543018

Using Computing and Data Grids for Large-Scale Science and Engineering

William E. Johnston[1]
National Energy Research Scientific Computing Division,
Lawrence Berkeley National Laboratory and
Numerical Aerospace Simulation Division, NASA Ames Research Center

Abstract

We use the term "Grid" to refer to a software system that provides uniform and location independent access to geographically and organizationally dispersed, heterogeneous resources that are persistent and supported. These emerging data and computing Grids promise to provide a highly capable and scalable environment for addressing large-scale science problems. We describe the requirements for science Grids, the resulting services and architecture of NASA's Information Power Grid ("IPG") and DOE's Science Grid, and some of the scaling issues that have come up in their implementation.

Keywords: Grids, distributed computing architecture, distributed resource management, security

1 Introduction

"Grids" (see [1]) are an approach for building dynamically constructed problem solving environments using geographically and organizationally dispersed high performance computing and data handling resources.

Functionally, Grids are tools, middleware, and services for

- providing a uniform look and feel to a wide variety of distributed computing and data resources
- supporting construction, management, and use of widely distributed application systems
- facilitating human collaboration and remote access to, and operation of, scientific and engineering instrumentation systems
- managing and securing this computing and data infrastructure

This is accomplished through a set of *uniform software services* (the Common Grid Services - described in more detail below) that manage and provide access to heterogeneous resources may be summarized as

♦ information services	♦ resource specification and request
♦ resource co-scheduling	♦ data access
♦ authentication and authorization	♦ security services
♦ auditing	♦ monitoring
♦ global event services	♦ global queuing
♦ data cataloguing	♦ resource brokering
♦ collaboration and remote instrument services	♦ data location management
♦ communication services	♦ fault management

[1] wejohnston@lbl.gov, http://www.itg.lbl.gov; wej@nas.nasa.gov, http://www.ipg.nasa.gov

The overall motivation for the current large-scale (multi-institutional) Grid projects is to enable the resource interactions that facilitate large-scale science and engineering such as aerospace systems design, high energy physics data analysis, climatology, large-scale remote instrument operation, etc.

The vision for computing, data, and instrument Grids is that they will provide significant new capabilities to scientists and engineers by facilitating *routine* construction of information based problem solving environments that are built on-demand from large pools of resources. That is, Grids will routinely – and easily, from the user's point of view – facilitate applications such as:

- coupled, multidisciplinary simulations too large for single computing systems (e.g., multi-component turbomachine simulation – see [2])
- management of very large parameter space studies where thousands of low fidelity simulations explore, e.g., the aerodynamics of the next generation space shuttle in its many operating regimes (from Mach 27 at entry into the atmosphere to landing)
- use of widely distributed, federated data archives (e.g., simultaneous access to metrological, topological, aircraft performance, and flight path scheduling databases supporting a National Air Transportation Simulation system)
- coupling large-scale computing and data systems to scientific and engineering instruments so that complex real-time data analysis results can be used by the experimentalist in ways that allow direct interaction with the experiment (e.g. Cosmology data analysis involving telescope and satellite interaction, and coupling to simulations)
- single computational problems too large for any single system (e.g. extremely high resolution rotocraft aerodynamic calculations)

2 Requirements for Grids

Analysis of several scientific application areas provides a broad set of requirements for Grids. Examples from large-scale engineering and large-scale science are presented, and requirements are derived. Many of the requirements overlap, especially for the toolbuilders, and these are presented only in the engineering section.

2.1 Engineering Examples and Requirements

The NASA Aerospace Engineering Systems arena provides both the perspective of the computational scientists who build computational design tools, and of the design engineer / analyst who must use those tools to accomplish a specific task ([3]). In summary, these requirements include:

- Discipline analyst / problem solver requirements
  - multiple datasets maintained by discipline experts at different sites that support both geometric and computational design processes must be accessed and updated by many collaborating analysts
  - analysts must be able to securely share all aspects of their work process
  - techniques are needed for coupling heterogeneous computer codes, resources, and data sources in ways so that they can work on integrated/coupled problems in order to provide whole system simulations ("multi-disciplinary simulation/optimization")
  - interfaces to data and computational tools must provide appropriate levels of abstraction for discipline problems solving

- Discipline toolbuilder requirements
  - process and workflow management techniques must provide transparent and uniform control over all distributed resources participating in problem solving environments
  - workflow definition should be possible via "visual programming" scenarios that integrate with the analyst "desktop" environment
  - new approaches to computational simulation and data analysis must be accommodated in the distributed work/resource environment
  - collaborative, multi-party sharing of user interfaces, data, instruments, and computation must be provided
  - techniques are needed to describe and manage diverse strategies for parameter space exploration/filling
  - tools need to automatically manage and catalogue the numerous datasets the result from parameter studies
  - mechanisms for managing generalized "faults" are required for all aspects of the working environment
  - location and architecture independent services must provide for various interprocess, interactive, data-intensive, and multi-point communication
  - techniques are needed for debugging distributed software for correctness and performance
  - it must be possible to audit and account for use of all resources
  - co-allocation of resources to support coordinated use of multiple resources and scheduled use of resources must be available, and must accommodate "fuzzy" reservation (resource needed sometime in a given period)
  - policy based quality-of-service should be available for all resources, including supporting construction of systems that have various "real-time" operating constraints
  - systems and operations professionals must be able to manage the distributed resources as part of a computing environment
  - resources should be "immune" to unauthorized access and manipulation
  - resource stakeholders/owners should have easily used mechanisms to enforce their use conditions and this must accommodate "fluid" work groups
  - the security and access control services must provide for easily specified characteristics and must be easily integrated into applications and problem solving environments
  - CPU resource queuing mechanisms must provide a general and flexible control over all aspects of enqueued and running jobs
  - global event management facilities must be able to signal job actions and application states
  - use of CORBA, Java, Java/RMI, and DCOM must be provided within the context of the distributed resource environment
  - tools are needed to manage distributed heterogeneous computing architectures
  - generalized resource discovery services are needed in order to provide readily available and detailed resource information
  - support for remote execution management should include automatic selection and installation of binaries and libraries appropriate for the target platform

Many of these requirements are common to all of the science and engineering communities that will use Grids, and in the following two examples we only discuss additional requirements or different emphasis.

2.2 Science Examples

DOE's High Energy and Nuclear Physics (HENP) data analysis paradigm [4] epitomizes the large-scale data analysis environment. Hundreds to thousands of physicists from hundreds of institutions in countries around the world must analyze terabytes of data generated at a single site (the accelerator and detector). The analysis results must then be recombined in a common database. Requirements of the HENP community in addition to those above derive from a data analysis environment that is dominated by managing the location of data to be analyzed. Data must be moved around global networks and cached in, e.g., regional data centers, and from there to investigator data centers, etc. Requirements for this environment in addition to those above include:

- comprehensive network monitoring to locate, analyze, and correct bandwidth bottlenecks
- data replica catalogues to provide global views of cached data
- the methodology and implementation of incorporating, using, and managing resources in the overall environment must be scalable to thousands of resources

Existing and planned cosmology observational programs in DOE's observational and computational cosmology program (for example Cosmic Microwave Background measurements and the Supernova Cosmology Program ([5], [6])) are generating sufficiently large and complex data sets that they will require many researchers using the facilities and expertise of multiple large scientific computing centers if they are to be successful scientifically.

