

A Gentle Introduction to Grid Computing and Technologies

Rajkumar Buyya and Srikumar Venugopal

Grid is an infrastructure that involves the integrated and collaborative use of computers, networks, databases and scientific instruments owned and managed by multiple organizations. Grid applications often involve large amounts of data and/or computing resources that require secure resource sharing across organizational boundaries. This makes Grid application management and deployment a complex undertaking. Grid middlewares provide users with seamless computing ability and uniform access to resources in the heterogeneous Grid environment. Several software toolkits and systems have been developed, most of which are results of academic research projects, all over the world. This paper presents an introduction to Grid computing and discusses two complimentary Grid technologies: Globus developed by researchers from Argonne National Laboratory and University of Southern California, USA; and Gridbus by researchers from the University of Melbourne, Australia. Globus primarily focuses on providing core Grid services whereas Gridbus focuses on providing user-level Grid services in addition to utility computing model for management of grid resources.

Keywords: Grid computing, e-Science, Grid middleware, Gridbus, and Globus.

1 Introduction

The last decade has seen a substantial increase in commodity computing and network performance, mainly as a result of faster hardware and more sophisticated software. These commodity technologies have been used to develop low-cost high-performance computing systems, popularly called clusters, to solve resource-intensive problems in a number of application domains. Particularly, in the scientific arena, the availability of powerful computing resources has allowed scientists to broaden their simulations and experiments to take into account more parameters than ever before. Fast networks have made it possible to share data from instruments and results of experiments with collaborators around the globe almost instantaneously. Recently, research bodies have begun to launch ambitious programs that facilitate creation of such collaborations to tackle large-scale scientific problems. Collectively, such programs are termed eScience [1] to denote the pivotal role played by the computational infrastructure for enabling collaborative research. A typical eScience scenario is shown in Figure 1. eScience also envisages sharing scientific instruments such as particle accelerators

(CERN Large Hadron Collider [2]), commissioned as national/international infrastructure due to the high cost of ownership.

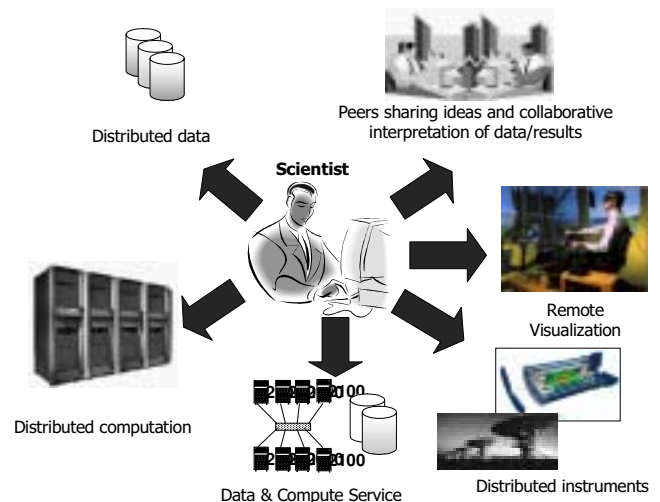


Fig. 1: A typical eScience scenario.

As a consequence of the large collaborations and the increased computational power, the data generated and analysed within eScience programs are both massive and inherently distributed.

Therefore, the challenges of such environments revolve around data – managing its access, distribution, processing and storage. These challenges thus motivate creation of a computational infrastructure by coupling wide-area distributed resources such as databases, storage servers, high-speed networks, supercomputers and clusters for solving large-scale problems, leading to what is popularly known as Grid computing [3] [4] [5]. This is analogous to the electrical power grid that provides consistent, pervasive, dependable, transparent access to electric power irrespective of its source.

As there are a large number of projects around the world working on developing Grids for different purposes at different scales, several definitions of Grid abound. The Globus Project defines Grid as “an infrastructure that enables the integrated, collaborative use of high-end computers, networks, databases, and scientific instruments owned and managed by multiple organizations.” Another utility notion based Grid definition put forward by the Gridbus Project is “Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed “autonomous” resources dynamically at runtime depending on their availability, capability, performance, cost, and users’ quality-of-service requirements.”

The development of the Grid infrastructure, both hardware and software, has become the focus of a large community of researchers and developers in both academia and industry. The major problems being addressed by Grid developments are the social problems involved in collaborative research:

- improving distributed management whilst retaining full control over locally managed resources;
- improving the availability of data and identifying problems and solutions to data access patterns; and
- providing researchers with a uniform user-friendly environment that enables access to a wider range of physically distributed facilities improving productivity.

