Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid

William E. Johnston[1], *Dennis Gannon*[2], *and Bill Nitzberg*[3]
Numerical Aerospace Simulation Division, NASA Ames Research Center, Moffett Field, CA

Abstract

Information Power Grid (IPG) is the name of NASA's project to build a fully distributed computing and data management environment – a Grid. The IPG project has near, medium, and long-term goals that represent a continuum of engineering, development, and research topics. The overall goal is to provide the NASA scientific and engineering communities a substantial increase in their ability to solve problems that depend on use of large-scale and/or dispersed resources: aggregated computing, diverse data archives, laboratory instruments and engineering test facilities, and human collaborators. The approach involves infrastructure and services that can locate, aggregate, integrate, and manage resources from across the NASA enterprise. An important aspect of IPG is to produce a common view of these resources, and at the same time provide for distributed management and local control. In addition to addressing the overall goal of enhanced science and engineering, there is a potential important side effect. With a large collection of resources that have common use interfaces and a common management approach, the potential exists for a considerable pool of computing capability that could relatively easily, e.g., be called on in extraordinary situations such as crisis response.

IPG is a collaboration between NASA and the NSF PACIs [1], and the initial set of distributed services are based on the Globus metacomputing system [5]. The near term goal of IPG is a prototype production Grid, which entails developing, deploying, and supporting infrastructure like LDAP information servers and PKI security services, and the engineering aspects of adapting R&D systems like Globus to a production environment.

1.0 Introduction

What is the Information Power Grid?

Computational Grids [4], e.g. NASA's Information Power Grid (IPG - www.nas.nasa.gov/IPG), will provide significant new capabilities to scientists and engineers by facilitating the solution of large-scale, complex, multi-institutional / multi-disciplinary, data and computational based problems using CPU, data storage, instrumentation, and human resources distributed across the NASA community. This entails technology goals of:

- Independent, but consistent, tools and services that support various programming environments for building applications in widely distributed environments
- Tools, services, and infrastructure for managing and aggregating dynamic, widely distributed collections of resources - CPUs, data storage / information systems, communications systems, real-time data sources and instruments, and human collaborators
- Facilities for constructing collaborative, application oriented workbenches / problem solving environments across the NASA enterprise based on the IPG infrastructure and applications. These constitute the primary science and engineering interface to Grids
- A common resource management approach that addresses, e.g., system management, user identification, resource allocations, accounting, security, etc.
- An operational Grid environment incorporating major computing and data resources at multiple NASA sites in order to provide an infrastructure capable of routinely addressing larger scale, more diverse, and more transient problems than is possible today.

What will IPG facilitate?

An environment with the characteristics noted above will enable NASA scientists to make strides in four classes of activities. First, it will allow for the construction and management of dynamic systems such as wide area testbeds and dynamically configured production environments. Second, it will allow NASA to prototype distributed systems that can adapt to future changes by using Grid services to flexibly manage changing environments, infrastructure, and resources.

Third, research teams will be able to construct just-in-time, large-scale systems to support scientific and engineering computing and data based activities that are not steady state, i.e. those that may require a different resource mix for every different problem. For example, simulations and their supporting computing platforms including data mining systems and their underlying data archives, instrumentation systems and human collaborators

1. wej@nas.nasa.gov
2. gannon@cs.indiana.edu
3. nitzberg@nas.nasa.gov (MRJ Technology Solutions, NASA contract NAS2-14303)

Finally, IPG will enable the routine use of wide area, data-intensive applications such as those involving remote access to high data-rate real-time data sources and instruments and large datasets as illustrated in [10].

Additionally, Grids and their services are intended to support "large-scale" environments, where "scale" refers to several dimensions:

- Large scale computational and storage capacity through aggregation of resources
- Scale in the complexity of resources independent of capacity. For example, data intensive computing tends to require a complex mix of resources, with or without high capacity. Management of very diverse data is one key problem and Grids will provide transparent access to these resources
- Scale in geographic and organizational scope
- Scale in the multiplicity of abstractions needed by different Grid user communities

How will IPG be accomplished?

Three main areas must be addressed in order to accomplish these goals:

1) new functionality and capability;
2) an operational environment that encompasses significant resources;
3) new services delivery model.

In the first area, Grids must provide services supporting uniform and location independent interfaces for aggregating, scheduling, and integrating numerous, diverse, and distributed resources.

Such services include resource description and discovery mechanisms; multi-party, secure, and fault tolerant communication; access control; data location management; job submission and data archive access; sharing mechanisms to support collaborative interfaces and toolkits for building problem solving environments.