These (and many other) widely distributed scientific collaboration environments involve several processes:

- remote production and remote storage of data
- community processing of data
- community execution of simulations
- management of multi-step data analysis and simulations where software components and data will use computing and storage resources at different sites
- remote instrument interactions

All of these processes incrementally contribute to an overall buildup of a scientific knowledge base, e.g. the accumulation of information from an on-going observational program, that is managed as part of the collaboration. This management involves the additional element of being highly collaborative, with contributions coming from many sources, but with the requirement of a overall, or project, view of this information.

These circumstances have lead to specific requirements for workflow and collaboration frameworks beyond those noted above. The basic workflow framework must provide for:

- describing and managing multi-step, asynchronous component workflows, including managing fault detection and recovery
- access to data and metadata publication and subscription mechanisms
- event mechanisms - e.g. notification of when data or simulation results come into existence anywhere in the space of resources of interest
- user interfaces to each of the above

Collaborative work support must address the human interaction aspects of collaborative data analysis:

- maintenance of shared knowledge bases that allow a distributed community to create and update information about the state of overall progress of data processing, simulation results, existence of new, more highly refined, derived data, etc.
- support for collaborative processing of data
- support for on-line meetings, document sharing, and messaging
- establishment and maintenance of the collaboration membership
- security and management of access rights for the collaboration data and information

Remote instrument interactions must be possible: Techniques are needed for coupling remote instrument system operation and data streams directly to computing and data management resources. Such systems should interoperate with tools supporting human sharing of computing environments.

The requirements of these several application areas lead to a characterization of the desired Grid functionality. This functionality may be represented as a hierarchically structured set of services and capabilities which are described below, and who's interrelationship is illustrated in Figure 1.

3 An Overall Model for Grids

Grid environments that meet the requirements noted above have services at a number of "levels". There are services that provide the user interfaces and application regime workflow management, tools and services supporting the development of application programs, the basic Grid services that provide uniformity and access to resources, and there are the resources themselves. Ancillary services such as security and system management are required at all levels.

3.1 Problem Solving Environments: User Interfaces and Workflow Management

The User Interface

A number of services directly support using the Grid, e.g., by engineers or scientists. These include the toolkits for construction of application frameworks / problem solving environments (PSE) that integrate Grid services and applications into the "desktop" environment. For example, the graphical components ("widgets" / applets) for application user interfaces and control of the computer mediated, distributed human collaboration that support interface sharing and management, the tools that access the resource discovery and brokering services, the tools that provide generalized workflow management services such as resource scheduling, and managing high throughput jobs, etc. Examples include SciRun [7], Ecce [8], and WebFlow [9].

Workflow Management

Reliable operation of large and complex data analysis and simulation tasks requires methods for their description and control. A workflow management system must provide for a rich and flexible description of the analysis processes and their inter-relationships, and also provide mechanisms for fault detection and recovery strategies in widely distributed systems. Then, through the use of appropriate distributed system services (i.e. the Grid Common Services described below), the workflow system will map these activities onto a sufficiently large and diverse set of computing and data handling resources (one of the goals of Grids is to provide such a pool) in order to not only accommodate the routine processing, but it have sufficient elasticity in the system to be able to rapidly locate and configure alternate resources in the event of faults.

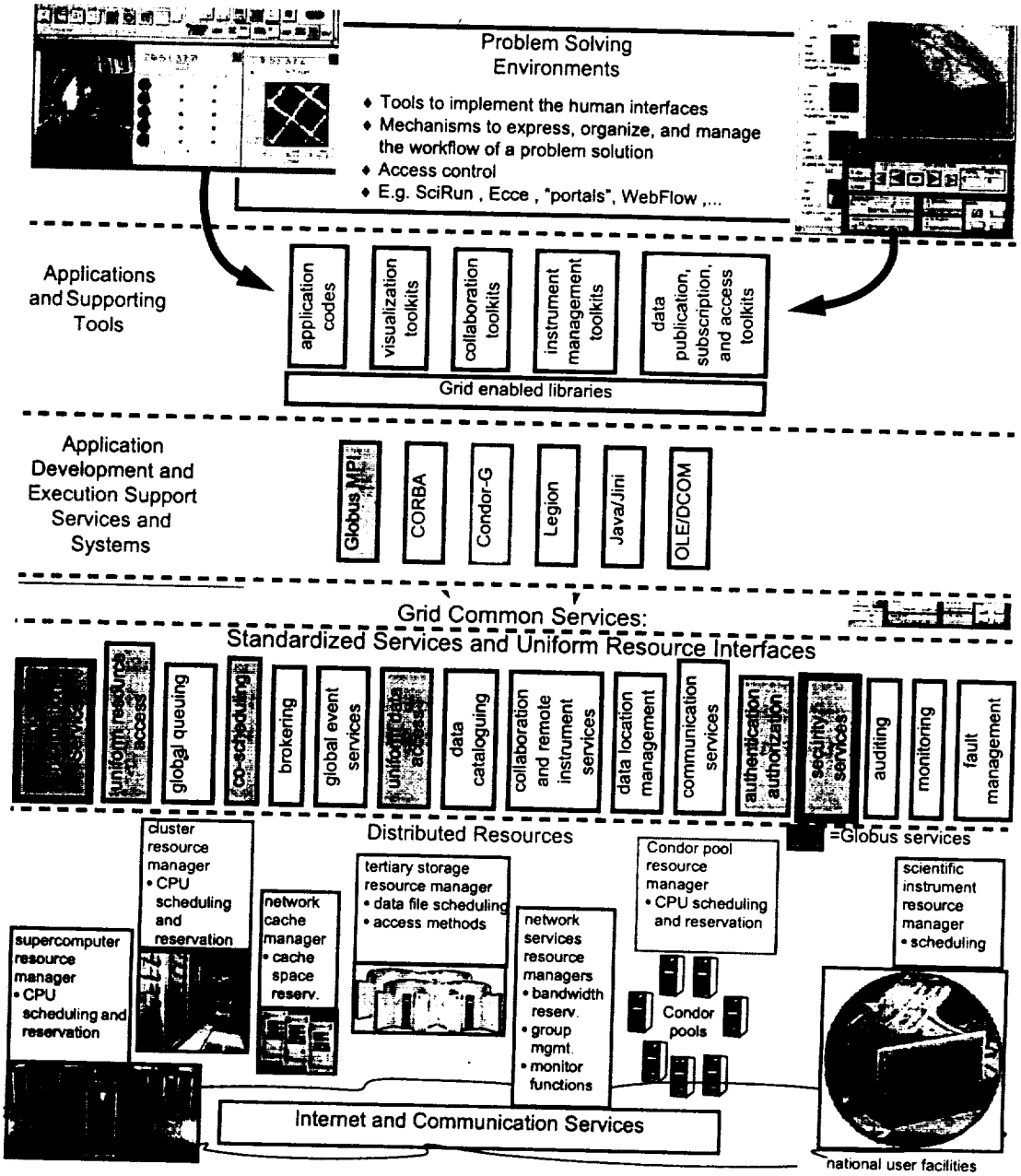

Figure 1 **A Grid Architecture - "upper" layers (top) and "lower" layers (bottom)**

3.2 Programming Services

Tools and techniques are needed for building applications that run in Grid environments. These cover a wide spectrum of programming paradigms, and must operate in a multi-platform, heterogeneous computing environments. For example, Grid enabled MPI [10], Java bindings to Grid services, CORBA [12] integrated with Grid services, Condor [13], Java/RMI [14], and perhaps DCOM [16], are all application oriented middleware systems that will have to interoperate with the Grid services in order to gain access to the resources managed by the Grid.

