resources based on the characteristics needed by a job (OS, CPU count, memory, etc.). The Globus Monitoring and Discovery Service (MDS) [25] provides this capability with two components. The Grid Resource Information Service (GRIS) runs on the Grid resources (computing and data systems) and handles the soft-state registration of the resource characteristics. The Grid Information Index Server (GIIS) is a user accessible directory server that supports searching for resource by characteristics. Other information may also be stored in the GIIS, and the GGF, Grid Information Services group is defining schema for various objects [64].

Plan for a GIIS at each distinct site with significant resources. This is important in order to avoid single points of failure, because if you depend on a GIIS at some other site, and it becomes unavailable, you will not be able to examine your local resources. Depending upon the number of local resources, it may be necessary to set up several GIISs at a site in order to accommodate the search load.

The initial testbed GIS model can be independent GIISs at each site. In this model, either cross-site searches require explicit knowledge of each of the GIISs that have to be searched

independently, or all resources cross-register in each GIIS. (Where a resource registers is a configuration parameter in the GRISs that run on each Grid resource.)

5.3.2 Build Globus on test systems

Use PKI authentication and initially use certificates from the Globus Certificate Authority ("CA"), or most any other CA that will issue you certificates for this test environment. (The OpenSSL CA [65] may be used for this testing.) Then validate access to, and operation of the GIS/GIISs at all sites, and test local and remote job submission using these certificates.

6 Cross-Site Trust Management

One of the most important contributions of Grids to supporting large-scale collaboration is the uniform Grid entity naming and authentication mechanisms provided by the Grid Security Infrastructure.

However, for this mechanism to be useful, the collaborating sites / organizations must establish mutual trust in the authentication process. The software mechanism of PKI, X.509 identity certificates, and their use in the GSI through TLS/SSL ([54]), are understood and largely accepted. The real issue is that of establishing trust in the process that each Certification Authority ("CA") uses for issuing the identity certificates to users and other entities, such as host systems and services. This involves two steps. First is the "physical" identification of the entities, verification of their association with the Virtual Organization that is issuing identity certificates, and then the assignment of an appropriate name. The second is the process by which an X.509 certificate is issued. Both of these steps are defined in CA policy.

In the PKI authentication environment assumed here, the CA policies are encoded as formal documents associated with the operation of the Certification Authority that issues your Grid identity credentials. These documents are called the Certificate Policy / Certification Practice Statement, and we will use "CP" to refer to them collectively. (See [66].)

6.1 Trust

Trust is "confidence in or reliance on some quality or attribute of a person or thing, or the truth of a statement."[a] Cyberspace trust starts with clear, transparent, negotiated, and documented policies associated with identity. When a Grid identity token (X.509 certificate in the current context) is presented for remote authentication and is verified using the appropriate cryptographic techniques, then the relying party should have some level of confidence that the person or entity that initiated the transaction is the person or entity that it is expected to be.

The nature of the policy associated with identity certificates depends a great deal on the nature of your Grid community and/or the Virtual Organizations associated with your Grid. It is relatively easy to establish policy for homogeneous communities, such as in a single organization, because an agreed upon trust model will likely already exist.

It is difficult to establish trust for large, heterogeneous virtual organizations involving people from multiple, international institutions, because the shared trust models do not exist. The typical issues related to establishing trust may be summarized as:

- Across administratively similar systems
  - e.g., within an organization

[a] Oxford English Dictionary, Second Edition (1989). Oxford University Press.

- informal / existing trust model can be extended to Grid authentication and authorization
- o Administratively diverse systems
  - e.g., across many similar organizations (e.g. NASA Centers, DOE Labs)
  - formal / existing trust model can be extended to Grid authentication and authorization
- o Administratively heterogeneous
  - e.g., cross multiple organizational types (e.g. science labs and industry)
  - e.g., international collaborations
  - formal / new trust model for Grid authentication and authorization will need to be developed

The process of getting your CP (and therefore your user's certificates) accepted by another Grids (or even by multi-site resources in your own Grid) involves identifying the people who can authorize remote users at all of the sites / organizations that you will collaborate with, and exchanging CPs with them. The CPs are evaluated by each party in order to ensure that local policy for remote user access is met. If it is not, then a period of negotiation ensues. The sorts of issues that are considered are indicated in the European Union DataGrid Acceptance and Feature matrices ([67]).

Hopefully the sites of interest already have people who are 1) familiar with the PKI CP process, and 2) focused on the scientific community of the institution rather than on the administrative community. (However, be careful that whomever you negotiate with actually has the authority to do so. Site security folks will almost always be involved at some point in the process, if that process it is appropriately institutionalized.)

Cross-site trust may, or may not, be published. Frequently it is. See, e.g., the European Union DataGrid list of acceptable CAs ([68]).

6.2 Establishing an Operational CA[a]

Set up, or identify, a Certification Authority to issue Grid X.509 identity certificates to users and hosts. Both the IPG and DOE Science Grids use the Netscape CMS software ([69]) for their operational CA because it is a mature product that allows a very scalable usage model that matches well with the needs of science Virtual Organizations.

Make sure that you understand the issues associated with the CP of your CA. As noted, one thing governed by CP is the "nature" of identity verification needed to issue a certificate, and this is a primary factor in determining who will be willing to accept your certificates as adequate authentication for resource access. Changing this aspect of your CP could well mean not just re-issuing all certificates, but requiring all users to re-apply for certificates.

Do not try and invent your own CP. The GGF is working on a standard set of CPs that can be used as templates, and the DOE Science Grid has developed a CP that supports international collaborations, and that is contributing to the evolution of the GGF CP. (The SciGrid CP is at http://www.doegrids.org/ [66].)

Think carefully about the space of entities for which you will have to issue certificates. These typically include human users, hosts (systems), services (e.g. GridFTP), and possibly security

[a] Much of the work described in this section is that of Tony Genovese (tony@es.net) and Mike Helm (helm@es.net), ESnet, Lawrence Berkeley National Laboratory.

domain gateways (e.g. the PKI to Kerberos gateway, KX509 [70]). Each of these must have a clear policy and procedure described in your CA's CP/CPS.

If you plan to interoperate with other CAs, then discussions about homogenizing the CPs and CPSs should begin as soon as possible, as this can be a lengthy process.

Establish and publish your Grid CP as soon as possible so that you will start to appreciate the issues involved.

6.2.1 Naming

One of the important issues in developing a CP is the naming of the principals (the "subject," i.e. the Grid entity identified by the certificate). While there is an almost universal tendency to try and pack a lot of information into the subject name (which is a multi-component, X.500 style name), increasingly there is an understanding that the less information of any kind put into a certificate, the better. This simplifies certificate management and re-issuance when users forget passphrases (which will happen with some frequency). More importantly, it emphasizes that ***all trust is local*** – that is, established by the resource owners and/or when joining a virtual community. The main reason for having a complicated subject name invariably turns out to be that people want to do some of the authorization based on the components of the name (e.g. organization). However, this usually leads to two problems. One is that people belong to multiple organizations, and the other is that the authorization implied by the issuing of a certificate will almost certainly collide with some aspect the authorization actually required at any given resource.

The CA run by ESnet (the DOE Office of Science scientific networks organization [71]) for the DOE Science Grid, for example, will serve several dozen different Virtual Organizations, several of which are international in their makeup. The certificates use what is essentially a flat namespace, with a "reasonable" common name (e.g. a "formal" human name) to which has been added a random string of alphanumeric digits to ensure name uniqueness.

However, if you do choose to use hierarchical institutional names in certificates, don't use colloquial names for institutions – consider their full organizational hierarchy in defining the naming hierarchy. Find out if anyone else in your institution, agency, university, etc., is working on PKI (most likely in the administrative or business units) and make sure that your names do not conflict with theirs, and if possible follow the same name hierarchy conventions

It should be pointed out that CAs set up by the business units of your organization frequently do not have the right policies to accommodate Grid users. This is not surprising since they are typically aimed at the management of institutional financial transactions.

6.2.2 The Certification Authority Model

There are several models for CAs, however increasingly associated groups of collaborations / Virtual Organizations are opting to find a single CA provider. The primary reason for this is that it is a formal and expensive process to operate a CA in such a way that it will be trusted by others.

One such model has a central CA that has an overall CP, and subordinate policies for a collection of VOs. The CA delegates to VOs (via Registration Agents) the responsibility of deciding who is a member of the particular VO, and how the subscriber / user will be identified in order to be issued a VO certificate. Each VO has an appendix in the CP that describes VO specific issues. VO Registration Agents are responsible for applying the CP identity policy to their users and other entities. Once satisfied, the RA authorizes the CA to issue (generate and sign) a certificate

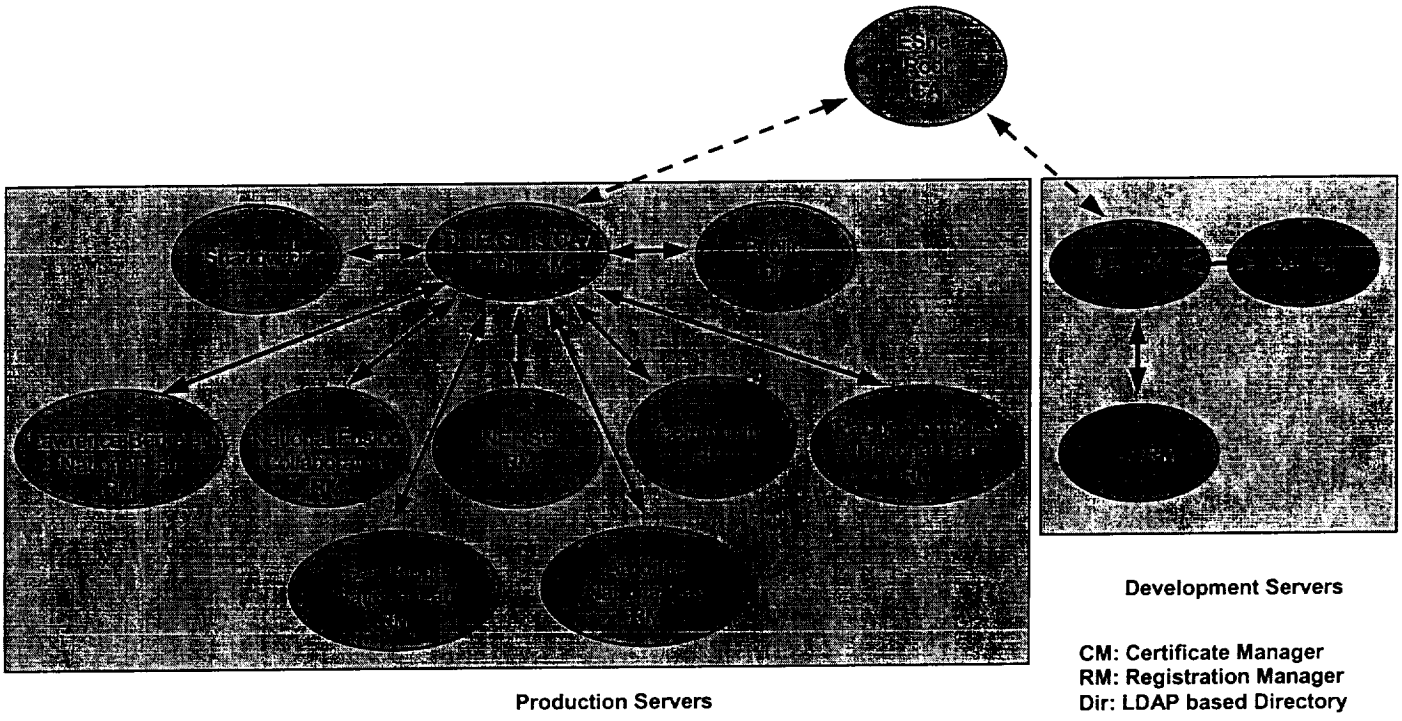

Figure 2. Software Architecture for 5/15/02 Deployment of the DOE Grids CA.
(Courtesy Tony Genovese (tony@es.net) and Mike Helm (helm@es.net), ESnet, Lawrence Berkeley National Laboratory.)

for the subscriber.

This is the model of the DOE Science Grid CA, for example, and it is intended to provide a CA that is scalable to dozens of Virtual Organizations and thousands of users. This approach to scalability is the usual divide and conquer, together with a hierarchical organization that maintains the policy integrity. The architecture of the DOE Science Grid CA is indicated in Figure 2, and has the following key features.

The Root CA (which is kept locked up and off-line) signs the certificates of the CA that issues user certs. With the exception of the "community" Registration Manager, all RMs are operated by the VOs that they represent. (The community RM addresses those "miscellaneous" people who legitimately need DOE Grid certificates, but for some reason are not associated with a

Virtual Organization.) The process of issuing a certificate to a user ("subscriber") is indicated in Figure 3.

ESnet [71] operates the CA infrastructure for DOE Science Grids, they do not interact with users. The VO RAs interface with certificate requestors. The overall policy oversight is provided by a Policy Management Authority, which is a committee that is chaired by ESnet, and is comprised of each RA, and a few others.