Some of these services exist and some must be designed, built, and evaluated. These services will knit together and provide access to the many compute and data engines and scientific instruments that will provide significantly increased levels of computing and data analysis capability.

The operational system is discussed below.

In the third area, Grids such as IPG, effectively define a new business model for operational organizations delivering large-scale computing and data resources 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. The second requirement is that new 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.

What is the State of IPG?

Point 1), above, is being addressed by a detailed examination of requirements generated by several NASA application communities, both in terms of specific capabilities identified by the applications community and as the result of analysis of the requirements and desired operating environments by computer scientists.

Addressing point 2), the two year IPG goal is an operational and persistent, "large-scale" prototype-production Information Power Grid providing access to computing, data, and instrument resources at NASA Centers around the country so that applications that cannot be done today are enabled.

The first phase (targeted for 10/99) is a baseline operating system that includes:

- approx. 300 CPU nodes in half a dozen SGI Origin 2000s at three or four NASA sites
- several workstation clusters
- 30-100 Terabytes of uniformly accessible mass storage
- wide area network interconnects of at least 100 mbit/s
- a stable and supported operational environment

Addressing point 3), the NAS Division at Ames is identifying the new services that will be delivered by IPG, and is creating groups that will develop (as necessary), test, deploy, and support these services. In addition to new local organizational structure and local R&D, NAS is coordinating related activities at the NSF supercomputer centers and at universities to provide various components of the new operational model.

Current progress is reflected in the IPG Engineering Working Group tasks: 30+ tasks have been identified as critical for the baseline system. Task groups are working on each of these, and they fall into the general areas of:

- Identification and testing of computing and storage resources for inclusion in IPG
- Initial IPG runtime system (Globus) deployment
- Global management of CPU queues, tracking, and monitoring tools
- Resource discovery system deployment
- Public-key security component integration and deployment
- Network infrastructure and QoS
- Mass storage system metadata catalogue and uniform access system
- Operational and system administration procedures for distributed systems
- User and operations documentation

+ Account and resource allocation management across systems with multiple stakeholders
+ Globus/MPI [7], CORBA [11], and Legion [6] programming middleware systems integration
+ High throughput job management tools
+ Distributed debugging and performance monitoring tools

2.0 Building IPG

The strategy for building IPG first involves and on-going requirements analysis. This task requires a generalization of the application-specific requirements to specific service definitions. Next, one must identify existing implementations and/or designs of prototype Grid services and identify and incorporate the underlying resources. One must also construct “characteristic” applications for validation of the Grid services. Finally, one must define and implement the required services and infrastructure.

2.1 Characterizing the User Communities

It is important to recognize that there are several “customer” communities for Grid services, and some of these communities have rather different requirements. For example:

I) Lay public, schools, community emergency services, and military field units will access the Grid through Web browser and kiosks.
II) Application domain scientists and engineers will use Problem Solving Environments / application frameworks.
III) Application domain computational scientists and tool developers will use middleware that supports distributed computation, aggregated and federated access to catalogued data, computer mediated collaboration and multiple programming paradigms.
IV) Distributed system developers use job management, access control, generalized communications services, resource discovery and brokering
V) Middleware / Grid common service developers will use local resource managers (queuing, network QoS, scheduled tape marshaling), security services, network services and resource information bases

IPG is primarily addressing the communities II, III, and IV.

2.2 Requirements Analysis

IPG development and deployment will be driven by addressing requirements obtained by analyzing a number of different application areas. However, the overall goal is to address a broad spectrum of NASA’s computing, data management, real-time data source, and collaboration needs through a general and extensible large-scale, heterogeneous computing environment.

General capability and services requirements come from experience with the way science and engineering R&D uses computer related resources.

Specific IPG requirements come from analyzing NASA application and programs. These include the Computational Aero-Sciences (CAS - http://cas.arc.nasa.gov/) activities, and the Aerospace Engineering Systems (AES) project which involves large-scale, multidisciplinary design environments. Additional requirements come from AES components including simulation and design systems for airframes, turbomachinery, spacecraft and propulsion systems. The Intelligent System and Intelligent Synthesis Environment program (http://www.ise.nasa.gov/) which targets distributed collaborative design systems provides a large number of important requirements. Other important contributions come from Earth Sciences, the Data Assimilation Office (http://dao.gsfc.nasa.gov/) production meteorological simulations, Astrobiology (http://astrobiology.arc.nasa.gov), the Aviation / Space Station remote “help desk” and on-line instrumentation systems such as wind tunnels (http://www-darwin.arc.nasa.gov).