3.3 Grid Common Services

"Grid Common Services" refers to the basic services that provide uniform and location independent access and management of distributed resources. Much of the operational effort to run Grids is involved in maintaining these services.

Many Grids (including NASA's IPG and DOE's Science Grid) currently use Globus [18] to provide the basic services to characterize and locate resources, initiate and monitor jobs, and provide secure authentication of users.

Execution Management:

Several services are critical to managing the execution of application codes in the Grid. The first is resource discovery and brokering. By discovery we mean the ability to ask questions like: how to find the set of objects (e.g. databases, CPUs, functional servers) with a given set of properties; how to select among many possible resources based on constraints such as allocation and scheduling; how to install a new object/service into the Grid; and how make new objects known as a Grid service. The second is execution queue management, which relates to global views of CPU queues and their user-level management tools. The third category is distributed application management. The last category includes tools for generalized fault management mechanisms for applications, and for monitoring and supplying information to knowledge based recovery systems.

Runtime:

Runtime services include checkpoint/restart mechanisms, access control, a global file system, and Grid communication libraries such as a network-aware MPI that supports security, reliable multicast and remote I/O.

Uniform naming and location transparent access must be provided to resources such as data objects, computations, instruments and networks. This, in turn requires uniform I/O mechanisms (e.g. read, write, seek) for all access protocols (e.g. http, ftp, nfs, Globus Access to Secondary Storage, etc.) and richer access and I/O mechanisms (e.g. "application level paging") that are present in existing systems.

High-speed, wide area, access to tertiary storage systems will always be critical in the science and engineering applications that we are addressing, and we require data management services to provide global naming and uniform access. High-performance applications require high-speed access to data files, and the system must be able to stage, cache, and automatically manage the location of local, remote and cached copies of files. We are also going to need the ability to dynamically manage large, distributed "user-level" caches and "windows" on off-line data. Support for object-oriented data management systems will also be needed.

Services supporting collaboration and remote instrument control, such as secure, reliable group communication ("multicast") are needed. In addition, application monitoring and application characterization, prediction, and analysis, will be important for both users and the managers of the Grid.

Finally, monitoring services will include precision time event tagging for distributed, multi-component performance analysis, as well as generalized auditing of data file history and control flow tracking in distributed, multi-process simulations.

Environment Management:

The key service that is used to manage the Grid environment is the "Grid Information Service." This service – currently provided by Globus GIS (formerly MDS, see [20]) – maintains detailed characteristics and state information about all resources, and will also need to maintain dynamic performance information, information about current process state, user identities, allocations and accounting information.

3.4 Resource Management for Co-Scheduling and Reservation

One of the most challenging and well known Grid problems is that of scheduling scarce resources such as supercomputers and large instruments. In many, if not most, cases the problem is really one of co-scheduling multiple resources. Any solution to this problem must have the agility to support transient experiments based on systems built on-demand for limited periods of time. CPU advance reservation scheduling and network bandwidth advance reservation are critical components to the co-scheduling services. In addition, tape marshaling in tertiary storage systems to support temporal reservations of tertiary storage system off line data and/or capacity is likely to be essential. The basic functionality for co-scheduling and/or resource reservation usually must be provided by the individual resource managers.

3.5 Operations and System Administration

Implementing a persistent, managed Grid requires tools for deploying and managing the system software. In addition, tools for diagnostic analysis and distributed performance monitoring are required, as are accounting and auditing tools. Operational documentation and procedures are essential to managing the Grid as a robust production service.

3.6 Access Control and Security

The first requirement for establishing a workable authentication and security model for the Grid is to provide a single-sign-on authentication for all Grid resources based on cryptographic credentials that are maintained in the users desktop / PSE environment(s) or on one's person. In addition, end-to-end encrypted communication channels are needed in for many applications in order to ensure data integrity and confidentiality. This is provided by X.509 identity certificates (see [21]) together with the Globus security services.

The second requirement is an authorization and access control model that provides for management of stakeholder rights (use-conditions) and trusted third parties to attest to corresponding user attributes. A policy-based access control mechanism that is based on use-conditions and user attributes is also a requirement. Several approaches are being investigated for providing these capabilities.

3.7 Services for Operability

To operate the Grid as a reliable, production environment is a challenging problem. Some of the identified issues include management tools for the Grid Information Service that provides global information about the configuration and state of the Grid; diagnostic tools so operations/systems staff can investigate remote problems, and; tools and common interfaces for system and user administration, accounting, auditing and job tracking. Verification suites, benchmarks, regression

analysis tools for performance, reliability, and system sensitivity testing are essential parts of standard maintenance.

3.8 Grid Architecture: How do all these services fit together?

We envision the Grid as a layered set of services (see Figure 1) that manage the underlying resources, and middleware that supports different styles of usage (e.g. different programming paradigms and access methods).

However, the implementation is that of a continuum of hierarchically related, independent and interdependent services, each of which performs a specific function, and may rely on other Grid services to accomplish its function.

Further, the "layered" model should not obscure the fact that these "layers" are not just APIs, but usually a collection of functions and management systems that work in concert to provide the "service" at a given "layer." The layering is not rigid, and "drill down" (e.g. code written for specific system architectures and capabilities) must be easily managed by the Grid services.

4 NASA's Information Power Grid

For NASA, Grids such as IPG effectively represent a new business model for operational organizations delivering large-scale computing and data resources. The goal of Grids in this environment is to deliver these services in ways that allow them to be integrated with other widely distributed resources controlled, e.g., by the user community. Implementing this service delivery model requires two things: First, tools for production support, management, and maintenance, of integrated collections of widely distributed, multi-stakeholder resources must be identified, built, and provided to the systems and operations staffs. Second, organizational structures must be evolved that account for the fact that operating Grids is different than operating traditional supercomputer centers, and management and operation of this new shared responsibility service delivery environment must be explicitly addressed.

4.1 How is IPG Being Built

A strategy for building multi-site Grids has evolved over the past two years at NASA, and this experience is summarized here. (Also see [19].)

1) Establish an Engineering Working Group that involves the Grid deployment teams at each site.
   - schedule weekly meetings / telecons
   - if at all possible involve a Globus consultant in these meetings
   - establish an Engineering Working Group email list that includes everyone involved in

2) Identify the computing and storage resources to be incorporated into the Grid.

3) Set up liaisons with the systems administrators for all systems that will be involved (computing and storage).