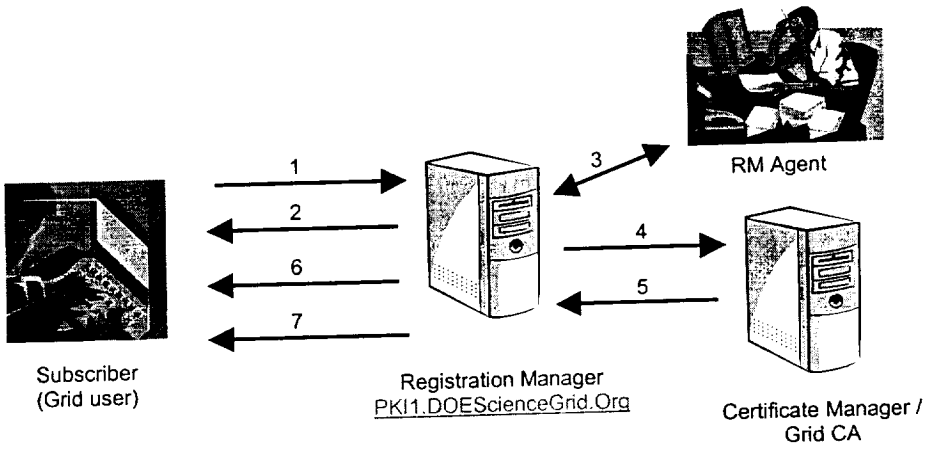

Figure 3. Certificate Issuing Process

(Courtesy Tony Genovese (tony@es.net) and Mike Helm (helm@es.net), ESnet, Lawrence Berkeley National Laboratory.

This approach uses an existing organization (ESnet) that is set up to run a secure, production infrastructure (its network management operation) to operate and protect the critical components of the CA. ESnet defers user contact to agents within the collaboration communities. In this case, the DOE Science Grid was fortunate in that ESnet personnel were also well versed in the issues of PKI and X.509 certificates, and so they were able to take a lead role in developing the Grid CA architecture and policy.

7 Transition to a Prototype-Production Grid

7.1 First Steps

Issue host certificates for all the computing and data resources and establish procedures for installing them. Issue user certificates.

Count on revoking and re-issuing all of the certificates at least once before going operational. This is inevitable if you have not previously operated a CA.

Using certificates issued by your CA, validate correct operation of the Grid Security Infrastructure (GSI) [72], GSS libraries, GSI-SSH [62], and GSI-FTP [73] and/or GridFTP [28] at all sites.

Start training a Grid application support team on this prototype.

7.2 Defining / Understanding the Extent of "Your" Grid

The "boundaries" of a Grid are primarily determined by three factors:

- Interoperability of the Grid software

  Many Grid sites run some variation of the Globus software, and there is fairly good interoperability between versions of Globus, so most Globus sites can potentially interoperate.

- What CAs you trust

  This is explicitly configured in each Globus environment on a per CA basis.

  Your trusted CAs establish the maximum extent of your user population, however there is no guarantee that every resource in what you think is "your" Grid trusts the same set of CAs – i.e. each resource potentially has a different space of users – this is a local decision. In fact, this will be the norm if the resources are involved in multiple virtual organizations as they frequently are, e.g., in the high energy physics experiment data analysis communities.

- How you scope the searching of the GIS/GIISs or control the information that is published in them

  This depends on the model that you choose for structuring your directory services.

So, the apparent "boundaries" of most Grids depend on who is answering the question.

7.3 The Model for the Grid Information System

Directory servers above the local GIISs (resource information servers) are an important scaling mechanism for several reasons. *paper).*

They expand the resource search space through automated cross-GIIS searches for resources, and therefore provide a potentially large collection of resources transparently to users. They also provide the potential for query optimization and query results caching. Further more, such directory services provide the possibility for hosting and/or defining virtual organizations, and for providing federated views of collections of data objects that reside in different storage systems.

There are currently two main approaches that are being used for building directory services above the local GIISs. One is a hierarchically structured set of directory servers and a managed namespace, al la X.500, and the other is "index" servers that provide ad-hoc, or virtual organization ("VO") specific, views of a specific set of other servers, such as a collection of GIISs, data collections, etc.

Both provide for "scoping" your Grid in terms of the resource search space, and in both cases many Grids use o=grid as the top level.

7.3.1 An X.500 Style Hierarchical Name Component Space Directory Structure

Using an X.500 Style hierarchical name component space directory structure has the advantage of organizationally meaningful names that represent a set of "natural" boundaries for scoping searches, and it also means that you can potentially use commercial metadirectory servers for better scaling.

Attaching virtual organization roots, data name spaces, etc., to the hierarchy makes them automatically visible, searchable, and in some sense "permanent" (because they are part of this managed name space).

If you plan to use this approach, try very hard to involve someone who has some X.500 experience because the directory structures are notoriously hard to get right, a situation that is compounded if VOs are included in the namespace.

7.3.2 Index Server Directory Structure

Using the Globus MDS ([25]) for the information directory hierarchy (see [74]) has several advantages.

The MDS research and development work has added to the usual LDAP based directory service capabilities several features that are important for Grids.

Soft-state registration provides for auto registration and de-registration, and for registration access control. This is very powerful. It keeps the information up-to-date (via a keep-alive mechanism) and it provides for a self configuring and dynamic Grid: A new resource registering for the first time is essentially no different than an old resource that is re-registering after, e.g., a system crash. The auto-registration mechanism also allows resources to participate in multiple information hierarchies, thereby easily accommodating membership in multiple VOs. The registration mechanism also provides a natural way to impose authorization on those who would register with your GIISs.

Every directory server from the GRIS on the resource, up to and including the root of the information hierarchy, is essentially the same, which simplifies the management of the servers.

Other characteristics of MDS include:

- resources are typically named using the components of their DNS name, which has the advantage of using an established and managed name space
- one must use separate "index" servers to define different relationships among GIISs, virtual organization, data collections, etc., on the other hand, this allows you to establish "arbitrary" relationships within the collection of indexed objects
- hierarchical GIISs (index nodes) are emerging as the preferred approach in the Grids community that uses the Globus software.

Apart from the fact that all of the directory servers must be run as persistent services and their configuration maintained, the only real issue with this approach is that we do not have a lot of experience with scaling this to multiple hierarchies with thousands of resources.

7.4 Local Authorization

As of yet there is no standard authorization mechanism for Grids. Almost all current Grid software uses some form of access control lists ("ACL"), which is straightforward, but typically does not scale very well.

The Globus mapfile is an ACL that maps from Grid identities (the subject names in the identity certificates) to local UIDs on the systems where jobs are to be run. The Globus Gatekeeper ([27]) replaces the usual login authorization mechanism for Grid based access, and uses the mapfile to authorize access to resources after authentication. Therefore, managing the contents of the mapfile is the basic Globus user authorization mechanism for the local resource.

The mapfile mechanism is fine in that it provides a clear-cut way for locally controlling access to a system by Grid users. However, it is bad in that for a large number of resources, especially if they all have slightly different authorization policies, it can be difficult to manage.

The first step in the mapfile management process is usually to establish a connection between user account generation on individual platforms and requests for Globus access on those systems. That is, generating mapfile entries is done automatically when the Grid user goes through the account request process. If your Grid users are to be automatically given accounts on a lot of different systems with the same usage policy, it may make sense to centrally manage the mapfile and periodically distribute it to all systems. However, unless the systems are administratively homogeneous, a non-intrusive mechanism, such as email to the responsible system admins to modify the mapfile, is best.

The Globus mapfile also allows a many-to-one mapping so that, e.g., a whole group of Grid users can be mapped to a single account. Whether or not the individual identity is preserved for accounting purposes is typically dependent on whether the batch queuing system can pass the Grid identity (which is carried along with a job request, regardless of the mapfile mapping) back to the accounting system. PBSPro, e.g., will provide this capability (see section 7.7).

One way to address the issues of mapfile management and disaggregated accounting within an administrative realm is to use the Community Authorization Service (CAS), which is just now being tested. See the notes at [75].

7.5 Site Security Issues

Incorporating any computing resource into a distributed application system via Grid services involves using a whole collection of IP communication ports that are otherwise not used. If your systems are behind a firewall, then these ports are almost certainly blocked, and you will have to negotiate with the site security folks to open the required ports.

Globus can be configured to use a restricted range of ports, but it still needs several tens, or so (depending on the level of usage of the resources behind the firewall), in the mid 700s. A Globus "port catalogue" is available to tell what each Globus port is used for, and this lets you provide information that your site security folks will likely want to know. It will also let you estimate how many ports have to be opened (how many per process, per resource, etc.). Additionally, GIS/GIIS needs some ports open, and the CA typically uses a secure Web interface (port 443). The Globus port inventory is given in [72]. The DOE Science Grid is in the process of defining Grid firewall policy document that we hope will serve the same role as the CA Certificate Practices Statement: It will lay out the conditions for establishing trust between the Grid

administrators and the site security folks who are responsible for maintaining firewalls for site cyber protection.

It is important to develop tools/procedures to periodically check that the ports remain open. Unless you have a very clear understanding with the network security folks, the Grid ports will be closed by the first network engineer that looks at the router configuration files and has not been told why these non-standards ports are open.

Alternate approaches to firewalls have various sorts of service proxies manage the intra service component communication so that one, or no, new ports are used. One interesting version of this approach that was developed for Globus 1.1.2 by Yoshio Tanaka at the Electrotechnical Laboratory (ETL, which is now the National Institute of Advanced Industrial Science and Technology (AIST)) in Tsukuba, Japan, is documented in [76] and [77].

7.6 High Performance Communications Issues

If you anticipate high data-rate distributed applications, whether for large-scale data movement or process-to-process communication, then enlist the help of a WAN networking specialist and check and refine the network bandwidth end-to-end using large packet size test data streams. (Lots of problem that can affect distributed application do not show up by pinging with the typical 32 byte packets.) Problems are likely between application host and site LAN/WAN gateways, WAN/WAN gateways, and along any path that traverses the commodity Internet.

Considerable experience exists in the DOE Science Grid in detecting and correcting these sort of problems, both in the areas of diagnostics and tuning.

End-to-end monitoring libraries/toolkits (e.g. NetLogger [78]) and pipechar [79]) are invaluable for application-level distributed debugging. NetLogger provides for detailed data path analysis, top-to-bottom (application to NIC) and end-to-end (across the entire network path) and is used extensively in the DOE Grid for this purpose. It is also being incorporated into some of the Globus tools. (For some dramatic examples of the use of NetLogger to debug performance problem in distributed applications see [80], [81], [82], and [83].)

If at all possible, provide network monitors capable of monitoring specific TCP flows and returning that information to the application for the purposes of performance debugging. (See, e.g., [84]).

In addition to identifying problems in network and system hardware and configurations, there are a whole set of issue relating to how current TCP algorithms work, and how they must be tuned in order to achieve high performance in high-speed, wide area networks. Increasingly, techniques for automatic or semi-automatic setting of various TCP parameters based on monitored network characteristics, are being used to relieve the user of having to deal with this complex area of network tuning that is critically important for high performance distributed applications. See, e.g., [85], [86], and [87].

7.7 Batch Schedulers[a]

There are several functions that are important to Grids that Grid middleware cannot emulate: these must be provided by the resources themselves.

[a] Thanks to Bill Nitzberg (bill@computer.org), one of the PBS developers and Area co-Director for the GGF Scheduling and Resource Management Area, for contributing to this section.

Some of the most important of these are the functions associated with job initiation and management on the remote computing resources. Development of the PBS batch scheduling system was an active part of the IPG project, and several important features were added in order to support Grids.

In addition to the scheduler providing a good interface for Globus GRAM/RLS (which PBS did), one of the things that we found was that people can become quite attached to the specific syntax of the scheduling system. In order to accommodate this PBS was componentized and the user interfaces and client-side process manager functions were packaged separately and interfaced to Globus for job submission.

PBS was somewhat unique in this regard, and it enabled PBS-managed jobs to be run on Globus-managed systems, as well as the reverse. This lets users use the PBS front-end utilities (submit via PBS 'qsub' command-line and 'xpbs' GUI, monitor via PBS 'qstat', and control via PBS 'qdel', etc.) to run jobs on remote systems managed by Globus. At the time, and probably today, the PBS interface was a more friendly option than writing Globus RSL.

This approach is also supported in Condor-G, which, in effect, provides a Condor interface to Globus.

PBS can provide time-of-day based advanced reservation. It actually creates a queue that "owns" the reservation. As such, all the access control features (allowing/disallowing specific users/groups) can be used to control access to the reservation. It also allows one to submit a string of jobs to be run during the reservation. In fact, you can use the existing job-chaining features in PBS to do complex operations like: run X; if X fails, run Y; if X succeeds, run Z.

PBS passes the Grid user ID back to the accounting system. This is important for allowing, e.g., the possibility of mapping all Grid users to a single account (and thereby not having to create actual user accounts for Grid user) but at the same time still maintaining individual accountability, typically for allocation management.

Finally, PBS supports access-controlled, high priority queues. This is of interest in scenarios where you might have to "commandeer" a lot of resources in a hurry to address a specific, potentially emergency, situation. Let us say for example that we have a collection of Grid machines that have been designated for disaster response / management. For this to be accomplished transparently, we need both lots of Grid managed resources, and ones that have high priority queues that are accessible to a small number of pre-approved people who can submit "emergency" jobs. For immediate response this means that they would need to be pre-authorized for the use of these queues, and that PBS has to do per queue, UID based access control. Further, these should be pre-emptive high-priority queues. That is, when a job shows up in the queue, it forces other, running, jobs to be checkpointed and rolled out, and/or killed, in order to make sure that the high priority job runs.

PBS has full "pre-emption" capabilities, and that, combined with the existing access control mechanisms, provides this sort of "disaster response" scheduling capability.

There is a configurable "preemption threshold' – if a queue's priority is higher than the preemption threshold, then any jobs ready to run in that queue will preempt all running work on the system with lower priority. This means you can actually have multiple levels of preemption. The preemption action can be configured to: a) Try to checkpoint, b) Suspend, and/or c) Kill and requeue, in any order.