Analysis of the problem solving approaches in these disciplines gives rise to a long list of requirements, each of which is addressed in the various IPG technology goals.

A summary of the requirements derived from discussions with analysts / problem solvers in these discipline areas, and from examining their work processes, is given below.

1-0 Discipline Analyst / Problem Solver Requirements

1-1 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.
1-2 Analysts must be able to securely share all aspects of their work process.
1-3 Data streams from instrument systems must be available both in real-time to computational data analysis systems and via well catalogued databases.
1-4 Existing heterogeneous sub-component simulations need to be coupled and operated simultaneously in order to provide whole system simulations (“multi-disciplinary simulation/optimization”).
1-5 Interfaces to data and computational tools must provide appropriate levels of abstraction for discipline problems solving.
1-6 Resource availability and scheduling must accommodate the “bursty” / on-demand nature of the human-driven design process.
1-7 Access to resources must be predictable and independent of location or time of day.

1-8 Sufficient resources must be available so that the analyst can perform the required tasks within the context of their normal work process.

1-9 Virtual reality and immersive techniques must operate in the distributed work/resource environment.

1-10 Techniques are needed to search, interpret, and fuse multiple data archives.

1-11 Frameworks employing "software agents" are needed to provide various functions to assist analysts.

2-0 Discipline Toolbuilder Requirements

2-1 Process and workflow management techniques must provide transparent and uniform control over all distributed resources participating in problem solving environments.

2-2 Tools to provide workflow definition via "visual programming" scenarios that integrate with the analyst "desktop" environment must be available.

2-3 New approaches to computational simulation and data analysis must be accommodated in the distributed work/resource environment.

2-4 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.

2-5 Collaborative, multi-party sharing of user interfaces, data, instruments, and computation must be provided.

2-6 Techniques are needed for more transparent incorporation of interactive/steering visualization into simulations, databases, and instruments.

2-7 Techniques are needed to describe and manage diverse strategies for parameter space exploration/filling.

2-8 Tools need to automatically manage and catalogue the numerous datasets the result from parameter studies.

2-9 Consistency of numerical and data accuracy is required across distributed, heterogeneous resources.

2-10 Mechanisms for managing generalized "faults" are required for all aspects of the working environment.

2-11 Techniques are needed for coupling heterogeneous computer codes, resources, and data sources in ways so that they can work on integrated/coupled problems.

2-12 Location and architecture independent services must provide for various interprocess, interactive, data-intensive, and multi-point communication.

2-13 Techniques are needed for debugging distributed software for correctness and performance.

2-14 The methodology and implementation of incorporating, using, and managing resources in the overall environment must be scalable.

2-15 It must be possible to audit and account for use of all resources.

2-16 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).

2-17 Policy based quality-of-service should be available for all resources, including supporting construction of systems that have various "real-time" operating constraints.

2-18 General resource scheduling must provide for use based on either: what resources are available during some period of time or the combination of resources that will give a desired performance.

2-19 Systems and operations professionals must be able to manage the distributed resources.

2-20 Techniques are needed for managing evolution of the resources and services.

2-21 Resources should be immune to unauthorized access and manipulation.

2-22 Resource stakeholders/owners should have easily used mechanisms to enforce their use conditions.

2-23 The scope of the stakeholder/owner for resources must be dynamic and easily changed in order to accommodate "fluid" work groups.

2-24 The security and access control services must provide for easily specified characteristics and must be easily integrated into applications and problem solving environments.

2-25 CPU resource queuing mechanisms must accommodate:

- *specification of "when" a job or set of jobs needs to complete*
- *large numbers of "small" jobs*
- *multiple service classes - e.g. interactive, debugging, batch*
- *queue management tools including query mechanism for current state, expected queue times, etc.*
- *rich queuing interface for specification of when and how jobs should be run*
- *global queuing - overall queue management for pools of resources*
- *"birth-to-death" tracking of jobs and processes*
- *coordinating with establishing the execution environment, required dataset movement, etc.*

2-26 Event management facilities are needed which allow:

- *generation and publish events associated with job execution, data generation, etc.*
- *selective subscription to events*
- *agent and rule frameworks to collect, manipulate, and act on events*

2-27 Use of CORBA [11], Java [8], Java/RMI [9], and DCOM [2] must be provided within the context of the distributed resource environment.