4) Build Globus on a test system and validate the operation of the GIS/MDS at multiple sites.
   - use PKI authentication and Globus or IPG issues certificates for this test environment

5) Determine the model of operation for the Grid Information service (MDS).
   - decide on Netscape LDAP hierarchy ("classic model") vs. Globus OpenLDAP model (almost certainly the Globus OpenLDAP approach for new Grids)
   - establish the GIS/resource namespace (Look at other Grids and see what they have done, then think carefully about the implications of your namespace and its relationship to the other institutions that might eventually participate. o=Grid might be a meaningful top level at some point.)
   - plan on a GIS sever at each distinct site
   - get the GIS operational

6) Build and test the security infrastructure.
   - assuming a PKI based GSI, set up an X.509 Certification Authority
   - issue host certificates for the resources
   - count on revoking and re-issuing all of the certificates at least once before going operational
   - validate correct operation of the GSI, GSI ssh, GSI ftp, etc.

7) Establish the conventions for the Globus mapfile.
   - maps user Grid identities to system UIDs – this is the basic authorization mechanism for each individual platform – compute and storage
   - establish the connection between user accounts on individual platforms and requests for Globus access on those systems (initially a non-intrusive mechanism such as email to the responsible sys admin to modify the mapfile is best)

8) Validate network connectivity between the sites and investigate firewall issues.
   - Globus can be configured to use a restricted range of ports, but it still needs a handful of open ports
   - GIS/MDS also needs some open ports

9) Establish a user help mechanism.
   - Grid user email list
   - if possible, a trouble ticket system
   - Web pages with pointers to documentation and examples
   - a Globus "Quick Start Guide" that is modified to be specific to your Grid, with examples that will work in your environment

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

Next step is to build a prototype-production environment.

10) Deploy and build Globus on at least two computing platforms at two different sites.

11) Establish GIS servers at each major site

12) Establish the relationship between Globus job submission and the local batch schedulers (one queue, several queues, a Globus queue, etc.)

13) Validate operation of this configuration.

14) Establish the model for moving data between the Grid systems and user systems.
- GSIftp (or GridFTP) servers should be deployed on the computing platforms and on the data storage platforms
- it may be necessary to disable the Globus restriction on forwarding of user proxies by third parties in order to allow, e.g., a job submitted from platform_1@site_A to platform_1@site_B to write back to a storage systems at site A (platform_2@site_A) (This issue will be addresses in the future with restricted proxy certificates. See the work in the Grid Forum [23] Security Working Group.)
- determine if any user systems will manage user data that is to be used in Globus jobs – if so, the GSI/GRID ftp server should be installed on those systems (so that data may be moved from user system to user job on computing platform, and back)
- validate that all of these data paths work correctly

15) Designate/train some Globus application specialists. These specialists should:
- be running sample jobs as soon as the prototype-production system is operational
- serve as the interface between users and the Globus system administrators

16) Identify early users and have the Globus application specialists assist them in getting applications running on the Grid.

17) Decide on a Grid job tracking and monitoring strategy.

18) Put up one of the various Web portals for Grid resource monitoring.

4.2 What is the State of IPG?

The Dec., 2000 version of IPG represents an operational and persistent, "large-scale" prototype-production Grid providing access to computing, data, and instrument resources at three NASA Centers around the country in order that large-scale problems are more easily addressed than is possible today.

This baseline IPG system (Figure 2) includes:
- approximately 600 CPU nodes in half a dozen SGI Origin 2000s at three NASA sites
- several PC clusters and Solaris systems
- a large Condor pool (currently about 300) integrated with Globus
- several Terabytes of uniformly accessible mass storage
- wide area network interconnects of at least 100 mbit/s
- a stable and supported operational environment

The components of this prototype production Grid are reflected in the IPG Engineering Working Group tasks (see [3]): 30+ tasks have been identified as critical, and groups have been organized around the major task areas:
- identification and testing of computing and storage resources for inclusion in IPG
- deployment of Globus ([18]) as the initial IPG runtime system
- global management of CPU queues, and job tracking, and monitoring throughout IPG
- definition and implementation of a reliable, distributed Grid Information Service that characterizes all of the IPG resources
- public-key security infrastructure integration and deployment to support single sign-on using X.509 cryptographic identity certificates (see [21])

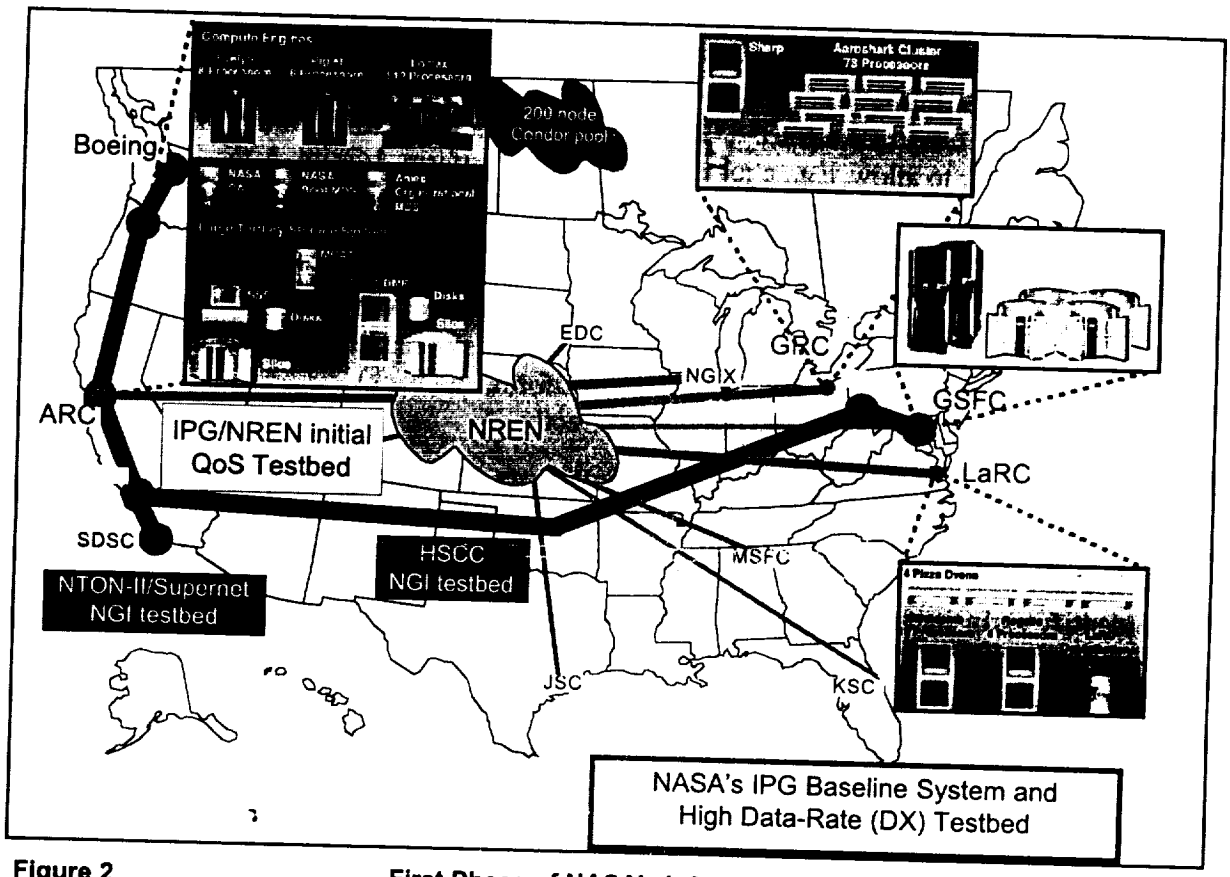

Figure 2 **First Phase of NASA's Information Power Grid**

- network infrastructure and QoS
- tertiary storage system metadata catalogue and uniform access system
- operational and system administration procedures for the distributed IPG
- user and operations documentation
- account and resource allocation management across systems with multiple stakeholders
- Grid MPI [10] and CORBA programming middleware systems integration
- high throughput job management tools
- distributed debugging and performance monitoring tools