For access control, every queue in PBS has an access control list that can include and exclude specific users and groups. All the usual stuff is supported, e.g., "everyone except bill", "groups foo and bar, but not joe", etc.

7.8 Preparing for Users

Try and find problems before your users do. Design test and validation suites that exercise your Grid in the same way that applications are likely to use your Grid.

As early as possible in the construction of your Grid, identify some test case distributed applications that require reasonable bandwidth, and run them across as many widely separated systems in your Grid as possible, and then run these test cases every time something changes in your configuration.

Establish user help mechanisms, including a Grid user email list and a trouble ticket system. Provide user oriented Web pages with pointers to documentation, including a Globus "Quick Start Guide" [88] that is modified to be specific to your Grid, and with examples that will work in your environment (starting with a Grid "hello world" example).

7.9 Moving from Testbed to Prototype Production Grid

At this point Globus, the GIS/MDS, and the security infrastructure should all be operational on the testbed system(s). The Globus deployment team should be familiar with the install and operation issues, and the system admins of the target resources should be engaged.

Deploy and build Globus on at least two production computing platforms at two different sites. Establish the relationship between Globus job submission and the local batch schedulers (one queue, several queues, a Globus queue, etc.)

Validate operation of this configuration.

7.10 Grid Systems Administration Tools[a]

Grids present special challenges for system administration due to the administratively heterogeneous nature of the underlying resources.

In the DOE Science Grid we have build Grid monitoring tools from Grid services. We have developed pyGlobus modules for the NetSaint [89] system monitoring framework that test GSIftp, MDS and the Globus gatekeeper. We have plans for, but have not yet implemented, a GUI tool that will use these modules to allow an admin to quickly test functionality of a particular host.

The harder issues in Grid Admin tools revolve around authorization and privilege management across site boundaries. So far we have concentrated only tools for identifying problems. We still use email to a privileged local user on the broken machine in order to fix things. Longer term we have been thinking about a framework that will use a more autonomic model for continuous monitoring and restart of services.

In both Grids, tools and techniques are being developed for extending Trouble Ticket based problem tracking systems to the Grid environment.

---

[a] Thanks to Keith Jackson (krjackson@lbl.gov) and Stephen Chan (sychan@lbl.gov) for contributing to this section.

In the future, we will have to evolve a Grid account system that tracks Grid user usage across a large number of machines and manages allocations in accordance with (probably varying) policy on the different systems. Some work by Jarosław Nabrzyski and his colleagues at the Poznan Supercomputing and Networking Center [90] in Poland is developing prototypes in this area. See [91].

7.11 Data Management and Your Grid Service Model

Establish the model for moving data between *all* of the systems involved in your Grid.

GridFTP servers should be deployed on the Grid computing platforms and on the Grid data storage platforms.

This presents special difficulties when data resides on user systems that are not usually Grid resources, and raises the general issue of your Grid "service model:" What services are necessary to support in order to achieve a Grid that is useful for applications, but are outside of your core Grid resources (e.g. GridFTP on user data systems), and how you will support these services, are issues that have to be recognized and addressed.

Determine if any user systems will manage user data that are to be used in Grid jobs. This is common in the scientific environment where individual groups will manage their experiment data, e.g., on their own systems. If user systems will manage data, then the GridFTP server should be installed on those systems so that data may be moved from user system to user job on

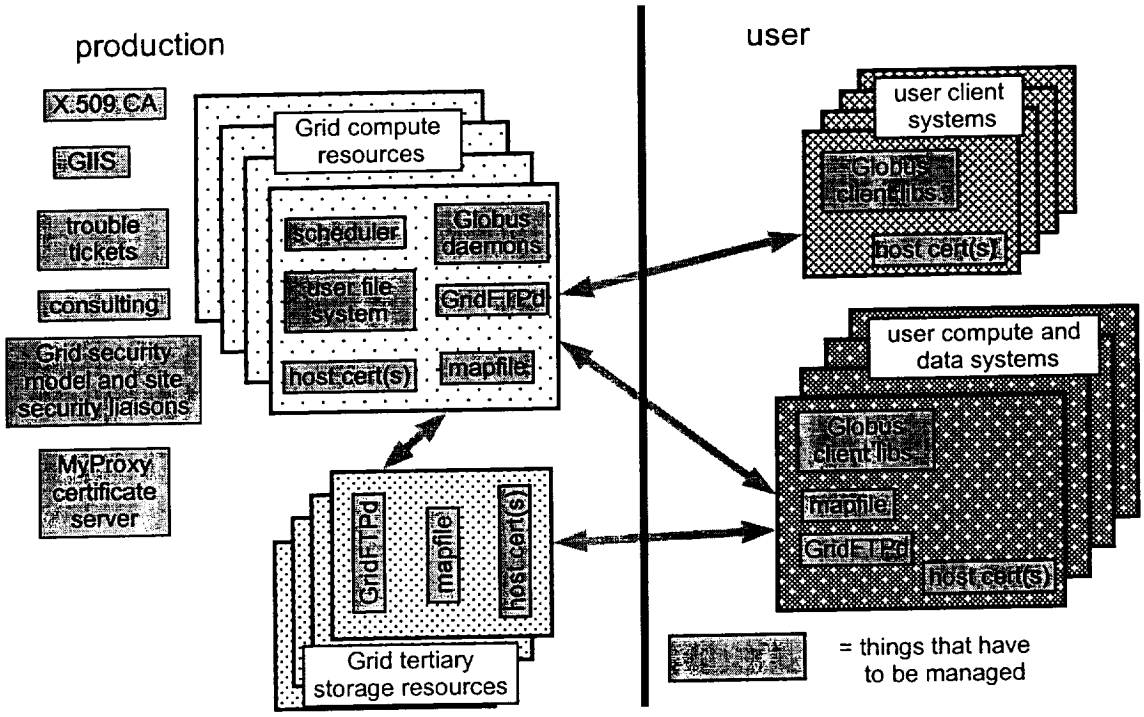

Figure 4. Establish Your Grid Service Model

the computing platform, and back.

Offering GridFTP on user systems may be essential, however managing long lived / root access Grid components on user systems may be "tricky" and/or require you to provide some level of system admin on user systems.

Validate that all of the data paths work correctly.

These issues are summarized in Figure 4.

7.12 Take Good Care of the Users as Early as Possible

If at all possible, establish a Grid/Globus application specialist group. This group should be running sample jobs as soon as the testbed is stable, and certainly as soon as the prototype-production system is operational. They should be able to assist generally with building Grid distributed applications, and specifically should serve as the interface between users and the Grid system administrators in order to solve Grid related application problems.

Identify specific early users and have the Grid application specialists encourage / assist them in getting jobs running on the Grid

One of the scaling / impediment-to-use issues currently is that extant Grid functions are relatively primitive (i.e., at a low level). This is being addressed by Grid middleware at various levels that provide aggregate functionality, more conveniently packaged functionality, toolkits for building Grid based portals, etc. Examples of such work in progress includes the Web Grid Services (e.g. the Open Grid Services Architecture [92] and the resulting Open Grid Services Interface [93] work at GGF), the Grid Web services testbed of the GGF GCE Working Group [94]), diverse interfaces to Grid functions (e.g., PyGlobus [95], CoG Kits [96], [97], and [98]), and the Grid Portal Development Kit [99]).

One approach that we have seen to be successful in the IPG and DOE Science Grid is to encourage applications that already have their own "frameworks" to port those frameworks on the Grid. This is typically not too difficult because many of these frameworks already have some form of resource management built in, and this is easily replaced / augmented with Grid resource management functions. This hides some of the "low level functionality" problem. Examples of such frameworks deployed in IPG and/or DOE Science Grid are NPSS [100, 101] and Cactus [38]. Another example of this approach is Ninf [6].

Another useful tool for users is a Grid job tracking and monitoring portal. IPG's LaunchPad [102] and NPACIs HotPage [103].

7.12.1 MyProxy Service

Consider providing a MyProxy service ([104] [105]) to simplify user management of certificates.

A frequent Grid user complaint relates to the constant management of GSI credentials, and the frequent necessity of generating proxy credentials so that Grid work can proceed. A related, and functionally more serious issue, is that in order to minimize the risk of the relatively unprotected proxy credentials their lifetimes are kept relatively short (typically 12 hours). This can create significant problems when, e.g., large jobs take longer than that to execute on remote computing systems, or if the batch queues are long, and the proxies expire before execution starts. In either case the job is likely to fail.

The MyProxy service is designed to alleviate these problems, as well as to ease the problem of trying to move the user's permanent identity credential to all of the systems from which the user will want to access the Grid.

The MyProxy service provides for creating and storing intermediate lifetime proxies that may be accessed by, e.g., Web based portals, job schedulers, etc., on behalf of the user. There are plans to extend the service so that it can manage the user's permanent identity credential as well.

MyProxy provides a set of client tools that let the user create, store, and destroy proxies, and for programs acting on behalf of the user to obtain valid (short term) proxies. The user can create a proxy with a lifetime of a week, or a few weeks, then store that proxy on a MyProxy server. The user and the MyProxy service establish a shared secret, and the user passes that secret to processes that need to obtain proxies on the user's behalf. In this way, a Grid service such as the Globus job initiator or the Condor job manager, can, after getting the user's access secret for MyProxy, contact the MyProxy service each time that they need a short term proxy to perform a task. Now when a Grid job manager finds that a job's proxy is about to expire, it can ask the MyProxy service for a new proxy without user intervention.

The user still has to supply proxy generating authority to MyProxy, but much less often than to a usual Grid task.

The security risks of this are analyzed in [104], and are found to be not only acceptable compared with direct user management of the short-lived proxies, but perhaps even less risky since the process is much less user error prone.

A key operational issue is that not only does the MyProxy server have to be managed as a persistent service, but as a secure persistent service. This means that it should probably be in a controlled physical environment (e.g. a controlled access machine room), should be a strictly single purpose system, and should probably be behind a content filtering firewall.

8 Conclusions

We have presented the essence of experience gained in building two production Grids, and provided some of the global context for this work.

As the reader might imagine, there were a lot of false starts, refinements to the approaches and to the software, and several substantial integration projects (SRB and Condor integrated with Globus) to get where we are today.

However, the point of this paper is to try and make it substantially easier for others to get to the point where IPG and the DOE Science Grids are today. This is what is needed in order to move us toward the vision of a common cyber infrastructure for science.

The author would also like to remind the readers that this paper primarily represents the actual experiences that resulted from specific architectural and software choices during the design and implementation of these two Grids. The choices made were dictated by the criteria laid out in section 1.

There is a lot more Grid software available today that there was four years ago, and various of these packages are being integrated into IPG and the DOE Grids.

However, the foundation choices of Globus, SRB, and Condor would not be significantly different today than they were four years ago. Nonetheless, if the GGF is successful in its work – and we have every reason to believe that it will be – then in a few years we will see that the

functions provided by these packages will be defined in terms of protocols and APIs, and there will be several robust implementations available for each of the basic components, especially the Grid Common Services.

The impact of the emerging Web Grid Services work is not yet clear. It will likely have a substantial impact on building higher level services, however it is the opinion of the author that this will in no way obviate the need for the Grid Common Services. These are the foundation of Grids, and the focus of almost all of the operational and persistent infrastructure aspects of Grids.

9 Acknowledgements

The experience represented in this paper is the result by a lot of hard work by the NASA and DOE Science Grid Engineering Teams.

The principals in the NASA IPG team are Tony Lisotta, Chair, Warren Smith, George Myers, Judith Utley, and formerly Mary Hultquist, all of the NASA Ames Research Center, and Isaac Lopez, of the NASA Glenn Research Center. This project is lead by William Johnston, Tom Hinke, and Arsi Vaziri of NASA Ames Research Center.

The principals in the DOE Science Grid Engineering team are Keith Jackson, Chair, Lawrence Berkeley National Laboratory; Tony Genovese and Mike Helm, ESnet; Von Welch, Argonne National Laboratory; Steve Chan, NERSC; Kasidit Chanchio, Oak Ridge National Laboratory, and; Scott Studham, Pacific Northwest National Laboratory. This project is lead by William E. Johnston, Lawrence Berkeley National Laboratory; Ray Bair, Pacific Northwest National Laboratory; Ian Foster, Argonne National Laboratory; Al Geist, Oak Ridge National Laboratory, and; William Kramer, LBNL / NERSC.

The IPG work is funded by NASA's Aero-Space Enterprise, Computing, Information, and Communication Technologies (CICT) Program (formerly the Information Technology), Computing, Networking, and Information Systems Project. Eugene Tu, Jerry Yan, and Cathy Schulbach are the NASA program managers.

The DOE Science Grid work is funded by the U.S. Dept. of Energy, Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Sciences Division (http://www.sc.doe.gov/ascr/mics/) under contract DE-AC03-76SF00098 with the University of California. This program office is lead by Walt Polansky. Mary Anne Scott is the Grids program manager and George Seweryniak is the ESnet program manager.

While not directly involved in funding the NASA or DOE Grids work, the author would also like to acknowledge the important support for Grids (e.g. the NSF Middleware Initiative [106]) provided by the National Science Foundation, in work funded by Alan Blatecky.

Without the support and commitment of program managers like these, and their counterparts in Europe and Asia Pacific, we would have little chance of realizing the vision of building a new and common cyber infrastructure for science.

Credit is also due the intellectual leaders of the major software projects that formed the basis of IPG and the DOE Science Grid. The Globus team is led by Ian Foster, University of Chicago and Argonne National Laboratory, and by Carl Kesselman, Univ. of Southern Calif., Information Sciences Institute. The SRB/MCAT team is led by Reagan Moore of the San Diego Supercomputer Center. The Condor project is led by Miron Livny at the Univ. of Wisc., Madison.