2-28 Visualization techniques need to provide services that facilitate interactive, collaborative steering, allow spontaneous sharing and "floor control", and can accommodate many coordinated/related activities (e.g. jobs of a parameter study)

2-29 Distributed data management techniques are needed to support:

- *global dataset naming*
- *uniform access mechanism for archival storage systems*
- *dataset location management*
- *remote I/O*
- *"presentation" functions (e.g. number format conversion)*

2-30 Techniques are needed to provide dataset description enabling self-describing access, including: standardized metadata description, extension, and manipulation techniques, standard file formats, and data modeling techniques.

2-31 Mechanisms are needed for easily/automatically tracking dataset modification history.

2-32 Tools are needed to manage distributed heterogeneous computing architectures, including:

- *architecture independent interprocess communication mechanisms*
- *architecture based code version management*
- *code porting and architecture based transformation tools*
- *techniques to automate architecture specific optimizations*
- *compilation and compiled code management tools*
- *standard computing system run-time libraries*
- *tools for managing/establishing the local execution environment*

2-33 Generalized resource discovery services needed to provide readily available and detailed resource information, including:

- *locations, use-restrictions*
- *all functionally significant characteristics*
- *current operating state*
- *user allocations*
- *powerful and flexible resource specification language*
- *tools to query and select resources based on characteristics*

2-34 Support is needed for remote execution management, including user-level checkpoint/restart, and facilities for automatic fault detection and recovery.

2.3 Target Functionality

Analysis of the specific requirements and of the work processes of the user communities, as well as some anticipation of where the technology and problem solving needs are going in the future leads to a characterization of the desired functionality.

Problem Solving Environments, Supporting Toolkits, and High-Level Services

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

One of the most important areas is the interface to the "global shell" which supports programming rule-based workflows, potentially driven from a published/subscribed event service. Data cataloguing and data archive access, security and access control are also essential components. The PSE must also provide functionality for remote operation of laboratory / experiment / analytical instrument systems, remote visualization support, data-centric interfaces and tools and support for multi-source data exploration.