A high-level view of activities involved within a seamless and scalable Grid environment is shown

in Figure 2. Grid resources are registered within one or more Grid information services. The end users submit their application requirements to the Grid resource broker which then discovers suitable resources by querying the Information services, schedules the application jobs for execution on these resources and then monitors their processing until they are completed. A more complex scenario would involve more requirements and therefore, Grid environments involve services such as security, information, directory, resource allocation, application development, execution management, resource aggregation, and scheduling. Software tools and services providing these capabilities to link computing capability and data sources in order to support distributed analysis and collaboration are collectively known as Grid middleware.

In order to provide users with a seamless computing environment, the Grid middleware systems need to solve several challenges originating from the inherent features of the Grid [6]. One of the main challenges is the heterogeneity that results from the vast range of technologies, both software and hardware, encompassed by the Grid. Another challenge involves the handling of Grid resources that are spread across political and geographical boundaries and are under the administrative control of different organizations. It follows that the availability and performance of Grid resources are unpredictable as requests from within an administrative domain may gain more priority over requests from outside. Thus, the dynamic nature of Grid environment poses yet another challenge.

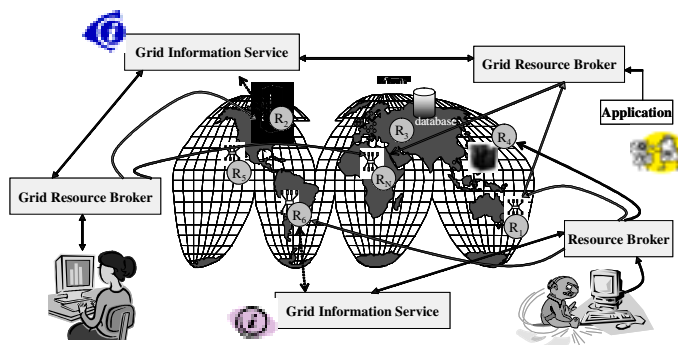


Fig. 2: A world-wide Grid computing environment.

To tackle these challenges, a Grid architecture has been proposed based on the creation of Virtual Organizations (VOs)

[7] by different physical (real-world) organizations coming together to share resources and collaborating in order to achieve a common goal. A VO defines the resources available for the participants and the rules for accessing and using the resources. Within a VO, participants belonging to member organizations are allocated resource share based on urgency and priority of a request as determined by the objectives of the VO. Another complimentary Grid architecture [8] is based on economic principles in which resource providers (owners) compete to provide the best service to resource consumers (users) who select appropriate resources based on their specific requirements, the price of the resources and their expectations of Quality-of-Service (QoS) from the providers. Two examples of QoS terms are the deadline by which the resource needs to be available and the maximum price (budget) that can be paid by the user for the service. QoS terms are enforced via SLAs (Service Level Agreements) between the providers and the consumers, the violation of which results in penalties.

2 Grid Components

In a world-wide Grid environment, capabilities that the infrastructure needs to support include:

- Remote storage and/or replication of data sets
- Publication of datasets using global logical name and attributes in the catalogue
- Security –access authorisation and uniform authentication
- Uniform access to remote resources (data and computational resources)
- Publication of services and access cost
- Composition of distributed applications using diverse software components including legacy programs.
- Discovery of suitable datasets by their global logical names or attributes
- Discovery of suitable computational resources
- Mapping and Scheduling of jobs (Aggregation of distributed services)
- Submission, monitoring, steering of jobs execution
- Movement of code/data between the user desktop machines and distributed resources
- Enforcement of quality of service requirements
- Metering and Accounting of resource usage

The above capabilities in Grid computing environments play a significant role in enabling a variety of scientific, engineering, and business applications. Various Grid components providing the above capabilities are arranged into layers. Each layer builds on the services offered by the lower layer in addition to interacting and co-operating with components at the same level (e.g., Resource broker invoking secure process management services provided by core middleware). Figure 3 shows the hardware and software stack within a typical Grid architecture. It consists of four layers: fabric, core middleware, user-level middleware, and applications and portals layers.

Grid Fabric layer consists of distributed resources such as computers, networks, storage devices and scientific instruments. The computational resources represent multiple architectures such as clusters, supercomputers, servers and ordinary PCs which run a variety of operating systems (such as UNIX variants or Windows). Scientific instruments such as telescope and sensor networks provide real-time data that can be transmitted directly to computational sites or are stored in a database.