Special thanks goes to Bill Feiereisen, who as NASA HPCC program manager conceived of the Information Power Grid, and coined the term Grid in this context. While NAS Division Chief at NASA Ames, he provided the unfailing support for the project that was necessary for its success. Bill is currently head of the Computing Division at Los Alamos National Laboratory.

Important contributions are also being made by industry, and in the author's opinion among the most important of these is the support of Grid tools like the Globus toolkit by computing systems manufacturers. This support – like the recently announced IBM support for Globus on the SP/AIX supercomputers [107], and Fujitsu's support for Globus on its VPP series supercomputers – will be very important for Grids as sustainable infrastructure.

The author would also like to thank the several reviewers that took the time to read drafts of this paper and to provide useful comments that improved the final version. One reviewer in particular made extensive and very useful comments, and that review is the basis of the Abstract.

10 Notes and References

[1] **UK eScience Program**. http://www.research-councils.ac.uk/escience/

In November 2000 the Director General of Research Councils, Dr John Taylor, announced £98M funding for a new UK e-Science programme. The allocations were £3M to the ESRC, £7M to the NERC, £8M each to the BBSRC and the MRC, £17M to EPSRC and £26M to PPARC. In addition, £5M was awarded to CLRC to 'Grid Enable' their experimental facilities and £9M was allocated towards the purchase of a new Teraflop scale HPC system. A sum of £15M was allocated to a Core e-Science Programme, a cross-Council activity to develop and broker generic technology solutions and generic middleware to enable e-Science and form the basis for new commercial e-business software. The £15M funding from the OST for the core e-Science Programme has been enhanced by an allocation of a further £20M from the CII Directorate of the DTI which will be matched by a further £15M from industry. The Core e-Science Programme will be managed by EPSRC on behalf of all the Research Councils.

The e-Science Programme will be overseen by a Steering Committee chaired by Professor David Wallace, Vice-Chancellor of Loughborough University. Professor Tony Hey, previously Dean of Engineering at the University of Southampton, has been seconded to EPSRC as Director of the e-Science Core Programme.

What is meant by e-Science? In the future, e-Science will refer to the large scale science that will increasingly be carried out through distributed global collaborations enabled by the Internet. Typically, a feature of such collaborative scientific enterprises is that they will require access to very large data collections, very large scale computing resources and high performance visualisation back to the individual user scientists.

The World Wide Web gave us access to information on Web pages written in html anywhere on the Internet. A much more powerful infrastructure is needed to support e-Science. Besides information stored in Web pages, scientists will need easy access to expensive remote facilities, to computing resources - either as dedicated Teraflop computers or cheap collections of PCs - and to information stored in dedicated databases.

The Grid is an architecture proposed to bring all these issues together and make a reality of such a vision for e-Science. Ian Foster and Carl Kesselman, inventors of the Globus approach to the Grid define the Grid as an enabler for Virtual Organisations: 'An infrastructure that enables flexible, secure, coordinated resource sharing among dynamic collections of individuals,

institutions and resources.' It is important to recognize that resource in this context includes computational systems and data storage and specialized experimental facilities.

[2] **EU DataGrid Project**. www.eu-datagrid.org/

DataGrid is a project funded by European Union. The objective is to build the next generation computing infrastructure providing intensive computation and analysis of shared large-scale databases, from hundreds of TeraBytes to PetaBytes, across widely distributed scientific communities.

[3] **NASA's Information Power Grid**. http://www.ipg.nasa.gov

The Information Power Grid (IPG) is NASA's high performance computational grid. Computational grids are persistent networked environments that integrate geographically distributed supercomputers, large databases, and high end instruments. These resources are managed by diverse organizations in widespread locations, and shared by researchers from many different institutions. The IPG is a collaborative effort between NASA Ames, NASA Glenn, and NASA Langley Research Centers, and the NSF PACI programs at SDSC and NCSA.

[4] **DOE Science Grid**. http://www.doesciencegrid.org

The DOE Science Grid's major objective is to provide the advanced distributed computing infrastructure based on Grid middleware and tools to enable the degree of scalability in scientific computing necessary for DOE to accomplish its missions in science.

[5] **AP Grid**. http://www.apgrid.org/

ApGrid is a partnership for Grid computing in the Asia Pacific region. ApGrid focuses on (1) sharing resources (2) developing Grid technologies (3) helping the use of our technologies in create new applications (4) building on each other work, etc., and ApGrid is not restricted to just a few developed countries, neither to a specific network nor its related group of researchers.

[6] **Ninf: A Network based Information Library for Global World-Wide Computing Infrastructure**. http://ninf.apgrid.org/welcome.shtml

Ninf is an ongoing global network-wide computing infrastructure project which allows users to access computational resources including hardware, software and scientific data distributed across a wide area network with an easy-to-use interface. Ninf is intended not only to exploit high performance in network parallel computing, but also to provide high quality numerical computation services and accesses to scientific database published by other researchers. Computational resources are shared as Ninf remote libraries executable at a remote Ninf server. Users can build an application by calling the libraries with the Ninf Remote Procedure Call, which is designed to provide a programming interface similar to conventional function calls in existing languages, and is tailored for scientific computation. In order to facilitate location transparency and network-wide parallelism, Ninf metaserver maintains global resource information regarding computational server and databases, allocating and scheduling coarse-grained computation to achieve good global load balancing. Ninf also interfaces with existing network service such as the WWW for easy accessibility.

[7] **NetCFD: a Ninf CFD component for Global Computing, and its Java applet GUI**, M. Sato, K. Kusano, H. Nakada, S. Sekiguchi and S. Matsuoka. In *Proc. of HPC Asia 2000.* 2000.http://ninf.apgrid.org/papers/hpcasia00msato/HPCAsia2000-netCFD.pdf

Ninf is a middleware for building a global computing system in wide area network environments. We designed and implemented a Ninf computational component, netCFD for CFD (Computational Fluid Dynamics). The Ninf Remote Procedure Call (RPC) provides an interface to a parallel CFD program running on any high performance platforms. The netCFD turns high performance platforms such as supercomputers and clusters into valuable components for use in global computing. Our experiment shows that the overhead of a remote netCFD computation for a typical application was about 10% comparing with its conventional local execution. The netCFD applet GUI which is loaded in a web browser allows a remote user to control and visualize the CFD computation results interactively.

[8] **GridLab: A Grid Application Toolkit and Testbed**. http://www.gridlab.org/

The GridLab project is currently running being funded under the Fifth Call of the Information Society Technology (IST) Program.

The GridLab project will develop a easy-to-use, flexible, generic and modular Grid Application Toolkit (GAT), enabling todays applications to make innovative use of global computing resources. The project [has two thrusts]

1. Co-development of Infrastructure and Applications: We [undertake] a balanced program with co-development of a range of Grid applications (based on Cactus, the leading, widely used Grid-enabled open source application framework, and Triana, a dataflow framework used in gravitational wave research) alongside infrastructure development, working on transatlantic testbeds of varied supercomputers and clusters. This practical approach ensures that the developed software truly enables easy and efficient use of Grid resources in a real environment. We [are] maintain[ing] and upgrad[ing] the testbeds through deployment of new infrastructure and large scale application technologies as they are developed. All deliverables will be immediately prototyped and continuously field tested by several user communities. Our focus on specific application frameworks allows us immediately to create working Grid applications to gain experience for more generic components developed during the project.

2. Dynamic Grid Computing: We [are developing] capabilities for simulation and visualization codes to be self aware of the changing Grid environment, and to be able to fully exploit dynamic resources for fundamentally new and innovative applications scenarios. For example, the applications themselves will possess the capability to migrate from site to site during the execution, both in whole or in part, to spawn related tasks, and to acquire/release additional resources demanded by both the changing availabilities of Grid resources, and the needs of the applications themselves.

This timely and exciting project will join together the following institutions and businesses: Poznan Supercomputing and Networking Center (PSNC), Poznan, Poland (Project Coordinator) Max-Planck Institut fuer Gravitationsphysik (AEI), Golm/Potsdam, Germany. Konrad-Zuse-Zentrum fuer Informationstechnik (ZIB), Berlin, Germany Masaryk University, Brno, Czech Republic MTA SZTAKI, Budapest, Hungary Vrije Universiteit (VU), Amsterdam, The Netherlands ISUFI/High Performance Computing Center (ISUFI/HPCC), Lecce, Italy Cardiff University, Cardiff, Wales National Technical University of Athens (NTUA), Athens, Greece University of Chicago, Chicago, USA Information Sciences Institute (ISI), Los Angeles, USA University of Wisconsin, Wisconsin, USA Sun Microsystems Gridware GmbH Compaq Computer EMEA

[9] **National Energy Research Scientific Computing Center.** www.nersc.gov

NERSC is one of the largest unclassified scientific supercomputer centers in the US. It's mission is to accelerate the pace of scientific discovery in the DOE Office of Science community by providing high-performance computing, information, and communications services. NERSC is the principal provider of high performance computing services to Office of Science programs -- Magnetic Fusion Energy, High Energy and Nuclear Physics, Basic Energy Sciences, Biological and Environmental Research, and Advanced Scientific Computing Research.

[10] **The Globus Project**. http://www.globus.org

The Globus project is developing fundamental technologies needed to build computational grids. Grids are persistent environments that enable software applications to integrate instruments, displays, computational and information resources that are managed by diverse organizations in widespread locations.

[11] **The Condor Project**. http://www.cs.wisc.edu/condor/

The goal of the Condor Project is to develop, implement, deploy, and evaluate mechanisms and policies that support High Throughput Computing (HTC) on large collections of distributively owned computing resources. Guided by both the technological and sociological challenges of such a computing environment, the Condor Team has been building software tools that enable scientists and engineers to increase their computing throughput.

[12] **The Storage Resource Broker**. http://www.npaci.edu/DICE/SRB/

The SDSC Storage Resource Broker (SRB) is a client-server middleware that provides a uniform interface for connecting to heterogeneous data resources over a network and accessing replicated data sets. SRB, in conjunction with the Metadata Catalog (MCAT), provides a way to access data sets and resources based on their attributes rather than their names or physical locations.

[13] **The Portable Batch Scheduler**. http://www.pbspro.com/tech_overview.html

The purpose of the PBS system is to provide additional controls over initiating or scheduling execution of batch jobs; and to allow routing of those jobs between different hosts [that run administratively coupled instances of PBS]. The batch system allows a site to define and implement policy as to what types of resources and how much of each resource can be used by different jobs. The batch system also provides a mechanism with which a user can insure a job will have access to the resources required to complete.

[14] **PKI Service - An ESnet White Paper**, T. J. Genovese. September 15, 2000, DOE Energy Sciences Network. http://envisage.es.net/Docs/old%20docs/WhitePaper-PKI.pdf

This white paper will explore PKI technology of the ESnet community. The need in our community has been growing and expectations have been varied. With the deployment of large DOE computational grids and the development of the Federal PKI Policy Authority's (FPKIPA) Federal Bridge Certificate Authority's (FBCA) CP and CPS, the importance for ESnet to deploy a PKI infrastructure has also grown.

[15] **ESnet & DOE Science Grid PKI - Overview**. January 24, 2002. http://envisage.es.net/Docs/PKIwhitepaper.pdf

ESnet is building a Public Key Infrastructure service to support the DOE Science Grid mission and other SciDAC projects. DOE scientist and engineers will be able to use the ESnet PKI service to participate in the growing national and international computational Grids. To build this

service and to insure the widest possible acceptance of its certificates, we will be participating in two international forums. First, the Global Grid Forum, which is working to establish international Grid standards/recommendations - specifically we are contributing to the Grid Certificate policy working group and the Grid Information Services Area. The Grid CP effort is focusing on development of a common Certificate Policy that all Grid PKI's could use instead of custom individual CPs that hamper certificate validation. Second, we will be working with the European Data Grid CA operations group to insure that the EDG Test beds will accept our certificates. The project website, Envisage.es.net will be used to track the progress of this project. The website will contain all project documents and status reports. This paper will provide a project overview of the immediate requirements for the DOE Science Grid PKI support and cover the long-term project goals described in the ESnet PKI and Directory project document.

[16] **ESnet's SciDAC PKI & Directory Project - Homepage**, T. Genovese and M. Helm. DOE Energy Sciences Network. http://envisage.es.net/

This is the ESnet PKI project site. ESnet is building a Public Key Infrastructure service to support the DOE Science Grid, SciDAC projects and other DOE research efforts. The main goal is to provide DOE scientist and engineers Identity and Service certificates that allow them to participate in the growing national and international computational Grids.

[17] **Application Experiences with the Globus Toolkit**, S. Brunett, K. Czajkowski, S. Fitzgerald, I. Foster, A. Johnson, C. Kesselman, J. Leigh and S. Tuecke. In *Proc. 7th IEEE Symp. on High Performance Distributed Computing*. 1998: IEEE Press.http://www.globus.org/research/papers.html#globus-apps

"..., SF-Express, is a distributed interactive simulation (DIS) application that harnesses multiple supercomputers to meet the computational demands of large-scale network-based simulation environments. A large simulation may involve many tens of thousands of entities and requires thousands of processors. Globus services can be used to locate, assemble, and manage those resources. For example, in one experiment in March 1998, SF-Express was run on 1352 processors distributed over 13 supercomputers at nine sites ...This experiment involved over 100,000 entities, setting a new world record for simulation and meeting a performance goal that was not expected to be achieved until 2002."

