NASA’s Information Power Grid: Distributed High-Performance Computing and Large-Scale Data Management for Science and Engineering

William E. Johnston, Dennis Gannon, and Bill Nitzberg

NASA Ames Research Center, Moffett Field, CA 94035, USA

Abstract

The term “Grid” refers to distributed, high performance computing and data handling infrastructure that incorporates geographically and organizationally dispersed, heterogeneous resources that are persistent and supported. We describe a NASA project to build such a computing and data grid

Keywords: distributed, computing

1 Introduction

“Grids” (see [1]) are an approach for building dynamically constructed problem solving environments using distributed and federated, high performance computing and data handling infrastructure that incorporates geographically and organizationally dispersed resources.

The overall motivation for NASA’s Information Power Grid (“IPG”) project – a computing and data Grid – 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-scal remote instrument operation, etc.

The vision for a computing, data, and instrument Grids is that they will provide significant new capabilities to scientists and engineers by facilitating *routin* construction of information based problem solving environments.

2 An Overall Model for Grids

Analysis of some specific requirements, of the work processes of the user communities, and fo remote instrument operation, 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 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 Figure1.

Problem Solving Environments, Supporting Toolkits, and High-Level Services: A number o services directly support building and using the Grid problem solving environment. These includ the toolkits for construction of application frameworks / problem solving environments (PSE that integrate Grid services and applications into the “desktop” environment. For example, th graphical components (“widgets” / applets) for application user interfaces and control; th computer mediated, distributed human collaboration that support interface sharing and management; the tools that access the resource discovery and brokering services; tools fo generalized workflow management services such as resource scheduling, and managing high throughput jobs, etc.

An important interface for developers of Grid based applications is a “global shell,” which, in general, will support creating and managing widely distributed, rule-based workflows driven from a published/subscribed global event service. Data cataloguing and data archive access, security and access control are also essential components. The PSE must also provide access to functionality for remote operation of laboratory / experiment / analytical instrument systems, remote visualization, and data-centric interfaces and tools that support multi-source dat exploration.

Programming Services The tools and techniques that are needed for building applications that run in Grid environments must cover a wide spectrum of programming paradigms, and must operate in a multi-platform, heterogeneous computing environments. IPG, e.g., will requir Globus support for Grid MPI [2] as well as Java bindings to Globus services. CORBA, Con dor[3], Java/RMI, Legion [4], and perhaps DCOM. These are all "application oriented middleware systems" that will have to interoperate with the Grid in order to gain access to th resources managed by the Grid.

Grid Common Services – Execution Management: Discovery and brokering provides for finding 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. Execution queue management provides global views of CPU queues and their user-level management tools. Workflow management and global shells provide basic job step management. Distributed application management includes tools for generalized fault management mechanisms for applications, and for monitoring and supplying information to knowledge based recovery systems.

Grid Common Services – Runtime: Globus [5] has been chosen as the initial IPG runtime system and supplies basic services to characterize and locate resources, initiate and monitor jobs, and provide secure authentication of users. Other needed runtime services include checkpoint/restart mechanisms, access control, a global file system, and Grid communication libraries such as network-aware MPI that supports security, reliable multicast and remote I/O.

High-speed, wide area, distributed data management services should provide global naming and uniform access, uniform naming and location transparent access to resources such as data objects, computations, instruments and networks that work through Grid-wide object brokers.

Data cataloguing and publishing services should provide the ability to automatically generate th meta-data about data formats. The ability to generate model based abstractions for data access using extended XML and XMI data models is also likely to be important in the complex and dat rich environment of, e.g., aero-space design systems.

High-speed, wide area, access to tertiary storage systems is critical for the science and engineering applications that we are addressing. In IPG we are using SDSC's Meta Dat Catalogue / Storage Resource Broker ("MCAT/SRB") [6] to provide widely distributed access to tertiary storage systems, independent of the nature of the underlying mass storage system implementation. High-performance applications require high-speed access to data files, and th 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. (See, e.g.,)

Services supporting collaboration and remote instrument control are needed. 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 th Grid environment is the "Grid Information Service." This service – currently provided by Globus GIS (formerly 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.

Resource Management for Co-Scheduling and Reservation: One of the most challenging 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 fo limited periods of time. CPU advance reservation scheduling and network bandwidth advanc reservation scheduling are critical components for the co-scheduling services. In addition, tap marshaling in tertiary storage systems to support temporal reservations of tertiary storage system off line data and/or capacity is likely to be essential..