Core Grid middleware offers services such as remote process management, co-allocation of resources, storage access, information registration and discovery, security, and aspects of Quality of Service (QoS) such as resource reservation and trading. These services abstract the complexity and heterogeneity of the fabric level by providing a consistent method for accessing distributed resources.

User-level Grid middleware utilizes the interfaces provided by the low-level middleware to provide higher level abstractions and services. These include application development environments, programming tools and resource brokers for managing resources and scheduling application tasks for execution on global resources.

Grid applications and portals are typically developed using Grid-enabled programming environments and interfaces and brokering and scheduling services provided by user-level middleware. An example application, such as parameter simulation or a grand-challenge problem, would require computational power, access to remote datasets, and may need to interact with scientific instruments. Grid portals offer Web-enabled application services, where users can submit and collect results for their jobs on remote resources through the Web.

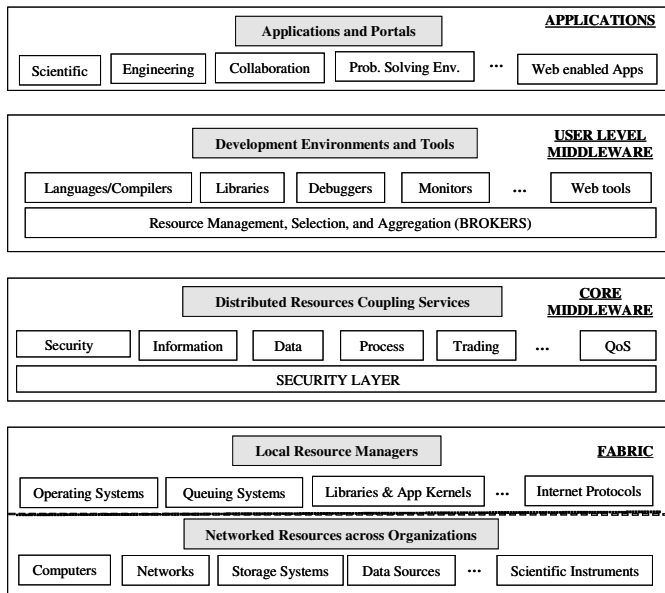


Fig. 3: A Layered Grid Architecture and components.

2.1 Operational Flow from Users' Perspective

Figure 4 shows Grid environment from an operations perspective with the components organized according to their deployment and functionality. To make resources constituents of the Grid, they need to be accessible from different management domains. This can be achieved by installing core Grid middleware such as Globus in Unix/Linux environment and Alchemi in Windows environment. Multi-node resources such as clusters need to be presented as a single resource to the Grid and this can be achieved by deploying job management systems such as Condor, PBS, or Sun Grid Engine on them. In a Grid environment where data needs to be federated for sharing among various interested parties, data grid technologies such as SRB, Globus RLS, or EU Data Grid needs to be deployed. The user-level middleware need to be deployed on resources responsible for providing resource brokering and application deployment services. The users may even access these services via Web portals.

In eScience Grid environments such as those established for LHC (Large Hadron Collider) data sharing and analysis, the key steps involved and the interaction between various grid components are as follows (see Figure 4):

1. The users compose their application as a distributed application (e.g., parameter sweep) using visual application development tools.
2. The users specify their analysis and quality-of-service requirements and submit them to the Grid resource broker.
3. The Grid resource broker performs resource discovery and their characteristics using the Grid information service.
4. The broker identifies resource service prices by querying the Grid market directory.
5. The broker identifies the list of data sources or replicas and selects the optimal ones.
6. The broker also identifies the list of computational resources that provides the required application services.
7. The broker ensures that the user has necessary credit or authorized share to utilise resources.
8. The broker scheduler maps and deploys data analysis jobs on resources that meet user quality-of-service requirements.
9. The broker agent on a resource executes the job and returns results.
10. The broker collates the results and passes to the user.
11. The metering system charges the user by passing the resource usage information to the accounting system.

The accounting system reports resource share allocation or credit utilisation to the user.