[18] **Legion**. http://legion.virginia.edu/

Legion, an object-based metasystems software project at the University of Virginia, is designed for a system of millions of hosts and trillions of objects tied together with high-speed links. Users working on their home machines see the illusion of a single computer, with access to all kinds of data and physical resources, such as digital libraries, physical simulations, cameras, linear accelerators, and video streams. Groups of users can construct shared virtual work spaces, to collaborate research and exchange information. This abstraction springs from Legion's transparent scheduling, data management, fault tolerance, site autonomy, and a wide range of security options.

Legion sits on top of the user's operating system, acting as liaison between its own host(s) and whatever other resources are required. The user isn't bogged down with time-consuming negotiations with outside systems and system administrators, since Legion's scheduling and security policies act on his or her behalf. Conversely, it can protect its own resources against other Legion users, so that administrators can choose appropriate policies for who uses which resources under what circumstances. To allow users to take advantage of a wide range of

possible resources, Legion offers a user-controlled naming system called context space, so that users can easily create and use objects in farflung systems. Users can also run applications written in multiple languages, since Legion supports interoperability between objects written in multiple languages.

[19] **UNICORE**. http://www.unicore.de/

UNICORE lets the user prepare or modify structured jobs through a graphical user interface on a local Unix workstation or a Windows PC. Jobs can be submitted to any of the platforms of a UNICORE GRID and the user can monitor and control the submitted jobs through the job monitor part of the client.

A UNICORE job contains a number of interdependent tasks. The dependencies indicate temporal relations or data transfer. Currently, execution of scripts, compile, link, execute tasks and data transfer directives are supported. An execution system request associated with a job specifies where its tasks are to be run. Tasks can be grouped into sub-jobs, creating a hierarchical job structure and allowing different steps to execute on different systems within the UNICORE GRID.

[20] **Grid Monitoring Architecture Working Group**, Global Grid Forum. http://www-didc.lbl.gov/GGF-PERF/GMA-WG/

The Grid Monitoring Architecture working group is focused on producing a high-level architecture statement of the components and interfaces needed to promote interoperability between heterogeneous monitoring systems on the Grid. The main products of this work are the architecture document itself, and accompanying case studies that illustrate the concrete application of the architecture to monitoring problems.

[21] ***The Grid: Blueprint for a New Computing Infrastructure***, I. Foster and C. Kesselman, eds. 1998, Morgan Kaufmann. http://www.mkp.com/books_catalog/1-55860-475-8.asp

[22] **The Anatomy of the Grid: Enabling Scalable Virtual Organizations**, I. Foster, C. Kesselman and S. Tuecke. International J. Supercomputer Applications, 2001. **15**(3). http://www.globus.org/research/papers.html#anatomy

Defines Grid computing and the associated research field, proposes a Grid architecture, and discusses the relationships between Grid technologies and other contemporary technologies.

[23] **GriPhyN (Grid Physics Network)**. http://www.griphyn.org

The GriPhyN collaboration is a team of experimental physicists and information technology (IT) researchers who are implementing the first Petabyte-scale computational environments for data intensive science in the 21st century. Driving the project are unprecedented requirements for geographically dispersed extraction of complex scientific information from very large collections of measured data. To meet these requirements, which arise initially from the four physics experiments involved in this project but will also be fundamental to science and commerce in the 21st century, GriPhyN will deploy computational environments called Petascale Virtual Data Grids (PVDGs) that meet the data-intensive computational needs of a diverse community of thousands of scientists spread across the globe.

GriPhyN involves technology development and experimental deployment in four science projects. The CMS and ATLAS experiments at the Large Hadron Collider will search for the origins of mass and probe matter at the smallest length scales; LIGO (Laser Interferometer Gravitational-wave Observatory) will detect the gravitational waves of pulsars, supernovae and

in-spiraling binary stars; and SDSS (Sloan Digital Sky Survey) will carry out an automated sky survey enabling systematic studies of stars, galaxies, nebulae, and large-scale structure.

[24] **Virtual Observatories of the Future**, Caltech. http://www.astro.caltech.edu/nvoconf/

Within the United States, there is now a major, community-driven push towards the National Virtual Observatory (NVO). The NVO will federate the existing and forthcoming digital sky archives, both ground-based and space based.

[25] **Grid Information Services / MDS**, Globus Project. http://www.globus.org/mds/

Grid computing technologies enable wide-spread sharing and coordinated use of networked resources. Sharing relationships may be static and long-lived—e.g., among the major resource centers of a company or university—or highly dynamic: e.g., among the evolving membership of a scientific collaboration. In either case, the fact that users typically have little or no knowledge of the resources contributed by participants in the "virtual organization" (VO) poses a significant obstacle to their use. For this reason, information services designed to support the initial discovery and ongoing monitoring of the existence and characteristics of resources, services, computations, and other entities are a vital part of a Grid system. ("Grid Information Services for Distributed Resource Sharing" - http://www.globus.org/research/papers/MDS-HPDC.pdf)

The Monitoring and Discovery Service architecture addresses the unique requirements of Grid environments. Its architecture consists of two basic elements:

- A large, distributed collection of generic information providers provide access to information about individual entities, via local operations or gateways to other information sources (e.g., SNMP queries). Information is structured in term of a standard data model, taken from LDAP: an entity is described by a set of "objects" comprised of typed attribute-value pairs.

- Higher-level services, collect, manage, index, and/or respond to information provided by one or more information providers. We distinguish in particular aggregate directory services, which facilitate resource discovery and monitoring for VOs by implementing both generic and specialized views and search methods for a collection of resources. Other higher-level services can use this information and/or information obtained directly from providers for the purposes of brokering, monitoring, troubleshooting, etc.

Interactions between higher-level services (or users) and providers are defined in terms of two basic protocols: a soft-state registration protocol for identifying entities participating in the information service, and an enquiry protocol for retrieval of information about those entities, whether via query or subscription. In brief, a provider uses the registration protocol to notify higher-level services of its existence; a higher-level service uses the enquiry protocol to obtain information about the entities known to a provider, which it merges into its aggregate view. Integration with the Grid Security Infrastructure (GSI) provides for authentication and access control to information.

[26] **Grid Security Infrastructure (GSI)**, Globus Project. 2002. http://www.globus.org/security/

The primary elements of the GSI are identity certificates, mutual authentication, confidential communication, delegation, and single sign-on.

GSI is based on public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) communication protocol. Extensions to these standards have been added for single sign-on and delegation. The Globus Toolkit's implementation of the GSI adheres to the Generic Security

Service API (GSS-API), which is a standard API for security systems promoted by the Internet Engineering Task Force (IETF).

[27] **Globus Resource Allocation Manager (GRAM)**, Globus Project. 2002. http://www-fp.globus.org/gram/overview.html

The Globus Resource Allocation Manager (GRAM) is the lowest level of Globus resource management architecture. GRAM allows you to run jobs remotely, providing an API for submitting, monitoring, and terminating your job.

To run a job remotely, a GRAM gatekeeper (server) must be running on a remote computer, listening at a port; and the application needs to be compiled on that remote machine. The execution begins when a GRAM user application runs on the local machine, sending a job request to the remote computer.

The request is sent to the gatekeeper of the remote computer. The gatekeeper handles the request and creates a job manager for the job. The job manager starts and monitors the remote program, communicating state changes back to the user on the local machine. When the remote application terminates, normally or by failing, the job manager terminates as well.

The executable, stdin and stdout, as well as the name and port of the remote computer, are specified as part of the job request. The job request is handled by the gatekeeper, which creates a job manager for the new job. The job manager handles the execution of the job, as well as any communication with the user.

[28] **The GridFTP Protocol and Software**, Globus Project. 2002. http://www.globus.org/datagrid/gridftp.html

GridFTP is a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks. The GridFTP protocol is based on FTP, the highly-popular Internet file transfer protocol. We have selected a set of protocol features and extensions defined already in IETF RFCs and added a few additional features to meet requirement from current data grid projects.

[29] **A Grid Monitoring Architecture**, B. Tierney, R. Aydt, D. Gunter, W. Smith, V. Taylor, R. Wolski and M. Swany. http://www-didc.lbl.gov/GGF-PERF/GMA-WG/

The current GMA specification from the GGF Performance Working Group may be found in the documents section of the Working Group Web page.

[30] **Distributed Monitoring Framework (DMF)**, Lawrence Berkeley National Lab. http://www-didc.lbl.gov/DMF/

The goal of the Distributed Monitoring Framework is to improve end-to-end data throughput for data intensive applications in a high-speed WAN environments, and to provide the ability to do performance analysis and fault detection in a Grid computing environment. This monitoring framework will provide accurate, detailed, and adaptive monitoring of all of distributed computing components, including the network. Analysis tools will be able to use this monitoring data for real-time analysis, anomaly identification, and response.

Many of the components of the DMF have already been prototyped or implemented by the DIDC Group. The NetLogger Toolkit includes application sensors, some system and network sensors, a powerful event visualization tool, and a simple event archive. The Network characterization Service has proven to be a very useful hop-by-hop network sensor. Our work on the Global Grid

Forum Grid Monitoring Architecture (GMA) addressed the event management system. JAMM (Java Agents for Monitoring Management) is preliminary work on sensor management. The Enable project produced a simple network tuning advice service.

[31] **Information and Monitoring Services Architecture**, European Union DataGrid - WP3. http://hepunx.rl.ac.uk/edg/wp3/documentation/doc/arch/index.html

The aim of this work package is to specify, develop, integrate and test tools and infrastructure to enable end-user and administrator access to status and error information in a Grid environment and to provide an environment in which application monitoring can be carried out. This will permit both job performance optimisation as well as allowing for problem tracing and is crucial to facilitating high performance Grid computing.

[32] **Globus I/O**, Globus Project. 2002. http://www-unix.globus.org/api/c-globus-2.0-beta1/globus_io/html/index.html

The globus_io library is motivated by the desire to provide a uniform I/O interface to stream and datagram style communications. The goals in doing this are: 1) To provide a robust way to describe, apply, and query connection properties. These include the standard socket options (socket buffer sizes, etc), as well as additional attributes. These include security attributes and, eventually, QoS attributes. 2) support nonblocking I/O and handle asynchronous file and network events. 3) Provide a simple and portable way to implement communication protocols. Globus components such as GASS and GRAM can use this to redefine their control message protocol in terms of TCP messages.

[33] **Maui Silver Metascheduler**. http://www.supercluster.org/documentation/silver/silveroverview.html

Silver is an advance reservation metascheduler. Its design allows it to load balance workload across multiple systems in completely independent administrative domains. How much or how little a system participates in this load sharing activity is completely up to the local administration. All workload is tracked and accounted for allowing 'allocation' exchanges to take place between the active sites.

[34] **Personal Condor and Globus Glide-In**, Condor Project. http://www.cs.wisc.edu/condor/condorg/README

A Personal Condor is a version of Condor running as a regular user, without any special privileges. The idea is that you can use your Personal Condor to run jobs on your local workstations and have Condor keep track of their progress, and then through "flocking" access the resources of other Condor pools. Additionally, you can "Glide-in" to Globus Managed resources, and create virtual-condor pool by running the Condor daemons on the globus resoures, and then letting your Personal Condor manage those resources.

[35] **Condor-G**, J. Frey, T. Tannenbaum, M. Livny, I. Foster and S. Tuecke. In *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*. 2001: IEEE Press.http://www.globus.org/research/papers.html#Condor-G-HPDC

In recent years, there has been a dramatic increase in the amount of available computing and storage resources. Yet few have been able to exploit these resources in an aggregated form. We present the Condor-G system, which leverages software from Globus and Condor to allow users to harness multi-domain resources as if they all belong to one personal domain. We describe the

structure of Condor-G and how it handles job management, resource selection, security, and fault tolerance.

[36] **European Union DataGrid Project**. http://eu-datagrid.web.cern.ch/eu-datagrid/

The DataGrid Project is a proposal made to the European Commission for shared cost research and technological development funding. The project has six main partners: CERN - The European Organization for Nuclear Research near Geneva, Swiss; CNRS - France - Le Comité National de la Recherche Scientifique; ESRIN - the European Space Agency's Centre in Frascati (near Rome), Italy; INFN - Italy - Istituto Nazionale di Fisica Nucleare; NIKHEF - The Dutch National Institute for Nuclear Physics and High Energy Physics, Amsterdam, and; PPARC - United Kingdom - Particle Physics and Astronomy Research Council.

The objective of the project is to enable next generation scientific exploration which requires intensive computation and analysis of shared large-scale databases, from hundreds of TeraBytes to PetaBytes, across widely distributed scientific communities. We see these requirements emerging in many scientific disciplines, including physics, biology, and earth sciences. Such sharing is made complicated by the distributed nature of the resources to be used, the distributed nature of the communities, the size of the databases and the limited network bandwidth available. To address these problems we propose to build on emerging computational Grid technologies, such as that developed by the Globus Project

[37] **The GridPort Toolkit: a System for Building Grid Portals**, M. Thomas, S. Mock, M. Dahan, K. Mueller, D. Sutton and J. R. Boisseau.
http://www.tacc.utexas.edu/~mthomas/pubs/GridPort_HPDC11.pdf

Grid portals are emerging as convenient mechanisms for providing the scientific community with familiar and simplified interfaces to the Grid. Our experience in implementing Grid portals has led to the creation of GridPort: a unique, layered software system for building Grid Portals. This system has several unique features: the software is portable and runs on most webservers; written in Perl/CGI, it is easy to support and modify; it is flexible and adaptable; it supports single login between multiple portals; and portals built with it may run across multiple sites and organizations. The feasibility of this portal system has been successfully demonstrated with the implementation of several application portals. In this paper we describe our experiences in building this system, including philosophy and design choices. We explain the toolkits we are building, and we demonstrate the benefits of this system with examples of several production portals. Finally, we discuss our experiences with Grid web service architectures.