4.3 What Types of Applications are Bring Run on IPG?

4.3.1 Data Mining

The University of Alabama in Huntsville has developed a data mining system called ADaM (Algorithm Development and Mining). The current design consists of a mining engine and a daemon-controlled database. The database contains information about the data to be mined including its type and its location. To mine for data, the user provides the mining engine with a mining plan that consists of the sequential list of mining operations that are to be performed along with any parameters that may be required for each mining operation. The mining engine consults the database in order to find out where the data to be mined is stored and then applies the mining

plan to the set of data that has been identified to the database. Each mining operation is represented as a shared-library file, one file per operation.

The IPG version of ADaM is structured so that the database and its associated daemon resides on a processor distinct from where the mining engine operates. For example, the database could be located on the user's workstation.

Using *globusrun*, the user is able to stage the mining engine to another processor to execute. As required, the mining engine will acquire mining operation executables in the form of shared-library files from an operator repository on the IPG. Since a single mining plan may involve only a handful of operators (out of the 70+ operations that ADaM currently support), this means that only the required mining operators need to be sent to the IPG node that is currently supporting the mining engine. This is accomplished using the Globus data transfer functions.

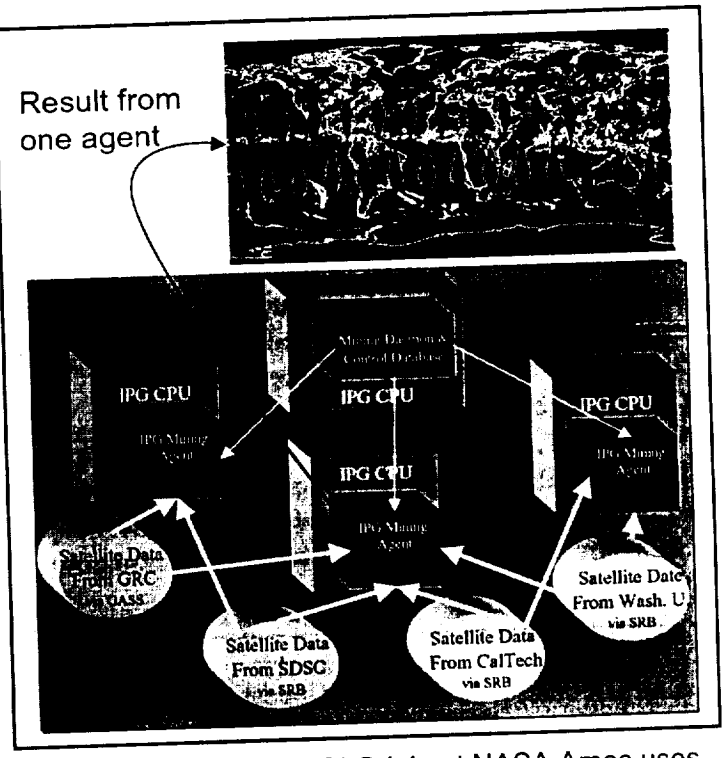

Figure 3 512 node SGI Origin at NASA Ames uses IPG uniform interface data access tools to simultaneously mine hydrology data from four sites.

As it executes, the mining engine stages the data to be mined from the data repository to the processor where the mining engine is executing. There are currently several sites that act as data repositories, and which currently pull data from NASA's Global Hydrology and Climate Center (which caches its most recent data holding in an FTP directory accessible through the web) so that it can be mined for severe storms.

This is work of Tom Hinke (thinke@mail.arc.nasa.gov).

4.3.2 Parameter Studies

ILab is an aerospace parameter study system that is designed to provide for substantial human efficiency in studying complex systems. It uses IPG to locate and manage compute resources for the individual jobs. See [24]. This is work of Maurice Yarrow (yarrow@nas.nasa.gov), NASA Ames.

The IPG Condor pool is used for another type of job. For example, a molecular design application coded in Java and managed by the Condor cycle scavenger is able to apply several gigaflop years of otherwise idle computing time to various problems in molecular design for nanotechnology devices and materials. These applications are coded in Java for platform independence, and the increased number of platforms where the code can run more than compensates for the

computational inefficiency of Java. See [25] and [26]. This is work of Al Globus (globus@nas.nasa.gov), NASA Ames.

4.3.3 High Latency Algorithm R&D

IPG provides services for aggregating computing resources in a parallel and distributed fashion. One future application of this will be single simulations that operate across many, widely distributed systems. Current algorithms, however, do not accommodate the high and variable latencies encountered in such a computing environment. The research branch of the NAS Division is investigating algorithms that are suitable for Grid computing environments. One candidate is overset grid codes that can tolerate timestep mis-matches on the intra-object boundaries. A version of the OVERFLOW, Navier-Stokes, CFD simulation code is being modified for this approach. It has been demonstrated operating across systems at ARC, GRC, and LaRC, solving for flow about large test objects mounted in a wind tunnel.This is work of Mohammad J. Djomehri (djomehri@nas.nasa.gov).

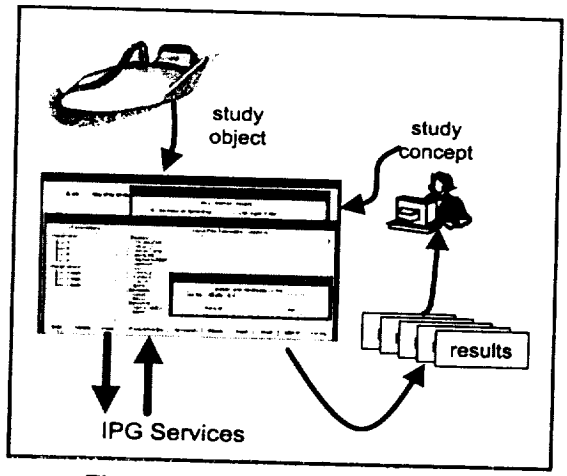

Figure 4 The ILab, aerospace parameter study system: A Grid based Problem Solving Environment.