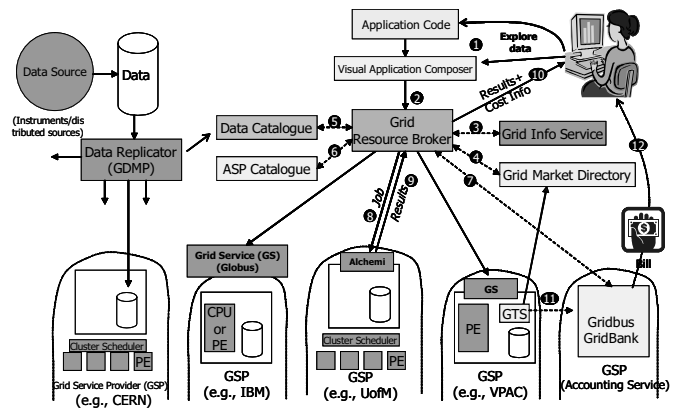


Fig. 4: Grid technologies for analysis of distributed data.

3 Grid Initiatives World-Wide

Given the possibilities of Grid computing, it is no surprise that there is a huge interest in this technology around the world. Currently, Grid projects that have been initiated around the world can be broadly classified into two classes: (1) Grid infrastructure development that involves setting up hardware, software and administrative mechanisms so that application scientists can make use of these facilities for their research and (2) Grid middleware research that investigates on the development of software and policy mechanisms that assist in realizing the full potential of Grid computing. Many of these projects are motivated by large-scale scientific projects that will involve production and analysis of data at an unprecedented scale. Often cited such large-scale scientific projects are the LHC experiments (CMS, ATLAS, Alice, LHCb) at CERN when they will begin data production in 2007. The volume of data that is expected to be generated by these experiments is expected to be in petabytes (PB) which need to be distributed to physicists around the world for analysis. As the Grid has been mandated as the IT infrastructure for handling the massive workloads of the LHC experiments, all the collaborating nations are setting up Grid infrastructure in one form or another. In the following sections, we will state some of the major Grid infrastructure and middleware projects around the world.

3.1 United States

Production Grid testbeds for various application domains have been deployed over physical (hardware) Grid infrastructure such as the US NSF (National Science Foundation)-funded TeraGrid that provides over 40 teraflops of computing power at eight sites around the US with 2 PB of available storage interconnected by 10-30 gigabits per second network. One of the major testbeds is Grid3 which covers 25 sites across US and Korea that are used collectively for application domains such as High-Energy Physics, Astronomy and Biology. Other Grid projects setting up testbeds include GriPhyN (Grid Physics Network), PPDG (Particle Physics Data Grid), LIGO (Laser Interferometer Gravitational-Wave Observatory) and SDSS (Sloan Digital Sky Survey). The BioInformatics Research Network (BIRN) is another testbed for the purpose of furthering biomedical science by sharing data stored in

different repositories around US. NEESGrid enables scientists in the earthquake engineering community to carry out experiments in distributed locations and analyse data through a uniform interface.

Out of the Grid middleware efforts in the United States, the Globus Toolkit from the Globus Alliance led by Argonne National Laboratory is the most widely known. We will be discussing this in detail later in this article. SRB (Storage Resource Broker) from San Diego Supercomputing Centre is another widely deployed middleware for federating disparate repositories and takes care of data storage, replication and access management. Other notable efforts are the Condor project (University of Wisconsin, Madison) for high throughput computing mechanisms, AppLeS (University of California, San Diego) for application scheduling, GrADS project headed by Rice University for application development environment, NetSolve (University of Tennessee, Knoxville) which develops libraries for Grid RPC (Remote Procedure Call) applications, and the Virtual Data Grid (VDG) project that is looking into reuse of data products for future executions. In addition, several commercial organizations such as IBM, Sun, HP, and Oracle are actively involved in the development of enterprise and global utility Grid technologies. [An article by Gentzsch in this special issue presents commercial Grid efforts in detail.]

3.2 Europe

The two major Grid efforts in Europe, started in early 2001, are the United Kingdom (UK)'s e-Science program and the European Union funded Data Grid project which is now succeeded by the EGEE (Enabling Grids for E-sciencE) project. The UK e-Science program proposed to focus on the promotion and advancement of UK involvement in the Grid through infrastructure provision, and development of Grid middleware. EGEE focuses on developing a Grid infrastructure available to scientists and to develop robust middleware for application deployment.

CERN, the European Organization for Nuclear Research, and the High-Energy Physics (HEP) community have established an International Data Grid project with intent to apply the work to other scientific communities such as Earth Observation and Bioinformatics. The project objectives are to establish a research network for data Grid

technology development, demonstrate data Grid effectiveness through the large-scale real world deployment of end-to-end application experiments, and to demonstrate the ability to use low-cost commodity components to build, connect, and manage large general-purpose, data intensive computer clusters.