[38] **Cactus**. http://www.cactuscode.org/

Cactus is an open source problem solving environment designed for scientists and engineers. Its modular structure easily enables parallel computation across different architectures and collaborative code development between different groups. Cactus originated in the academic research community, where it was developed and used over many years by a large international collaboration of physicists and computational scientists.

The name Cactus comes from the design of a central core (or "flesh") which connects to application modules (or "thorns") through an extensible interface. Thorns can implement custom developed scientific or engineering applications, such as computational fluid dynamics. Other thorns from a standard computational toolkit provide a range of computational capabilities, such as parallel I/O, data distribution, or checkpointing.

Cactus runs on many architectures. Applications, developed on standard workstations or laptops, can be seamlessly run on clusters or supercomputers. Cactus provides easy access to many cutting edge software technologies being developed in the academic research community, including the Globus Metacomputing Toolkit, HDF5 parallel file I/O, the PETSc scientific library, adaptive mesh refinement, web interfaces, and advanced visualization tools.

[39] **Supporting Efficient Execution in Heterogeneous Distributed Computing Environments with Cactus and Globus**, G. Allen, T. Dramlitsch, I. Foster, N. Karonis, M. Ripeanu, E. Seidel and B. Toonen. In *SC 2001*. 2001.http://www.globus.org/research/papers.html#sc01ewa

Members of the Cactus and Globus projects have won one of this year's Gordon Bell Prizes in high-performance computing for the work described in their paper: Supporting Efficient Execution in Heterogeneous Distributed Computing Environments with Cactus and Globus . The international team comprised of Thomas Dramlitsch, Gabrielle Allen and Ed Seidel, from the Max Planck Institute for Gravitational Physics, along with colleagues Matei Ripeanu, Ian Foster, Brian Toonen from the University of Chicago and Argonne National Laboratory, and Nicholas Karonis from Northern Illinois University. The special category award was presented during SC2001, a yearly conference showcasing high-performance computing and networking, this year held in Denver, Colorado.

The prize was awarded for the group's work on concurrently harnessing the power of multiple supercomputers to solve Grand Challenge problems in physics which require substantially more resources than can be provided by a single machine. The group enhanced the communication layer of Cactus, a generic programming framework designed for physicists and engineers, adding techniques capable of dynamically adapting the code to the available network bandwidth and latency between machines. The message passing layer itself used MPICH-G2, a grid-enabled implementation of the MPI protocol which handles communications between machines separated by a wide area network. In addition, the Globus Toolkit was used to provide authentification and staging of simulations across multiple machines.

From "Cactus, Globus and MPICH-G2 Win Top Supercomputing Award" at http://www.cactuscode.org/News/GordonBell2001.html

[40] **A Jini-based Computing Portal System**, T. Suzumura, S. Matsuoka and H. Nakada. In *Proceeding of SC2001*. 2001.http://ninf.apgrid.org/papers/sc01suzumura/sc2001.pdf

JiPANG(A Jini-based Portal Augmenting Grids) is a portal system and a toolkit which provides uniform access interface layer to a variety of Grid systems, and is built on top of Jini distributed object technology. JiPANG performs uniform higher-level management of the computing services and resources being managed by individual Grid systems such as Ninf, NetSolve, Globus, etc. In order to give the user a uniform interface to the Grids JiPANG provides a set of simple Java APIs called the JiPANG Toolkits, and furthermore, allows the user to interact with Grid systems, again in a uniform way, using the JiPANG Browser application. With JiPANG, users need not install any client packages beforehand to interact with Grid systems, nor be concerned about updating to the latest version. Such uniform, transparent services available in a ubiquitous manner we believe is essential for the success of Grid as a viable computing platform for the next generation.

[41] **GridRPC Tutorial**, H. Nakada, S. Matsuoka, M. Sato and S. Sekiguchi. http://ninf.apgrid.org/papers/gridrpc_tutorial/gridrpc_tutorial_e.html

GridRPC is a middleware that provides remote library access and task-parallel programming model on the Grid. Representative systems include Ninf, Netsolve, etc. We employ Ninf to exemplify how to program Grid applications using Ninf, in particular, how to "Gridify" a numerical library for remote RPC execution, how to perform parallel parameter sweep survey using multiple servers on the Grid.

[42] **NetSolve**. http://icl.cs.utk.edu/netsolve/

NetSolve is a client-server system that enables users to solve complex scientific problems remotely. The system allows users to access both hardware and software computational resources distributed across a network. NetSolve searches for computational resources on a network, chooses the best one available, and using retry for fault-tolerance solves a problem, and returns the answers to the user. A load-balancing policy is used by the NetSolve system to ensure good performance by enabling the system to use the computational resources available as efficiently as possible. Our framework is based on the premise that distributed computations involve resources, processes, data, and users, and that secure yet flexible mechanisms for cooperation and communication between these entities is the key to metacomputing infrastructures.

[43] **ILab: An Advanced User Interface Approach for Complex Parameter Study Process Specification on the Information Power Grid**, M. Yarrow, K. M. McCann, R. Biswas and R. F. V. d. Wijngaart. In *Grid 2000: First IEEE/ACM International Workshop*. 2000. Bangalore, India.http://www.ipg.nasa.gov/research/papers/nas-00-009.pdf

The creation of parameter study suites has recently become a more challenging problem as the parameter studies have become multi-tiered and the computational environment has become a supercomputer grid. The parameter spaces are vast, the individual problem sizes are getting larger, and researchers are seeking to combine several successive stages of parameterization and computation. Simultaneously, grid-based computing offers immense resource opportunities but at the expense of great difficulty of use. We present ILab, an advanced graphical user interface approach to this problem. Our novel strategy stresses intuitive visual design tools for parameter study creation and complex process specification, and also offers programming-free access to grid-based supercomputer resources and process automation.

[44] **US-CMS Testbed Production - joint news update with GriPhyN/iVDGL**, Particle Physics Data Grid. 2002. http://www.ppdg.net/ppdg_news.htm

Members of the CMS experiment working in concert with PPDG, iVDGL, and GriPhyN have carried out the first production-quality simulated data generation on a data grid comprising sites at Caltech, Fermilab, the University of California-San Diego, the University of Florida, and the University of Wisconsin-Madison. This is a combination of efforts supported by DOE SciDAC, HENP, MICS, and the NSF as well as the EU funded EU-DataGrid project.

The deployed data grid serves as an integration framework where grid middleware components are brought together to form the basis for distributed CMS Monte Carlo Production (CMS-MOP) and used to produce data for the global CMS physics program. The middleware components include Condor-G, DAGMAN, GDMP, and the Globus Toolkit packaged together in the first release of the Virtual Data Toolkit.

[45] **Condor-G**, Condor Project. http://www.cs.wisc.edu/condor/condorg/

Condor-G provides the grid computing community with a powerful, full-featured task broker. Used as a front-end to a computational grid, Condor-G can manage thousands of jobs destined to

run at distributed sites. It provides job monitoring, logging, notification, policy enforcement, fault tolerance, credential management, and it can handle complex job-interdependencies. Condor-G's flexible and intuitive commands are appropriate for use directly by end-users, or for interfacing with higher-level task brokers and web portals.

[46] **Peer-to-Peer Area**, Global Grid Forum. http://www.gridforum.org/4_GP/P2P.htm

This is a very new GGF activity, and initially the GGF Peer-to-Peer Area consists of the Working Groups of the previous Peer-to-Peer Working Group organization, which has merged with GGF. These WGs are:

- NAT/Firewall
- Taxonomy
- Peer-to-Peer Security
- File Services
- Trusted Library

[47] **MPICH-G2**. http://www.hpclab.niu.edu/mpi/

MPICH-G2 is a grid-enabled implementation of the MPI v1.1 standard. That is, using Globus services (e.g., job startup, security), MPICH-G2 allows you to couple multiple machines, potentially of different architectures, to run MPI applications. MPICH-G2 automatically converts data in messages sent between machines of different architectures and supports multiprotocol communication by automatically selecting TCP for intermachine messaging and (where available) vendor-supplied MPI for intramachine messaging.

[48] **PVM**. http://www.csm.ornl.gov/pvm/pvm_home.html

PVM (Parallel Virtual Machine) is a software package that permits a heterogeneous collection of Unix and/or Windows computers hooked together by a network to be used as a single large parallel computer. Thus large computational problems can be solved more cost effectively by using the aggregate power and memory of many computers. The software is very portable. The source, which is available free thru netlib, has been compiled on everything from laptops to CRAYs.

[49] **PACX-MPI: Extending MPI for Distributed Computing**, E. Gabriel, M. Mueller and M. Resch. High Performance Computing Center in Stuttgart.
http://www.hlrs.de/organization/pds/projects/pacx-mpi/

Simulation using several MPPs requires a communication interface wich enables both efficient message-passing between the nodes inside each MPP and between the machines itself. At the same time the data exchange should rely on a standard interface.

PACX-MPI (PArallel Computer eXtension) was initially developed to connect a Cray-YMP to an Intel Paragon. Currently it has been extended to couple two and more MPPs to form a cluster of high-performance computers for Metacomputing.

[50] **Trans-Atlantic Metacomputing**. http://www.hoise.com/articles/AE-PR-11-97-7.html

Stuttgart, 15 November 97: An international team of computer experts combined the capacity of machines at three large supercomputer centers that exceeded three Tflop/s. The meta-computing effort used for the simulations linked 3 of the top 10 largest supercomputers in the world. Involved were HLRS, the High-Performance Computing-Center at the University of Stuttgart, Germany, Sandia National Laboratories, SNL, Albuquerque, NM, Pittsburgh Supercomputing Center, Pittsburgh, PA. They demonstrated a trans-Atlantic meta-computing and meta-

visualization environment. The demonstration is a component of this official G7 Information Society pilot programme.

[51] **Condor User's Manual**, Condor Project. http://www.cs.wisc.edu/condor

The goal of the Condor Project is to develop, implement, deploy, and evaluate mechanisms and policies that support High Throughput Computing (HTC) on large collections of distributively owned computing resources. Guided by both the technological and sociological challenges of such a computing environment, the Condor Team has been building software tools that enable scientists and engineers to increase their computing throughput.

[52] **Particle Physics Data Grid (PPDG)**. http://www.ppdg.net/

The Particle Physics Data Grid collaboration was formed in 1999 because its members were keenly aware of the need for Data Grid services to enable the worldwide distributed computing model of current and future high-energy and nuclear physics experiments. Initially funded from the NGI initiative and later from the DOE MICS and HENP programs, it has provided an opportunity for early development of the Data Grid architecture as well as evaluating some prototype Grid middleware.

PPDG involves work with and by four major high energy physcics experiments who are developing and testing data Grid technology.

[53] **Data Mining on NASA's Information Power Grid**, T. Hinke and J. Novonty. In *Ninth IEEE International Symposium on High Performance Distributed Computing*. 2000.http://www.ipg.nasa.gov/engineering/presentations/PDF_presentations/21-Hinke.pdf

This paper describes the development of a data mining system that is to operate on NASA's Information Power Grid (IPG). Mining agents will be staged to one or more processors on the IPG. There they will grow using just-intime acquision of new operations. They will mine data delivered using just-in-time delivery. Some initial experimental results are presented.

[54] **Overview of the Grid Security Infrastructure (GSI)**, Globus Project. 2002. http://www-fp.globus.org/security/overview.html

The GSI uses public key cryptography (also known as asymetric cryptography) as the basis for its functionality. Many of the terms and concepts used in this description of the GSI come from its use of public key cryptography. The PKI context is described here.

[55] **Global Access to Secondary Storage (GASS)**, Globus Project. 2002. http://www-fp.globus.org/gass/

GASS provides a Unix I/O style access to remote files. Operations supported include remote read, remote write and append (achieved by copying the entire file to the local system on file open, and back on close). GASS also provide for local caching of file so that they may be reused during a job without re-copying.

A typical use of GASS is to put a GASS server on or near a tertiary storage system that can access files by filename. This allows remote, file-like access by replicating the file locally. A second typical use is to locate a GASS server on a user system where files (such as simulation input files) are managed so that Grid jobs can read data from those systems.

[56] **A Replica Management Service for High-Performance Data Grids**, Globus Project. 2002. http://www-fp.globus.org/datagrid/replica-management.html

Replica management is an important issue for a number of scientific applications. Consider a data set that contains one petabyte (one thousand million megabytes) of experimental results for a particle physics application. While the complete data set may exist in one or possibly several physical locations, it is likely that few universities, research laboratories or individual researchers will have sufficient storage to hold a complete copy. Instead, they will store copies of the most relevant portions of the data set on local storage for faster access. Replica Management is the process of keeping track of where portions of the data set can be found.

[57] **The Globus Replica Catalog**, Globus Project. 2002. http://www-fp.globus.org/datagrid/replica-catalog.html

The Globus Replica Catalog supports replica management by providing mappings between logical names for files and one or more copies of the files on physical storage systems.

[58] **Globus Replica Management**, Globus Project. 2002. http://www-fp.globus.org/datagrid/replica-management.html

Replica management is an important issue for a number of scientific applications. Consider a data set that contains one petabyte (one thousand million megabytes) of experimental results for a particle physics application. While the complete data set may exist in one or possibly several physical locations, it is likely that few universities, research laboratories or individual researchers will have sufficient storage to hold a complete copy. Instead, they will store copies of the most relevant portions of the data set on local storage for faster access. Replica Management is the process of keeping track of where portions of the data set can be found.

Globus Replica Management integrates the Globus Replica Catalog (for keeping track of replicated files) and GridFTP (for moving data) and provides replica management capabilities for data grids.