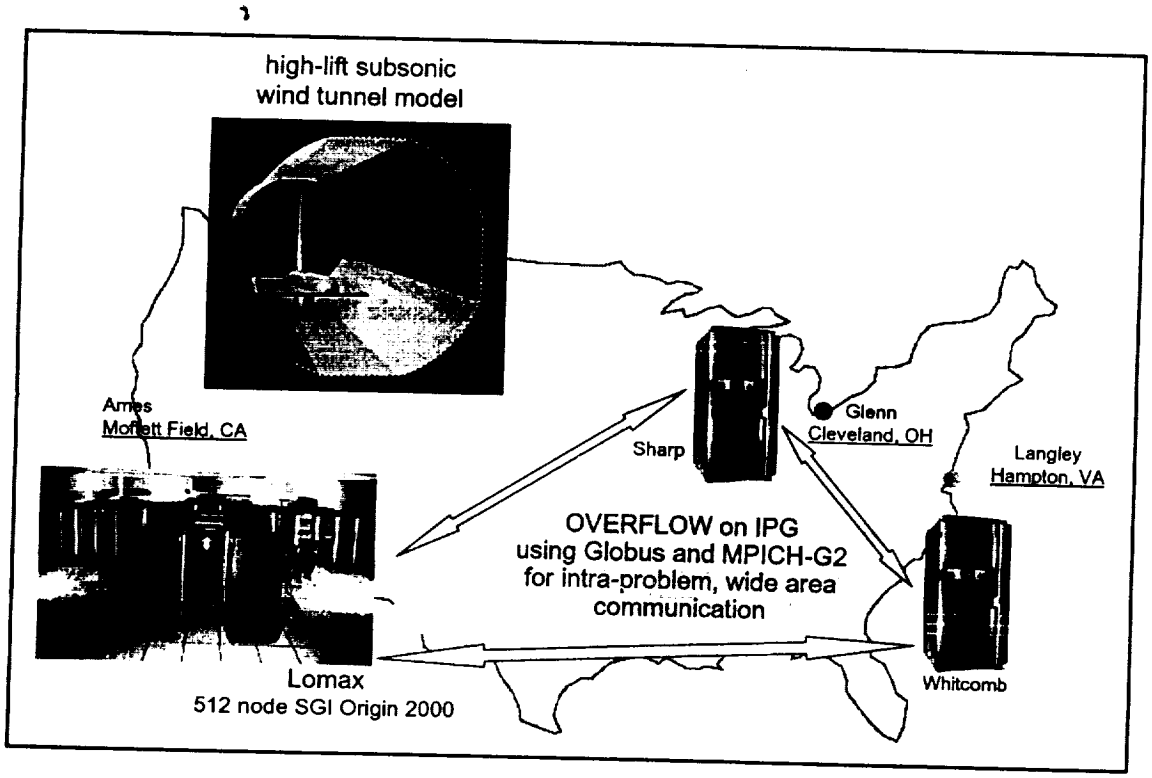

Figure 5 Experiments in latency tolerant algorithms.

5 Scaling Issues for Grids

Experience with NASA's IPG, DOE's High Energy Physics Grids, DOE's Energy Sciences Network's large-scale directory services, and the DOE2000 Collaboratory program's experience with security and PKI, can be used to identify a list of issues that, apart from the basic Grid software functionality, will also have to be addresses in order to build large-scale Grids for science. These include:

- Directory services
  - naming to support many virtual organizations that may have some resources and participants in common
  - scaling the directories themselves to support searches of thousands of resources across hundreds of organizations
  - support for general cataloguing services (e.g. data replica catalogues) within the Science Grid directory infrastructure and namespace
- Security
  - "root" X.509 identity certificate, Certification Authority (to sign the institutional CA certificates in order to facilitate cross-institutional identities)
  - Grid-wide Certificate Revocation List "repository"
- User service functions
  - a common trouble ticket system for collecting, codifying, and correcting problems
- "Fault" management
  - infrastructure-wide monitoring to identify performance bottlenecks in various resources (networks, computing, and storage)
- Integration with large tertiary storage systems
  - interface with local MSS groups to provide GridFTP [11]
- Integration with large numbers of computing resources
  - work with local sys admins to get Globus server side software and authorization files installed and debugged on local systems
- Resource usage auditing
  - provide tools and repositories for standard resource accounting records that can be used for project-level resource allocation management across the Grid
- Test and validation suites
  - provide standard Grid resource configuration validation test suites and facilitate the routine use and results analysis for these suites across the Grid

Work is being done in all of these areas, however in this paper we will look in detail only at information services.

5.1 The Grid Information Service

Grids will be global infrastructure, and will depend heavily on the ability to locate information about computing, data, and human resources for particular purposes, and within particular contexts. Further, most Grids will serve virtual organizations whose members are affiliated by a common administrative parent (e.g. the DOE Science Grid and NASA's Information Power Grid), common long-lived project (e.g. the High Energy Physics, Atlas experiment), common funding source, etc.

The user/functional requirements and operational requirements for a Grid Information Service to satisfy these needs presents a substantial problem in scaling the current LDAP based approaches. (This work is done in collaboration with Mike Helm (helm@fionn.es.net), Lawrence Berkeley National Lab.)

5.1.1 User Requirements

Searching

The basic sort of question that a GIS must be able to answer is for all resources in a virtual organization, provide a list of those with specific characteristics.

For example:

> "Within the scope of the Atlas collaboration, return a list of all Sun systems with at least 2 CPUs and 1 gigabyte of memory, that are running Solaris 2.6 or Solaris 2.7, and for which I have an allocation."

Answering this question involves examining both the virtual organization attribute and the resource attributes in order to produce a list of candidates.

Virtual Organizations

It should be possible to provide "roots" for virtual organizations. These nodes provide search scoping by establishing roots that sit at the top of a hierarchy of virtual org. resources, and therefore starting places for searches. Like other named objects in the Grid, these virtual org. nodes might have characteristics specified by attributes and values. In particular, the virtual organization node probably needs a name reflecting the org. name, however some names (e.g. for resources) may be inherited from the Internet DNS domain names. Virtual organizations may find it convenient to register with a Grid "root" so that they can share resources if policy allows. This is addressed by layers in Figure 6 labeled "root," "virtual org," and "resources."

Information and Data Objects

A variety of other information will probably require cataloguing and global access, and the GIS should accommodate this in order to minimize the number of long-lived servers that have to be managed:

- dataset metadata
- dataset replica information
- database registries
- Grid system and state monitoring objects
- Grid entity certification/registration authorities (e.g. X.509 Certificate Authorities)
- Grid Information Services object schema

Therefore it should be possible to create arbitrary nodes to represent other types of information, such as information object hierarchies.

This sort of information has to be consistently named in a global context, will have to be locatable, and in some cases will have an inherently hierarchical structure.

Requirements for these catalogues include:

- providing unique and consistent object naming
- access control

- searching, discovery, and publish/subscribe

5.1.2 Operational Requirements

Performance and Reliability

- Queries, especially local queries, should be satisfied in times that are comparable to other queries like uncached DNS data. E.g., seconds or fractions of seconds.
- Local sites should not be dependent on remote servers to locate and search local resources.
- It should be possible to restrict searches to local resources of a single, local, administrative domain.
- Site administrative domains may wish to restrict access to local information, and therefore will want control over a local, or set of local, information servers.

These imply the need for servers intermediate between local resources and the virtual org. root that are under local control for security, performance management, and reliability management. This is addressed by the layer labeled "local control" in Figure 6.

(Note that in the Globus terminology that these intermediate directory servers are called GIIS's.)

Multiple Membership

Many objects/resources will have membership in multiple virtual organizations. This information, like other resource attributes, will likely be maintained at the resources in order to minimize management tasks at the upper level nodes.

- It must be possible for a resource to register with multiple virtual organizations (note the multiply connected resources in Figure 6).

Minimal Manual Management

The management of the information servers above the resources (in the case of a resource catalogue) must be as automatic/minimal as possible.