Other notable EU funded projects include GridLab providing application development toolkit, Cactus framework for scientific programming, GridSphere for creating a web portal environment for Grid users, P-Grade providing visual environment for application development, and Triana for workflow formulation. OGSA-DAI, part of UK e-Science program, supports integration of relational databases in Grid environments.

3.3 Asia-Pacific

Several countries in the Asia-Pacific region have started national Grid programs similar to those initiated in United States and Europe. In addition, countries such as Australia, China, Japan, South Korea and Singapore are active participants in worldwide Grid projects such as the LHC Computational Grid (LCG) and the Virtual Observatory projects. Some of the notable Grid programs are the NAREGI (National Research Grid Initiative) in Japan, APAC Grid (Australia), China National Grid (China), K*Grid (South Korea) and APGrid (Asia Pacific Grid).

Prominent Grid middleware projects include Ninf project (Tokyo Institute of Technology) for building a Grid-based RPC (remote procedure call) system, the Grid Datafarm (Gfarm) project (AIST, Japan) for providing a petascale data storage and processing system, the Nimrod project (Monash University, Australia) for parametric computations on Grid resources and the Gridbus project (University of Melbourne, Australia) for service-oriented utility computing. We will look at the Gridbus project in detail later in this article.

3.4 India

Centre for Development of Advanced Computing is setting up Grid infrastructure by connecting supercomputers such as PARAM 10000 by high-speed networks. The Indian Institute of Information Technology and Management, Kerala is heading a Grid project that aims to bring educational materials

closer to educators wherever they are located. Known as Kerala Education Grid, this project envisages the linking of colleges and universities to resource centres that will supply education materials on demand and increase cooperation and networking among the affiliated academics. [see an article by Sherly in this special issue for details on Kerala Education Grid.]

Information Technology companies in India have shown great interest in the potential of Grid technology for obtaining the maximum benefits out of enterprise infrastructure. Corporations such as Infosys and Satyam have already started initial forays into this area.

3.5 Standardization Efforts

Given the large amount of middleware development happening in this area of research, standardization is important to ensure interoperability between different products and implementations. In the past few years, Grid standardization efforts led by the Global Grid Forum (GGF) [9] have produced standards for almost all aspects of Grid technology. Work at the GGF have produced the Open Grid Service Infrastructure (OGSI) specification and its successor, the Web Services Resource Framework (WSRF) that pave the way for integration of Web Services within Grid architecture. This is important as Web Services allow Grid developers to take advantage of standard message formats and communications mechanisms such as HTTP and XML for communicating between heterogeneous components and architectures. Other standardization bodies such as W3C (World Wide Web Consortium), OASIS (Organization for Advancement of Structured Information Standards) and IETF (Internet Engineering Task Force) also produce standards relevant to aspects of Grid Computing.

In the following sections, we will take a look at two different middleware projects – the Globus Toolkit [9] and the Gridbus Project [11]. The Globus toolkit is widely used and provides the functionalities of core Grid middleware as mentioned above. The Gridbus Project, while extending across the Grid middleware stack, concentrates more on user-level middleware with special emphasis on economy based Grid resource management guided by user-defined QoS attributes.

4 Globus Toolkit

The Globus project provides open source software toolkit that can be used to build computational grids and grid based applications. It allows sharing of computing power, databases, and other resources securely across corporate, institutional and geographic boundaries without sacrificing local autonomy. The core services, interfaces and protocols in the Globus toolkit allow users to access remote resources seamlessly while simultaneously preserving local control over who can use resources and when. The Globus architecture, shown in Figure 5, has three main groups of services accessible through a security layer. These groups are Resource Management, Data management and Information Services.

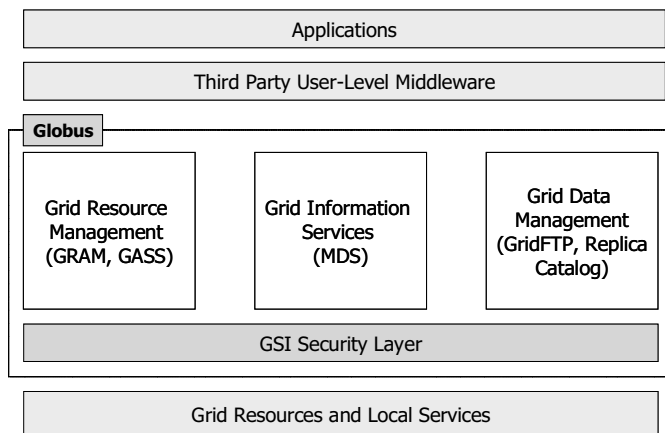


Fig. 5: The Globus Architecture.