[59] **Grid Data Management Pilot (GDMP): A Tool for Wide Area Replication**, A. Samar and H. Stockinger. In *IASTED International Conference on Applied Informatics (AI2001), Innsbruck, Austria, February 2001.* 2001.http://web.datagrid.cnr.it/pls/portal30/GRID.RPT_DATAGRID_PAPERS.show

The stringent requirements of data consistency, security and high-speed transfer of huge amounts of data, imposed by the physics community need to be satisfed by an asynchronous replication mechanism. A pilot project called the Grid Data Management Pilot (GDMP) has been initiated which is responsible for asynchronously replicating large object-oriented data stores over the wide-area network to globally distributed sites.

The GDMP software consists of several modules that closely work together but are easily replaceable. In this section we describe the modules and the software architecture of GDMP. The core modules are Control Communication, Request Manager, Security, Database Manager and the Data Mover. An application which is visible as a command-line tool uses one or several of these modules.

[60] **Distributed-Parallel Storage System (DPSS)**. http://www-didc.lbl.gov/DPSS/

The DPSS is a data block server, which provides high-performance data handling and architecture for building high-performance storage systems from low-cost commodity hardware components. This technology has been quite successful in providing an economical, high-performance, widely distributed, and highly scalable architecture for caching large amounts of data that can potentially be used by many different users.

Current performance results are 980 Mbps across a LAN and 570 Mbps across a WAN.

[61] **Using GSI Enabled SSH**, NCSA.
http://www.ncsa.uiuc.edu/UserInfo/Alliance/GridSecurity/GSI/Tools/GSSH.html

SSH is a well-known program for doing secure logon to remote hosts over an open network. GSI enabled SSH (GSSH) is a modification of SSH version 1.2.27 that allows SSH to use Alliance certificates and designated proxy certificates for authentication.

[62] **GSI-Enabled OpenSSH**, NCSA.
http://www.ncsa.uiuc.edu/Divisions/ACES/GSI/openssh/

NCSA maintains a patch to OpenSSH that adds support for GSI authentication.

[63] **Access Grid**. http://www-fp.mcs.anl.gov/fl/accessgrid/default.htm

The Access Grid (AG) is the ensemble of resources that can be used to support human interaction across the grid. It consists of multimedia display, presentation and interactions environments, interfaces to grid middleware, interfaces to visualization environments. The Access Grid will support large-scale distributed meetings, collaborative work sessions, seminars, lectures, tutorials and training. The Access Grid design point is group to group communication (thus differentiating it from desktop to desktop based tools that focus on individual communication). The Access Grid environment must enable both formal and informal group interactions. Large-format displays integrated with intelligent or active meeting rooms are a central feature of the Access Grid nodes. Access Grid nodes are "designed spaces" that explicitly contain the high-end audio and visual technology needed to provide a high-quality compelling user experience.

The Access Grid complements the computational grid, indeed the Access Grid node concept is specifically targeted to provide "group" access to the Grid. This access maybe for remote visualization or interactive applications, or for utilizing the high-bandwidth environment for virtual meetings and events.

Access Grid Nodes (global AG sites) provide a research environment for the development of distributed data and visualization corridors and for studying issues relating to collaborative work in distributed environments.

[64] **Grid Information Services**, Global Grid Forum.
http://www.gridforum.org/1_GIS/gis.htm

The (GIS) Grid Information Services Area is concerned with services that either provide information or consume information pertaining to the Grid.

[65] **OpenSSL Certification Authority**. http://www.openssl.org/docs/apps/ca.html#

The *ca* command is a minimal CA application. It can be used to sign certificate requests in a variety of forms and generate CRLs it also maintains a text database of issued certificates and their status.

[66] **DOE Science Grid PKI Certificate Policy And Certification Practice Statement**.
http://www.doegrids.org/

This document represents the policy for the DOE Science Grid Certification Authority operated by ESnet. It addresses Certificate Policy (CP) and Certification Practice Statement (CPS). The CP is a named set of rules that indicates the applicability of a certificate to a particular

community and/or class of application with common security requirements. For example, a particular certificate policy might indicate applicability of a type of certificate to the authentication of electronic data interchange transactions for the trading of goods within a given price range. The CPS is a statement of the practices, which a certification authority employs in issuing certificates.

[67] **Certification Authorities Acceptance and Feature Matrices**, European Union DataGrid. 2002. http://www.cs.tcd.ie/coghlan/cps-matrix/

The Acceptance and Feature matrices are key aspects of establishing cross-site trust.

[68] **Certification Authorities**, European Union DataGrid. 2002. http://marianne.in2p3.fr/datagrid/ca/ca-table-ca.html

The current list of EU DataGrid recognized CAs and their certificates.

[69] **Netscape Certificate Management System**, Netscape. http://wp.netscape.com/cms

Use Netscape Certificate Management System, the highly scalable and flexible security solution, to issue and manage digital certificates for your extranet and e-commerce applications.

[70] **KX.509/KCA**, NSF Middleware Initiative. 2002. http://www.nsf-middleware.org/documentation/KX509KCA/

KX.509, from the University of Michigan, is a Kerberized client-side program that acquires an X.509 certificate using existing Kerberos tickets. The certificate and private key generated by KX.509 are normally stored in the same cache alongside the Kerberos credentials. This enables systems that already have a mechanism for removing unused Kerberos credentials to also automatically remove the X.509 credentials. There is then a PKCS11 library that can be loaded by Netscape and Internet Explorer access and use these credentials for https web activity.

The Globus Toolkit normally uses X.509 credentials. KX.509 allows a user to authenticate to a host that is running Globus software using Kerberos tickets instead of requiring X.509 certificates to be installed.

To use Globus utilities on a (local or remote) machine, a user is required to authenticate to the machine using appropriate X.509 certificates. These long-term certificates are used to create a short-term proxy, which is used for authentication and Globus utilities. The proxy will expire after a preset amount of time, after which a new one must be generated from the long-term X.509 certificates again.

KX.509 can be used in place of permanent, long-term certificates. It does this by creating X.509 credentials (certificate and private key) using your existing Kerberos ticket. These credentials are then used to generate the Globus proxy certificate.

[71] **ESnet**. http://www.es.net/

The Energy Sciences Network, or ESnet, is a high-speed network serving thousands of Department of Energy scientists and collaborators worldwide. A pioneer in providing high-bandwidth, reliable connections, ESnet enables researchers at national laboratories, universities and other institutions to communicate with each other using the collaborative capabilities needed to address some of the world's most important scientific challenges. Managed and operated by the ESnet staff at Lawrence Berkeley National Laboratory, ESnet provides direct connections to all major DOE sites with high performance speeds, as well as fast interconnections to more than 100 other networks. Funded principally by DOE's Office of Science, ESnet services allow

scientists to make effective use of unique DOE research facilities and computing resources, independent of time and geographic location. ESnet is funded by the DOE Office of Science to provide network and collaboration services in support of the agency's research missions.

[72] **Using Globus/GSI with a firewall**, Globus Project.
http://www.globus.org/Security/v1.1/firewalls.html

Describes the network traffic generated by Globus and GSI applications.

[73] **GSI-Enabled FTP**, Globus Project.
http://www.globus.org/security/v1.1/index.htm#gsiftp

GSIFTP is a standard Unix FTP program modified to use the GSI libraries for authentication. It replaces the normal password authentication with Globus certificate authentication.

[74] **Creating a Hierarchical GIIS**, Globus Project. http://www.globus.org/mds/

This document describes by example how to create a hierarchical GIIS for use by MDS 2.1. The following topics are covered in this document: Configuration files used in creating a hierarchical GIIS architecture; Renaming of distinguished names (DNs) in the GIIS hierarchy; Timing Issues and Registration Control; Additional information on site policy, configuration files, and command syntax.

[75] **Community Authorization Service (CAS)**, Globus Project. 2002.
http://www.globus.org/security/CAS/

CAS allows resource providers to specify course-grained access control policies in terms of communities as a whole, delegating fine-grained access control policy management to the community itself. Resource providers maintain ultimate authority over their resources but are spared day-to-day policy administration tasks (e.g. adding and deleting users, modifying user privileges). Briefly, the process is: 1) A CAS server is initiated for a community: a community representative acquires a GSI credential to represent that community as a whole, and then runs a CAS server using that community identity. 2) Resource providers grant privileges to the community. Each resource provider verifies that the holder of the community credential represents that community and that the community's policies are compatible with the resource provider's own policies. Once a trust relationship has been established, the resource provider then grants rights to the community identity, using normal local mechanisms (e.g. gridmap files and disk quotas, filesystem permissions, etc.). 3) Community representatives use the CAS to manage the community's trust relationships (e.g., to enroll users and resource providers into the community according to the community's standards) and grant fine-grained access control to resources. The CAS server is also used to manage its own access control policies; for example, community members who have the appropriate privileges may authorize additional community members to manage groups, grant permissions on some or all of the community's resources, etc. 4) When a user wants to access resources served by the CAS, that user makes a request to the CAS server. If the CAS server's database indicates that the user has the appropriate privileges, the CAS issues the user a GSI restricted proxy credential with an embedded policy giving the user the right to perform the requested actions. 5) The user then uses the credentials from the CAS to connect to the resource with any normal Globus tool (e.g. GridFTP). The resource then applies its local policy to determine the amount of access granted to the community, and further restricts that access based on the policy in the CAS credentials, This serves to limit the user's privileges to the intersection of those granted by the CAS to the user and those granted by the resource provider to the community.

[76] **Resource Manager for Globus-based Wide-area Cluster Computing**, Y. Tanaka, M. Sato, M. Hirano, H. Nakada and S. Sekiguchi. In *1st IEEE International Workshop on Cluster Computing (IWCC'99)*. 1999.http://ninf.apgrid.org/papers/iwcc99tanaka/IWCC99.pdf

[77] **Performance Evaluation of a Firewall-compliant Globus-based Wide-area Cluster System**, Y. Tanaka, M. Sato, M. Hirano, H. Nakada and S. Sekiguchi. In *9th IEEE International Symposium on High Performance Distributed Computing*. 2000.http://ninf.apgrid.org/papers/hpdc00tanaka/HPDC00.pdf

[78] **NetLogger: A Toolkit for Distributed System Performance Analysis**, D. Gunter, B. Tierney, B. Crowley, M. Holding and J. Lee. In *IEEE Mascots 2000: Eighth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. 2000.http://www-didc.lbl.gov/papers/NetLogger.Mascots.paper.ieee.pdf

Diagnosis and debugging of performance problems on complex distributed systems requires end-to-end performance information at both the application and system level. We describe a methodology, called NetLogger, that enables real-time diagnosis of performance problems in such systems. The methodology includes tools for generating precision event logs, an interface to a system eventmonitoring framework, and tools for visualizing the log data and real-time state of the distributed system. Low overhead is an important requirement for such tools, therefore we evaluate efficiency of the monitoring itself. The approach is novel in that it combines network, host, and application-level monitoring, providing a complete view of the entiresystem.

[79] **Pipechar Network Characterization Service**, G. Jin. http://www-didc.lbl.gov/NCS/

Tools based on hop-by-hop network analysis are increasingly critical to network troubleshooting on the rapidly growing Internet. Network characterization service (NCS) provides ability to diagnose and troubleshooot networks hop-by-hop in an easy and timely fashion. Using NCS makes applications capable to fully ulitilize the high-speed networks, e.g., saturating 1Gbps local network from a single x86 platform. This page contains rich information about NCS and network measurement algorithms. Tutorials for using NCS to analyze and troubleshoot network problems is presented below.

[80] **High-Speed Distributed Data Handling for On-Line Instrumentation Systems**, W. Johnston, W. Greiman, G. Hoo, J. Lee, B. Tierney, C. Tull and D. Olson. In *ACM/IEEE SC97: High Performance Networking and Computing*. 1997.http://www-itg.lbl.gov/~johnston/papers.html

[81] **The NetLogger Methodology for High Performance Distributed Systems Performance Analysis**, B. Tierney, W. Johnston, B. Crowley, G. Hoo, C. Brooks and D. Gunter. In *Proc. 7th IEEE Symp. on High Performance Distributed Computing*. 1998.http://www-didc.lbl.gov/NetLogger/

[82] **A Network-Aware Distributed Storage Cache for Data Intensive Environments**, B. Tierney, J. Lee, B. Crowley, M. Holding, J. Hylton and F. Drake. In *Proc. 8th IEEE Symp. on High Performance Distributed Computing*. 1999.http://www-didc.lbl.gov/papers/dpss.hpdc99.pdf

[83] **Using NetLogger for Distributed Systems Performance Analysis of the BaBar Data Analysis System**, B. Tierney, D. Gunter, J. Becla, B. Jacobsen and D. Quarrie. In *Proceedings of Computers in High Energy Physics 2000 (CHEP 2000)*. 2000.http://www-didc.lbl.gov/papers/chep.2K.Netlogger.pdf

[84] **Dynamic Monitoring of High-Performance Distributed Applications**, D. Gunter, B. Tierney, K. Jackson, J. Lee and M. Stoufer. In *11th IEEE Symposium on High Performance Distributed Computing, HPDC-11*. 2002.http://www-didc.lbl.gov/papers/HPDC02-HP-monitoring.pdf

[85] **Applied Techniques for High Bandwidth Data Transfers across Wide Area Networks**, J. Lee, D. Gunter, B. Tierney, W. Allock, J. Bester, J. Bresnahan and S. Tuecke. In *Proceedings of Computers in High Energy Physics 2001 (CHEP 2001)*. 2001. Beijing China.http://www-didc.lbl.gov/papers/dpss_and_gridftp.pdf

[86] **TCP tuning Guide for Distributed Applications on Wide Area Networks**, B. Tierney. In *Usenix ;login Journal*. 2001.http://www-didc.lbl.gov/papers/usenix-login.pdf

Also see **http://www-didc.lbl.gov/tcp-wan.html**

[87] **A TCP Tuning Daemon**, T. Dunigan, M. Mathis and B. Tierney. In *Proceeding of IEEE Supercomputing 2002*. 2002. Baltimore, Maryland, USA.http://www-didc.lbl.gov/publications.html

[88] **Globus Quick Start Guide**, Globus Project. 2002. http://www.globus.org/toolkit/documentation/QuickStart.pdf

This document is intended for people who use Globus-enabled applications. It includes information on how to set up one's environment to use the Globus Toolkit and applications based on Globus software.