- Information about a resource should be maintained at that resource, and should propagate automatically to superior information servers.

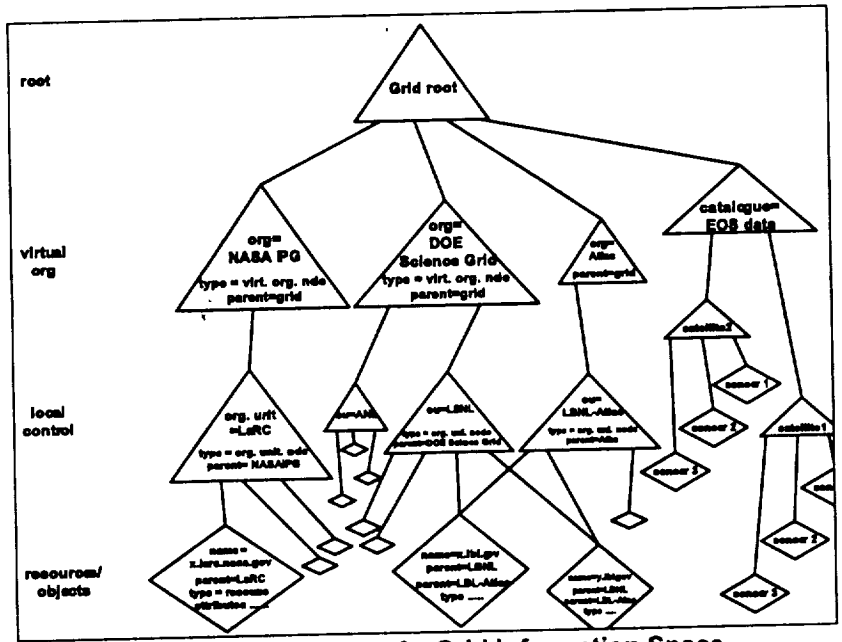

Figure 6 **Structure of a Grid Information Space.**

Control over Information Propagation

At each level of information management (four have emerged so far) there are various reasons why both import and export controls will have to be established.

- At the object / resource level (see Figure 7), the local administrators must have control over what information is exported for the purposes of registration.

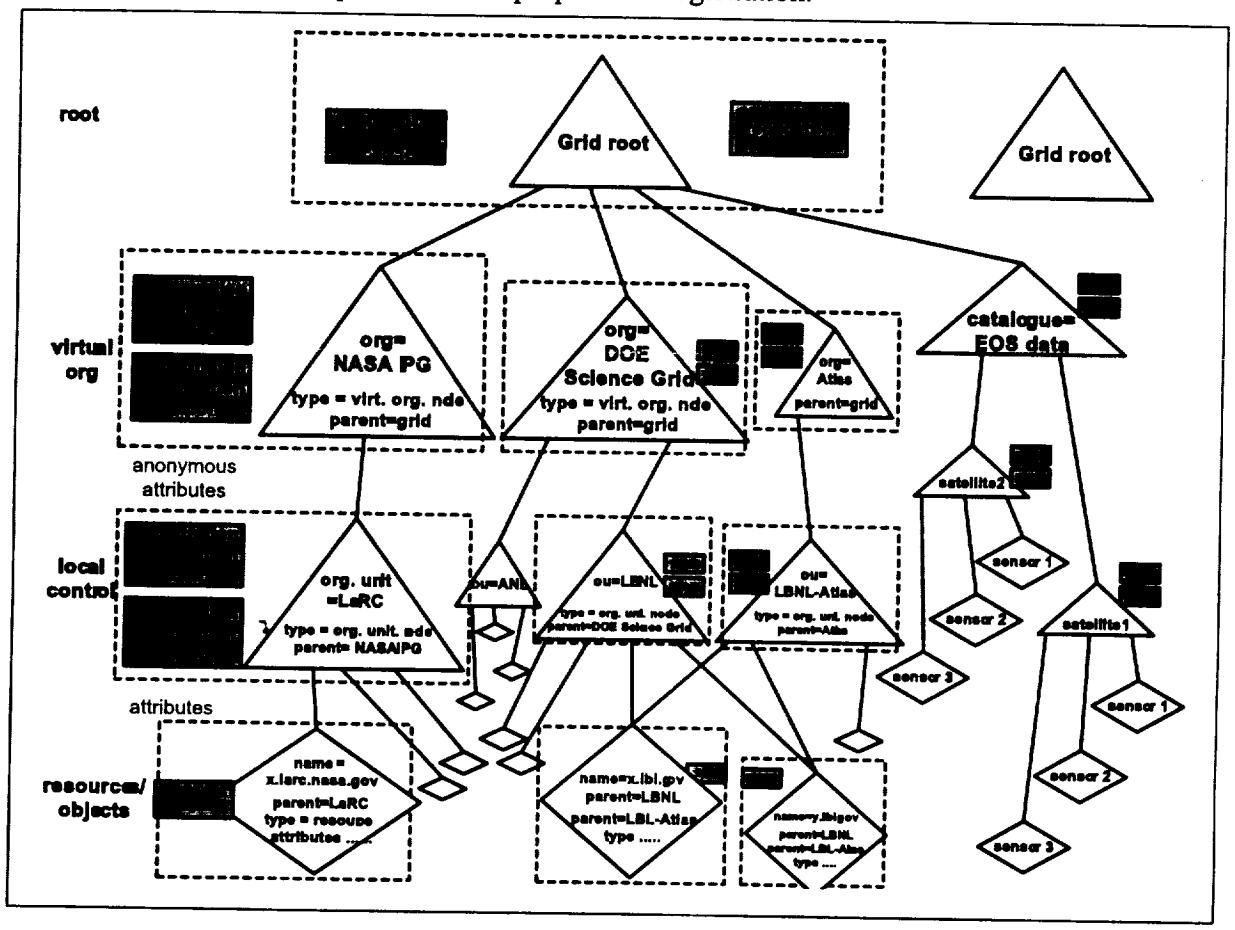

Figure 7 **Information import and export must be automatic and the content subject to management.**

- At the object / resource level there must be access control mechanisms to restrict the types of queries or the detail that queries return.
- The nodes at the level of "local control" are meant to model a common system administration domain, and must support a common security policy, including who is allowed to register (import control) and what information is passed outside of the security domain (export control). It should, e.g., be possible to implement policies such as making anonymous the information that is passed to the next level up (either for registration or as search results).
- Such anonymous information should allow broad searches at the upper levels, but limit specific searches to the lower levels, where searches can be authorized based on the relationship of the searcher to the resource.

- The same sorts of capabilities as exist at the local control level must be available at the virtual organization level in order to maintain control over the characteristics of the virtual organization
- At the root, again it must be possible to apply policy to registration (e.g. to prevent nodes below the virtual org. level from registering at the root).
- The ability to do automatic node replication for reliability will exist at all levels.
- Information import and export must be automatic and the content subject to management. See the "import/export" nodes in Figure 7.

Performance and Robustness

Finally, when we consider the information flow implied by the structure in Figure 7 it is apparent that a lot of information may flow in complex patterns. Well tested components and procedures will be needed. We will probably need some modeling or measurement information on the volume and rate of data flow in such an environment in order to assess the scaling issues. Warren Smith (wwsmith@nas.nasa.gov), NASA Ames, is working on this problem.