The local services layer contains the operating system services, network services like TCP/IP, cluster scheduling services provided by Load Leveler, job-submission, query of queues, and so on. The higher layers of the Globus model enable the integration of multiple or heterogeneous clusters. The core services layer contains the Globus toolkit building blocks for security, job submission, data management, and resource information management. The high-level services and tools layer contains tools that integrate the lower level services or implement missing functionality.

4.1 GSI Security Layer:

The Grid Security Infrastructure (GSI) provides methods for authentication of Grid users and secure

communication. It is based on SSL (Secure Sockets Layer), PKI (Public Key Infrastructure) and X.509 Certificate Architecture. The GSI provides services, protocols and libraries to achieve the following aims for Grid security:

- Single sign-on for using Grid services through user certificates
- Resource authentication through host certificates
- Data encryption
- Authorization
- Delegation of authority and trust through proxies and certificate chain of trust for certificate authorities

Users gain access to resources by having their Grid certificate subjects mapped to an account on the remote machine by its system administrators. This also requires that the CA that signed the user certificate be trusted by the remote system. Access permissions have to be enforced in the traditional UNIX manner through restrictions on the remote user account.

CAs (Certificate Authorities) are also a part of realising the notion of VOs. A user who has a certificate signed by the CA of the VO gains access to the resources authenticated by the same CA. VOs can cooperate between themselves by recognizing each others CAs so that users can access resources between collaborations. These mechanisms are used in many Grid testbeds. Depending on the structure of the testbed and the tools used, the users may gain access automatically to the resources or may have to contact the system administrators individually to ensure access.

Most services require mutual authentication before carrying out their functions. This guarantees non-repudiability and data security on both sides. However, the current state of GSI tools makes it more likely that some users may share the usage of a single certificate to gain access to higher number of resources or that they may be mapped to the same account on the remote machine. This may raise serious questions on the authenticated users and the confidentiality of user data on the remote machine. Production testbeds have policies in place to restrict this behaviour but there is still some way to go before these are restricted at the middleware level.

4.2 Resource Management:

The resource management package enables resource allocation through job submission, staging of executable files, job monitoring and result gathering. The components of Globus within this package are:

Globus Resource Allocation Manager (GRAM): GRAM provides remote execution capability and reports status for the course of the execution. A client requests a job submission to the gatekeeper daemon on the remote host. The gatekeeper daemon checks if the client is authorized (i.e., the client certificate is in order and there is a mapping of the certificate subject to any account on the system). Once authentication is over, the gatekeeper starts a job manager that initiates and monitors the job execution. Job managers are created depending on the local scheduler on that system. GRAM interfaces to various local schedulers such as Portable Batch System (PBS), Load Sharing Facility (LSF) and LoadLeveler.

The job details are specified through the Globus Resource Specification Language (RSL), which is a part of GRAM. RSL provides syntax consisting of attribute-value pairs for describing resources required for a job including the minimum memory and the number of CPUs.

Globus Access to Secondary Storage (GASS): GASS is a file-access mechanism that allows applications to pre-fetch and open remote files and write them back. GASS is used for staging-in input files and executables for a job and for retrieving output once it is done. It is also used to access the standard output and error streams of the job. GASS uses secure HTTP based streams to channel the data and has GSI-enabled functions to enforce access permissions for both data and storage.

4.3 Information Services:

The information services package provides static and dynamic properties of the nodes that are connected to the Grid. The Globus component within this package is called Monitoring and Discovery Service (MDS).

MDS provides support for publishing and querying of resource information. Within MDS, schema define classes that represent various properties of the system. MDS has a three-tier structure at the bottom of which are Information Providers (IPs) that gather data about resource

properties and status and translate them into the format defined by the object classes. The Grid Resource Information Service (GRIS) forms the second tier and is a daemon that runs on a single resource. GRIS responds to queries about the resource properties and updates its cache at intervals defined by the time-to-live by querying the relevant IPs. At the topmost level, the GIIS (Grid Information Index Service) indexes the resource information provided by other GRISs and GIISs that are registered with it.