[89] **NetSaint**. http://www.netsaint.org/

NetSaint is a program that will monitor hosts and services on your network. It has the ability to email or page you when a problem arises and when it gets resolved. ... It can run either as a normal process or as a daemon, intermittently running checks on various services that you specify. The actual service checks are performed by external "plugins" which return service information to NetSaint. Several CGI programs are included with NetSaint in order to allow you to view the current service status, history, etc. via a web browser.

[90] **Poznan Supercomputing and Networking Center**. http://www.man.poznan.pl/

[91] **Virtual User Account System**, N. Meyer and P. Wolniewicz. http://www.man.poznan.pl/metacomputing/cluster/

This system allows the running of jobs into a distributed cluster (between supercomputing centres) without additional administration overhead. A user does not have to have accounts on all supercomputers in the cluster, he only submits a job into the LSF on a local machine. The job is calculated on a virtual user account, but all billings and results are generated for the real user.

[92] **The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration**, I. Foster, C. Kesselman, J. Nick and S. Tuecke. http://www.globus.org/research/papers.html#OGSA

[We] define .... how a Grid functions and how Grid technologies can be implemented and applied. .... we focus here on the nature of the services that respond to protocol messages. We view a Grid as an extensible set of Grid services that may be aggregated in various ways to meet the needs of VOs, which themselves can be defined in part by the services that they operate and

share. We then define the behaviors that such Grid services should possess in order to support distributed systems integration. By stressing functionality (i.e., "physiology"), this view of Grids complements the previous protocol-oriented ("anatomical") description.

Second, we explain how Grid technologies can be aligned with Web services technologies ... to capitalize on desirable Web services properties, such as service description and discovery; automatic generation of client and server code from service descriptions; binding of service descriptions to interoperable network protocols; compatibility with emerging higher-level open standards, services and tools; and broad commercial support. We call this alignment-and augmentation-of Grid and Web services technologies an Open Grid Services Architecture (OGSA), with the term architecture denoting here a well-defined set of basic interfaces from which can be constructed interesting systems, and open being used to communicate extensibility, vendor neutrality, and commitment to a community standardization process. This architecture uses the Web Services Description Language (WSDL) to achieve self-describing, discoverable services and interoperable protocols, with extensions to support multiple coordinated interfaces and change management. OGSA leverages experience gained with the Globus Toolkit to define conventions and WSDL interfaces for a Grid service, a (potentially transient) stateful service instance supporting reliable and secure invocation (when required), lifetime management, notification, policy management, credential management, and virtualization. OGSA also defines interfaces for the discovery of Grid service instances and for the creation of transient Grid service instances. The result is a standards-based distributed service system (we avoid the term distributed object system due to its overloaded meaning) that supports the creation of the sophisticated distributed services required in modern enterprise and interorganizational computing environments.

Third, we focus our discussion on commercial applications rather than the scientific and technical applications ..... We believe that the same principles and mechanisms apply in both environments. However, in commercial settings we need, in particular, seamless integration with existing resources and applications, and with tools for workload, resource, security, network QoS, and availability management. OGSA's support for the discovery of service properties facilitates the mapping or adaptation of higher-level Grid service functions to such native platform facilities. OGSA's service orientation also allows us to virtualize resources at multiple levels, so that the same abstractions and mechanisms can be used both within distributed Grids supporting collaboration across organizational domains and within hosting environments spanning multiple tiers within a single IT domain. A common infrastructure means that differences (e.g., relating to visibility and accessibility) derive from policy controls associated with resource ownership, privacy, and security, rather than interaction mechanisms. Hence, as today's enterprise systems are transformed from separate computing resource islands to integrated, multitiered distributed systems, service components can be integrated dynamically and flexibly, both within and across various organizational boundaries.

[93] **Open Grid Service Interface Working Group**, Global Grid Forum.
http://www.gridforum.org/ogsi-wg/

The purpose of the OGSI Working Group is to review and refine the Grid Service Specification and other documents that derive from this specification, including OGSA-infrastructure-related technical specifications and supporting informational documents.

[94] **Grid Computing Environments Research Group**, Global Grid Forum.
http://www.computingportals.org/

Our working group is aimed at contributing to the coherence and interoperability of frameworks, portals, Problem Solving Environments, and other Grid-based computing environments and Grid services. We do this by choosing "best practices" projects to derive standards, protocols, API's and SDK's that are required to integrate technology implementations and solutions.

[95] **Python Globus(pyGlobus)**, K. Jackson. Lawrence Berkeley National Laboratory. http://www-itg.lbl.gov/gtg/projects/pyGlobus/index.html

- Provide a clean object-oriented interface to the Globus toolkit.
- Provide similar performance to using the underlying C code as much as possible.
- Minimize the number of changes necessary when aspects of Globus change.
- Where possible, make Globus as natural to use from Python as possible.
- For example, the gassFile module allows the manipulation of remote GASS files as Python file objects.

[96] **CoG Kits: Enabling Middleware for Designing Science Appl**, K. Jackson and G. v. Laszewski. Submitted SciDAC proposal, March, 2001, Lawrence Berkeley National Laboratory and Argonne National Laboratory.

[97] **CoG Kits: A Bridge Between Commodity Distributed Computing and High-Performance Grids**, G. v. Laszewski, I. Foster and J. Gawor. In *ACM 2000 Java Grande Conference*. 2000. San Francisco.http://www.globus.org/cog/documentation/papers/index.html

[98] **A Java Commodity Grid Kit**, G. v. Laszewski, I. Foster, J. Gawor and P. Lane. Concurrency: Experience and Practice, 2001. http://www.globus.org/cog/documentation/papers/index.html

[99] **The Grid Portal Development Kit**, J. Novotny. Concurrency - Practice and Experience, 2000. http://www.doesciencegrid.org/Grid/projects/GPDK/gpdkpaper.pdf

[100] **NPSS on NASA's IPG: Using CORBA and Globus to Coordinate Multidisciplinary Aeroscience Applications**, G. Lopez, J. Follen, R. Gutierrez, I. Foster, B. Ginsburg, O. Larsson, S. Martin, S. Tuecke and D. Woodford. 2000.http://www.ipg.nasa.gov/research/papers/NPSS_CAS_paper.html

[The] NASA Glenn Research Center is developing an environment for the analysis and design of aircraft engines called the Numerical Propulsion System Simulation (NPSS). The vision for NPSS is to create a "numerical test cell" enabling full engine simulations overnight on cost-effective computing platforms. To this end, NPSS integrates multiple disciplines such as aerodynamics, structures, and heat transfer and supports "numerical zooming" from 0-dimensional to 1-, 2-, and 3-dimensional component engine codes. To facilitate the timely and cost-effective capture of complex physical processes, NPSS uses object-oriented technologies such as C++ objects to encapsulate individual engine components and Common Object Request Broker Architecture (CORBA) Object Request Brokers (ORBs) for object communication and deployment across heterogeneous computing platforms.

IPG implements a range of Grid services such as resource discovery, scheduling, security, instrumentation, and data access, many of which are provided by the Globus toolkit. IPG facilities have the potential to benefit NPSS considerably. For example, NPSS should in principle be able to use Grid services to discover dynamically and then coschedule the resources required for a particular engine simulation, rather than relying on manual placement of ORBs as at present. Grid services can also be used to initiate simulation components on massively parallel

computers (MPPs) and to address intersite security issues that currently hinder the coupling of components across multiple sites.

... This project involves, first, development of the basic techniques required to achieve coexistence of commodity object technologies and Grid technologies, and second, the evaluation of these techniques in the context of NPSS-oriented challenge problems.

The work on basic techniques seeks to understand how "commodity" technologies (CORBA, DCOM, Excel, etc.) can be used in concert with specialized Grid technologies (for security, MPP scheduling, etc.). In principle, this coordinated use should be straightforward because of the Globus and IPG philosophy of providing low-level Grid mechanisms that can be used to implement a wide variety of application-level programming models. (Globus technologies have previously been used to implement Grid-enabled message-passing libraries, collaborative environments, and parameter study tools, among others.) Results obtained to date are encouraging: a CORBA to Globus resource manager gateway has been successfully demonstrated that allows the use of CORBA remote procedure calls (RPCs) to control submission and execution of programs on workstations and MPPs; a gateway has been implemented from the CORBA Trader service to the Grid information service; and a preliminary integration of CORBA and Grid security mechanisms has been completed.

[101] **A CORBA-based Development Environment for Wrapping and Coupling Legacy Codes**, G. Follen, C. Kim, I. Lopez, J. Sang and S. Townsend. In *Tenth IEEE International Symposium on High Performance Distributed Computing*. 2001. San Francisco.http://cnis.grc.nasa.gov/papers/hpdc-10_corbawrapping.pdf

http://cnis.grc.nasa.gov/papers/hpdc-10_corbawrapping.pdf

[102] **LaunchPad**. http://www.ipg.nasa.gov/launchpad/launchpad

The IPG Launch Pad provides access to compute and other resources at participating NASA related sites. The initial release provides the ability to:

- Submit Jobs to "batch" compute engines
- Execute commands on compute resources
- Transfer files between two systems
- Obtain status on systems and jobs
- Modify the user's environment

Launch Pad was developed using the Grid Portal Development Kit created by Jason Novotny.

[103] **HotPage**. https://hotpage.npaci.edu/

HotPage [is] the NPACI Grid Computing Portal. HotPage enables researchers to find information about each of the resources in the NPACI computational grid, including technical documentation, operational status, load and current usage, and queued jobs.

New tools allow you to:

- obtain a portal account online.
- personalize your view of the status bar.
- access and manipulate your files and data once you are logged in.
- submit, monitor, and delete jobs on HPC resources.

[104] **An Online Credential Repository for the Grid: MyProxy**, J. Novotny, S. Tuecke and V. Welch. In *Tenth IEEE International Symposium on High Performance Distributed Computing*. 2001. San Francisco.http://www.globus.org/research/papers.html#MyProxy

Grid Portals, based on standard Web technologies, are increasingly used to provide user interfaces for Computational and Data Grids. However, such Grid Portals do not integrate cleanly with existing Grid security systems such as the Grid Security Infrastructure (GSI), due to lack of delegation capabilities in Web security mechanisms. We solve this problem using an online credentials repository system, called MyProxy. MyProxy allows Grid Portals to use the GSI to interact with Grid resources in a standard, secure manner. We examine the requirements of Grid Portals, give an overview of the GSI, and demonstrate how MyProxy enables them to function together. The architecture and security of the MyProxy system are described in detail.

[105] **MyProxy**, NCSA. http://www.ncsa.uiuc.edu/Divisions/ACES/MyProxy/

MyProxy provides a server with client-side utilities to store and retrieve medium-term lifetime (of order a week) delegated X.509 credentials via the Grid Security Infrastructure (GSI). The myproxy-init program delegates a proxy credential to the myproxy-server, which stores the proxy to disk. The myproxy-get-delegation program retrieves stored proxy credentials from the myproxy-server. The myproxy-destroy program removes credentials stored on a myproxy-server.

[106] **NSF Middleware Initiative (NMI)**. http://www.nsf-middleware.org/

A new package of software and other tools will make it easier for U.S. scientists, engineers and educators to collaborate across the Internet and use the Grid, a group of high-speed successor technologies and capabilities to the Internet that link high-performance networks and computers nationwide and around the world.

The package of "middleware," or software and services that link two or more otherwise unconnected applications across the Internet, was developed under the auspices of the National Science Foundation's (NSF) Middleware Initiative (NMI). NSF launched the initiative in September 2001 by commiting $12 million over three years to create and deploy advanced network services that simplify access to diverse Internet information and services.

NMI Release 1.0 (NMI-R1) represents the first bundling of such Grid software as the Globus Toolkit, CondorG and the Network Weather Service, along with security tools and best practices for enterprise computing such as eduPerson and Shibboleth. By wrapping them in a single package, NMI project leaders intend to ease the use and deployment of such middleware, making distributed, collaborative environments such as Grid computing and desktop video-conferencing more accessible.

[107] **IBM and Department of Energy Supercomputing Center to Make DOE Grid Computing a Reality - DOE Science Grid To Transform Far-Flung Supercomputers into a Utility-like Service**. http://www.nersc.gov/news/IBMgrids032202.html

ARMONK, NY and BERKELEY, CA, March 22, 2002 -- IBM and the U.S. Department of Energy's (DOE) National Energy Research Scientific Computing Center (NERSC) today announced a collaboration to begin deploying the first systems on a nationwide computing Grid, which will empower researchers to tackle scientific challenges beyond the capability of existing computers.

Beginning with two IBM supercomputers and a massive IBM storage repository, the DOE Science Grid will ultimately grow into a system capable of processing more than five trillion calculations per second and storing information equivalent to 200 times the number of books in the Library of Congress. The collaboration will make the largest unclassified supercomputer and largest data storage system within DOE available via the Science Grid by December 2002 -- two years sooner than expected.