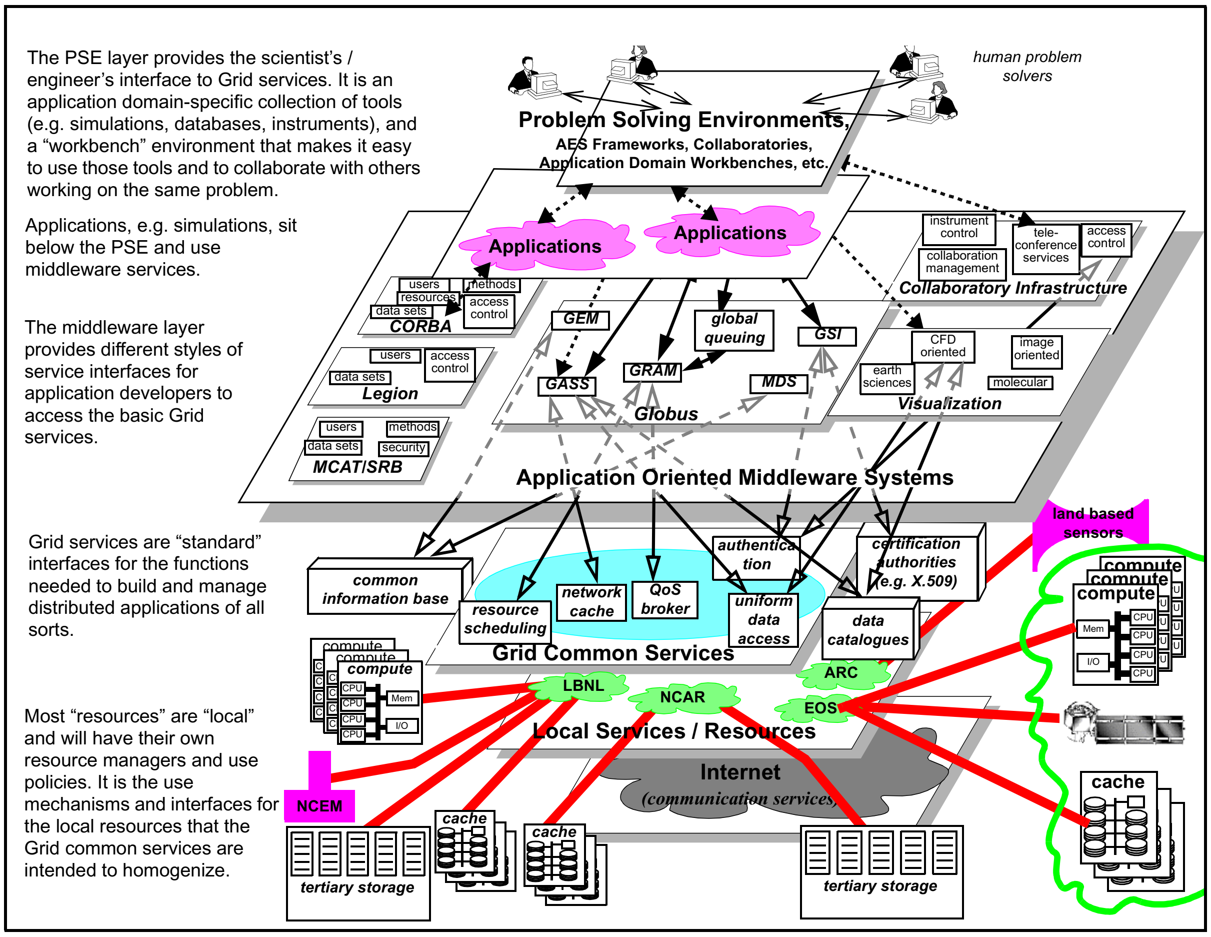
Programming Services

Tools and techniques for building applications that run in Grid environments cover a wide spectrum of programming paradigms and must operate in a multi-platform, heterogeneous computing environments. We require Globus support for Grid MPI as well as Java bindings to Globus services. CORBA, Condor [3], Java/RMI, Legion, DCOM all play a significant role in Grid programming. Compilation environment management, distributed debugging and performance analyses are difficult and important areas that must also be addressed.

Tools are needed for converting and "wrapping" legacy codes for operation in Grids and incorporating legacy Fortran codes into CORBA environments. Grid-enabled numerical solution libraries that can be optimized for distributed architectures are important, as are services such as NetSolve (http://www.cs.utk.edu/netsolve/).

Grid Common Services: Execution Management

The IPG group has identified five services that are critical to execution management. 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 queues and their user-level management tools. Workflow management and global shells is the third category. The forth category is distributed application management. The


The PSE layer provides the scientist's / engineer's interface to Grid services. It is an application domain-specific collection of tools (e.g. simulations, databases, instruments), and a "workbench" environment that makes it easy to use those tools and to collaborate with others working on the same problem.

Applications, e.g. simulations, sit below the PSE and use middleware services.

The middleware layer provides different styles of service interfaces for application developers to access the basic Grid services.

Grid services are "standard" interfaces for the functions needed to build and manage distributed applications of all sorts.

Most "resources" are "local" and will have their own resource managers and use policies. It is the use mechanisms and interfaces for the local resources that the Grid common services are intended to homogenize.

Figure 1 Grid Architecture

last category includes tools for generalized fault management including multi-level autonomous management mechanisms for system components and applications and process monitoring and supplying information to knowledge based recovery systems.

Grid Common Services: Runtime

Globus has been chosen as the initial IPG runtime system. However, other runtime services that are needed include checkpoint/restart mechanisms, access control as a Grid service, a global file system, and grid communication libraries such as a network-aware MPI that supports security, reliable multicast and remote I/O.

High-speed, wide area, distributed data management services include global naming and uniform access, uniform naming and location transparent access to resources such as data objects, computations, instruments and networks and URNs that work through Grid-wide object brokers. This, in turn requires uniform I/O mechanisms (e.g. read, write, seek) for all access protocols (e.g. http, ftp, nfs, gass...) and richer access and I/O mechanisms (e.g. "application level paging") that are present in existing systems.

Data cataloguing and publishing constitute another class of services. This includes the ability to automatically generate the meta-data about data formats and management of use conditions and access control. the ability to generate model based abstractions for data access using extended XML and XMI [12] data models is going to be very important.

Of course, high-speed, wide area, access to tertiary storage systems will always be critical. 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 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 dispersed, multi-component performance analysis as well as generalized auditing data file history and control flow tracking in distributed, multi-process simulations.

Grid Common Services: Environment Management

The key service that is used to manage the grid environment is the “Grid resource information service,” which we refer to as the “Common Information Base.” This service – currently provided by Globus MDS – 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.

Autonomous system management and fault management services provide the other aspect of the environmental services.

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 a 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 scheduling based on differentiated IP services are critical components to the co-scheduling services. In addition, tape marshaling in tertiary storage systems to support temporal reservations is essential.