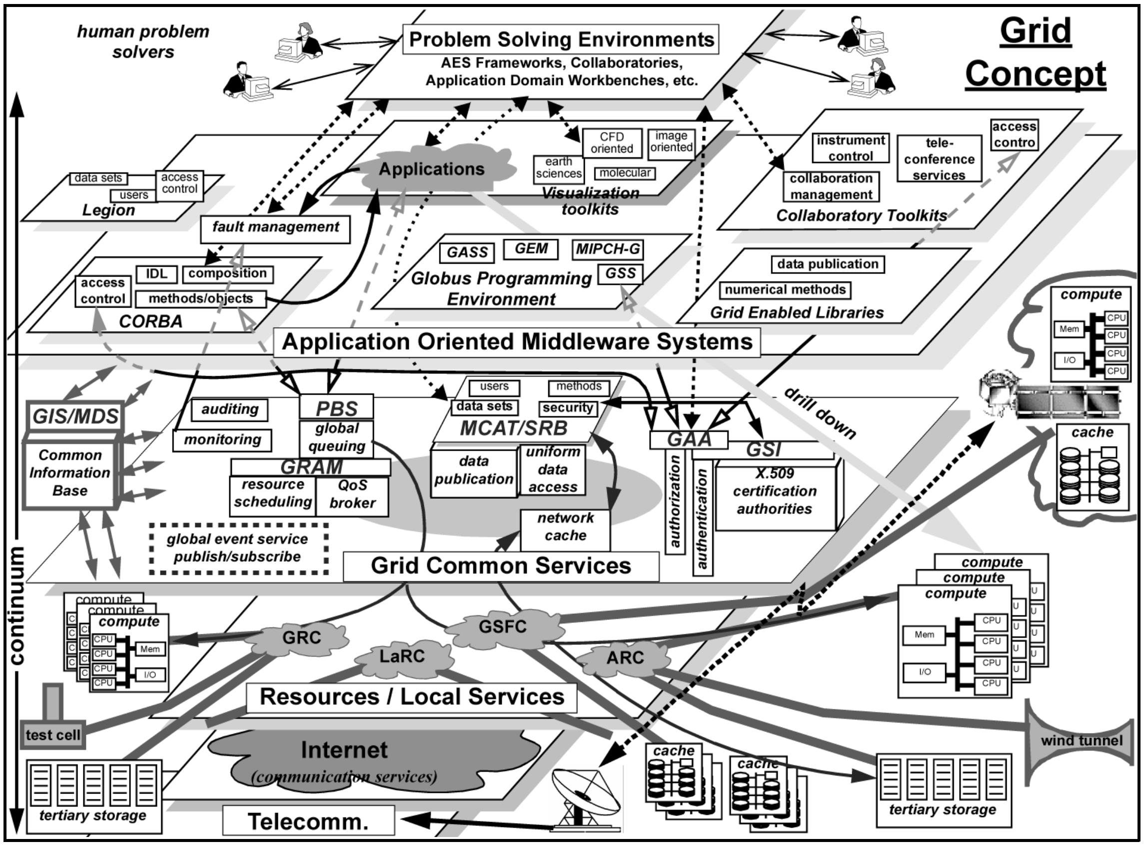

Figure 1 A Representation of Grid Architectur

Operations and System Administration: Implementing a persistent, managed Grid requires tools for diagnostic analysis and distributed performance monitoring, accounting and auditing, etc. Operational documentation and procedures 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 that are maintained in the users desktop / PSE environment(s or on one's person. This is provided by X.509 identity certificates. In addition, end-to-end encrypted communication channels are needed in for many applications in order to ensure dat integrity and confidentiality.

The second requirement is an authorization and access control model that provides fo 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.

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

We envision the Grid as a layered set of services (see Figure1) 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 o hierarchically related, independent and interdependent services, each of which performs specific function, and may rely on other Grid services to accomplish its function. Further, th "layered" model should not obscure the fact that these "layers" are not just APIs, but usually collection of functions and management systems that work in concert to provide the "service" at given "layer." The layering is not rigid, and "drill down" (e.g. code written for specific system architectures and capabilities) are easily managed by Grid services.

The arrows in the figure between several of the layers and services are intended to indicate how real application involving a team working on a computational fluid dynamics ("CFD") based design problem might interact with Grid services, top to bottom

4 The State of IPG

The first milestone in building IPG (achieved in late 1999) is a baseline system that includes:

- Globus providing the runtime system together with a global queuing system
- approximately 600 CPU nodes in half a dozen SGI Origin 2000s at four NASA sites
- several Condor managed workstation clusters
- 30-100 Terabytes of uniformly accessible mass storage managed by MCAT/SRB
- wide area network interconnects of at least 100mbit/s
- a stable and supported operational environment

Current progress is also reflected in the IPG Engineering Working Group tasks: 30+ tasks have been identified as critical for the baseline system, and groups have been organized around th major task areas.

See www.nas.nasa.gov/~wej/IPG for project information, pointers, and the IPG implementation plan. IPG is funded by NASA's Aero-Space Enterprise, Information Technology program (http://www.nas.nasa.gov/IT/overview.html).

5 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. http://www.mkp.com/books_catalog/1-55860-475-8.asp

[2] 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

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

[4] 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.

[5] 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)

[6] Moore, R., et al, "Massive Data Analysis Systems," San Diego Supercomputer Center. Se http://www.sdsc.edu/MDAS

[7] Tierney, B. Lee, J., Crowley, B., Holding, M., Hylton, J., Drake, F., "A Network-Awar Distributed Storage Cache for Data Intensive Environments", Proceeding of IEEE High Performance Distributed Computing conference (HPDC-8), August 1999. See http://www-itg.lbl.gov/DPSS/