6 Conclusion

There are many challenges in making Grids a reality, in the sense that they can provide new capabilities in production quality environments.

While the basic Grid services have been demonstrated, e.g. in the IPG prototype demonstrated at the Supercomputing'1999 conference, and the GUSTO testbed ([22]) demonstrated in 1998, a general purpose computing, data management, and real-time instrumentation Grid involves many more services. One challenge is to identify the minimal set of such services and another is to scale the services to Grids that knit together resources at hundreds of sites.

In the case of the analysis of the information services scaling issues described above, Mike Helm of DOE's Energy Sciences Network is building on ESNet's long experience in running very large X.500 services to devise solutions for the issues noted above. (The solution will likely involve careful name space definition and management, and use of commercial meta-directory systems to off-load the GIS LDAP servers, etc. A paper on this will be available at http://www.lbl.gov/~mike/globus.) And Warren Smith of NASA Ames is building a performance modeling and simulation system for the GIS.

In addition to the NASA and DOE projects, Grids are being developed by a substantial and increasing community of people who work together in a loosely bound coordinating organization called the Grid Forum (www.gridforum.org - [23]).

7 Acknowledgements

Almost everyone in the NAS Division of the NASA Ames Research Center, numerous other people at the NASA Ames, Glenn, and Langley Research Centers, as well as many people involved with the NSF PACIs (especially Ian Foster, Argonne National Lab., Carl Kesselman, USC/ISI, Randy Butler, NCSA, and Reagan Moore, SDSC) have contributed to IPG. Bill Feiereisen, NAS DIvision Chief, provided the initial vision for IPG, and strong support for it's development and deployment. Bill Nitzberg (now with Veridian Systems), Dennis Gannon

(Indiana University), Leigh Ann Tanner, and Arsi Vaziri of NASA Ames have made special contributions to IPG. Tom Hinke, Maurice Yarrow, Al Globus, Mohammad J. Djomehri, and Warren Smith of NASA Ames, and Mike Helm, of DOE's ESNet, are all acknowledged in the text.

IPG is funded primarily by NASA's Aero-Space Enterprise, Information Technology (IT) program (http://www.nas.nasa.gov/IT/overview.html). DOE's Science Grid 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/production/octr/mics) under contract DE-AC03-76SF00098 with the University of California.

8 References

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

[2] Numerical Propulsion System Simulation (NPSS) - see http://hpcc.lerc.nasa.gov/npssintro.shtml

[3] "Information Power Grid." See http://www.ipg.nasa.gov for project information. The implementation plan, including a requirements analysis section, is located at http://www.ipg.nasa.gov/Engineering/requirements.html .

[4] See, e.g., http://www.cacr.caltech.edu/ppdg/

[5] Science Magazine Names Supernova Cosmology Project 'Breakthrough of the Year'," LBNL Research News, http://www.lbl.gov/supernova/

[6] "High Redshift Supernova Search Home Page of the Supernova Cosmology Project." See http://www-supernova.lbl.gov/

[7] SCIRun is a scientific programming environment that allows the interactive construction, debugging and steering of large-scale scientific computations. http://www.cs.utah.edu/~sci/software/

[8] Ecce - www.emsl.pnl.gov

[9] WebFlow - A prototype visual graph based dataflow environment, WebFlow, uses the mesh of Java Web Servers as a control and coordination middleware, WebVM: See http://iwt.npac.syr.edu/projects/webflow/index.htm

[10] Foster, I., N. Karonis, "A Grid-Enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems." Proc. 1998 SC Conference. Available at http://www-fp.globus.org/documentation/papers.html

[11] See http://www.globus.org/datagrid/

[12] Otte, R., P. Patrick, M. Roy, *Understanding CORBA*, Englewood Cliffs, NJ, Prentice Hall, 1996.

[13] Livny, M, et al, "Condor." See http://www.cs.wisc.edu/condor/

[14] Sun Microsystems, "Remote Method Invocation (RMI)." See http://developer.java.sun.com/developer/technicalArticles//RMI/index.html .

[15] Grimshaw, A. S., W. A. Wulf, and the Legion team, "The Legion vision of a worldwide virtual computer", *Communications of the ACM*, 40(1):39-45, 1997.

[16] Microsoft Corp., "DCOM Technical Overview." November 1996. See http://msdn.microsoft.com/library/backgrnd/html/msdn_dcomtec.htm .

[17] See, e.g., http://www.nas.nasa.gov/Main/Features/2001/Winter/launchpad.html

[18] Foster, I., C. Kesselman, Globus: A metacomputing infrastructure toolkit", *Int'l J. Supercomputing Applications*, 11(2);115-128, 1997. (Also see http://www.globus.org)

[19] "Hitchhiker's Guide to the Grid," http://www.ipg.nasa.gov/ -> User Support -> Documentation

[20] Fitzgerald, S., I. Foster, C. Kesselman, G. von Laszewski, W. Smith, S. Tuecke, "A Directory Service for Configuring High-Performance Distributed Computations." Proc. 6th IEEE Symp. on High-Performance Distributed Computing, pg. 365-375, 1997. Available from http://www.globus.org/documentation/papers.html .

[21] Public-Key certificate infrastructure ("PKI") provides the tools to create and manage digitally signed certificates.For more information, see, e.g.,RSA Lab's "Frequently Asked Questions About Today's Cryptography" http://www.rsa.com/rsalabs/faq/, *Computer Communications Security: Principles, Standards, Protocols, and Techniques*. W. Ford, Prentice-Hall, Englewood Cliffs, New Jersey, 07632, 1995, or *Applied Cryptography*, B. Schneier, John Wiley & Sons, 1996.

[22] "Globus Ubiquitous Supercomputing Testbed Organization" (GUSTO). At Supercomputing 1998, GUSTO linked around 40 sites, and provides over 2.5 TFLOPS of compute power, thereby representing one of the largest computational environments ever constructed at that time. See http://www.globus.org/testbeds .

[23] Grid Forum. The Grid Forum (www.gridforum.org) is an informal consortium of institutions and individuals working on wide area computing and computational grids: the technologies that underlie such activities as the NCSA Alliance's National Technology Grid, NAPES's Metasystems efforts, NASA's Information Power Grid, DOE ASIA's DISCOM program, and other activities worldwide.

[24] "Maurice Yarrow, Karen M. McCann, Rupak Biswas, and Rob F. Van der Wijngaart "An Advanced User Interface Approach for Complex Parameter Study Process Specification on the Information Power Grid." http://www.nas.nasa.gov/Research/Reports/Techreports/2000/nas-00-009-abstract.html

[25] Al Globus, Sean Atsatt, John Lawton, and Todd Wipke, "JavaGenes: Evolving Graphs with Crossover," 2000. http://www.nas.nasa.gov/~globus/papers/JavaGenes/paper.html

[26] Al Globus, Eric Langhirt, Miron Livny, Ravishankar Ramamurthy, Marvin Solomon, and Steve Traugott, "JavaGenes and Condor: Cycle-Scavenging Genetic Algorithms." Java Grande 2000, sponsored by ACM SIGPLAN, San Francisco, California, 3-4 June 2000. http://www.nas.nasa.gov/~globus/papers/JavaGrande2000/JavaGrandePaper.html