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. An often overlooked service that we are also addressing involves the operational documentation and procedures that are essential to managing the Grid as a robust production service.

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 maintained in the users desktop / PSE environment(s) or on one’s person. In addition, end-to-end encrypted communication channels is needed in for many applications in order to ensure data integrity and confidentiality.

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 attributes is also a requirement.

Security and infrastructure protection are, of course, essential requirements for the resource owners.

Services for Scalability

There are a number of services and design considerations that are necessary to ensure that the Grid will scale numerically, geographically, organizationally, and functionally. The ability to broker and manage resources and handle faults autonomously is an important consideration. An additional consideration is very reliable access to “global” system state information. Finally, one must have general policy based access control and use-condition management that operates relatively automatically and has distributed management.

Services for Operability

To operate the Grid as a reliable, production environment is a challenging problem. Management tools for the Grid Common Information Base that provides global information about the configuration and state of the Grid are needed. In addition, diagnostic tools so operations/systems staff can investigate remote problems are essential. Other required services include 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.

2.4 IPG Will Identify Some “Benchmark” Applications that Validate and Test the Services

Several applications have been identified as characteristic because of the mix of services and resources that they require. These applications are being used as realistic validators of the IPG services and system. Some of the current applications include:

- Distributed multi-component NPSS system from NASA Glenn Research Center (hpcc.lerc.nasa.gov).
- Overflow, which is a NASA Ames CFD code is being tested as a widely distributed numerical simulation.
- NASA Ames’ Darwin project which involves distributed instrumentation will be an important IPG driver.

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

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

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 autonomous management systems that work in concert to provide the "service" at a given "layer."

4.0 Conclusions

The IPG as a prototype production Grid system is intended to provide NASA with a persistent ability to access, utilize and manage a significant body of distributed computing and data resources in order to facilitate application such as:

- Coupled, multidisciplinary simulations distributed across several computing systems
- Remote data analysis based control of on-line instruments
- Coupling of remote, on-line instruments to large-scale computation simulation
- Coordinated use of many dispersed data archives
- Large-scale simulations distributed across several computing systems
- Remote access to high data-rate real-time data sources / instruments and very large datasets
- Work by dispersed groups based, e.g., on central design databases (e.g. airframe or turobmachinery geometry and performance)
- On-demand data and simulation based crisis response

In addition to these mission-oriented capabilities, IPG is an experiment in using a uniform approach to distributed management of locally controlled resources.

5.0 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. and Carl Kesselman, USC/ISI) have contributed to this work. In particular, we would like to thank Bill Feiereisen, NAS Division Chief, and while the NASA HPCC Program Manager the initiator of IPG, Alex Woo, NAS Research Branch Chief, Bob Ciotti, NAS Engineering Branch Chief, and Bill Thigpen and Mary Hultquist of the NAS Division.

6.0 References

[1] The NSF PACIs are the Alliance/NCSA (http://www.ncsa.uiuc.edu/) and NPACI/SDSC (http://www.npaci.edu/).

[2] Chappell, D., *Understanding ActiveX and OLE*, Redmond, WA, Microsoft Press, 1997.

[3] Epema, D. H. J., Livny, R. van Dantzig, X. Evers, J. Pruyne, "A worldwide flock of Condors: Load sharing among workstation clusters", *Future Generation Computer Systems*, 12:53-65, 1996.

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

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

[6] 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.

[7] Gropp, W., E. Lusk, A. Skjellum, *Using MPI: Portable Parallel Programming with the Message Passing Interface*, Cambridge, MA, MIT Press, 1994.

[8] *Java Team, J. Gosling, B. Joy, G. Steele, The Java Language Specification*, Addison-Wesley, 1996, www.javasoft.com.

[9] JavaSoft, "RMI: The JDK 1.1 specification", javasoft.com/products/jdk/1.1/docs/guide/rmi/index.html, 1997.

[10] "Johnston, W., G. Jin, C. Larsen, J. Lee, G. Hoo, M. Thompson, and B. Tierney (LBNL) and J. Terdiman (Kaiser Permanente Division of Research). "Real-Time Generation and Cataloguing of Large Data-Objects in Widely Distributed Environments." Invited paper, International Journal of Digital Libraries - Special Issue on "Digital Libraries in Medicine". May, 1998. http://www-itg.lbl.gov/WALDO/

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

[12] World Wide Web Consortium, "Extensible Markup Language (XML)", http://www.w3.org/XML/