4.4 Data Management:

The data management package provides utilities and libraries for transmitting, storing and managing massive data sets that are part and parcel of many scientific computing applications. The elements of this package are:

GridFTP: It is an extension of the standard FTP protocol that provides secure, efficient and reliable data movements in grid environments. In addition to standard FTP functions, GridFTP provides GSI support for authenticated data transfer, third-party transfer invocation and striped, parallel and partial data transfer support.

Replica Location and Management: This component supports multiple locations for the same file throughout the grid. Using the replica management functions, a file can be registered with the Replica Location Service (RLS) and its replicas can be created and deleted. Within RLS, a file is identified by its Logical File Name (LFN) and is registered within a logical collection. The record for a file points to its physical locations. This information is available from the RLS upon querying.

The major Grid tools and application projects making use of Globus as their low-level middleware include: AppLeS, Ninf, Nimrod-G, NASA IPG, Condor-G, Gridbus Broker, UK eScience Project, GriPhyN, and EU Data Grid.

5 Gridbus Middleware

Grid technologies such as Globus provide capabilities and services required for secure access and execution of a job on a resource in a uniform manner on heterogeneous resources. However, to achieve the complete vision of Grid as a utility computing environment, a number of challenges need to be addressed. They include composition of analysis software as a distributed application, resource brokering methodologies and strategies for scheduling data-driven applications, data Grid

economy for data management, application service specification, and accounting of resource consumption. The application development and deployment services need to scale from desktop environment to global Grids and need to support both scientific and business applications.

The Gridbus project is engaged in the design and development of service-oriented cluster and Grid middleware technologies to support eScience and eBusiness applications. It extensively leverages related software technologies and provides an abstraction layer to hide idiosyncrasies of heterogeneous resources and low-level middleware technologies from application developers. In addition, it extensively focuses on realization of utility computing model scaling from clusters to Grids and to peer-to-peer computing systems. It uses economic models in efficient management of shared resources and promotes commoditization of their services. Thus, it enhances the tradability of Grid services and manages efficiently the supply and demand for resources.

Gridbus supports commoditization of Grid services at various levels:

- Raw resource level (e.g., selling CPU cycles and storage resources)
- Application level (e.g., molecular docking operations for drug design application)
- Aggregated services (e.g., brokering and reselling of services across multiple domains)

The idea of a computational economy helps in creating a service-oriented computing architecture where service providers offer paid services associated with a particular application and users, based on their requirements, would optimize by selecting the services they require and can afford within their budget. Gridbus emphasizes the end-to-end quality-of-services driven by computational economy at various levels - clusters, peer-to-peer (P2P) networks, and the Grid - for the management of distributed computational, data, and application services.

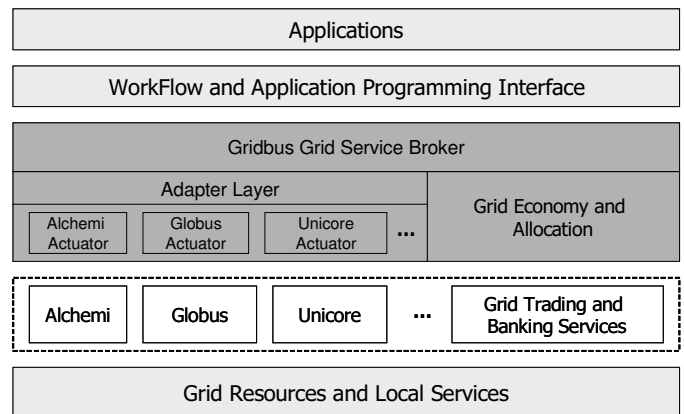


Fig. 6: The Gridbus Architecture.

Figure 6 shows a layered architecture depicting the Gridbus components in conjunction with other middleware technologies such as Globus that have been discussed before. Gridbus provides software technologies that spread across the following categories:

- Enterprise Grid Middleware (Alchemi)
- Service-Level Agreements based Allocation of Cluster Resources (Libra)
- Grid Economy and Virtual Enterprise (Grid Market Directory, Compute Power Market)
- Grid Trading and Accounting Services (GridBank)
- Grid Resource Brokering and Scheduling (Gridbus Broker)
- Grid Workflow Management (Gridbus Workflow Engine)
- Grid Application Programming Interface (Visual Parametric Modeller)
- Grid Portals (GMonitor, Gridscape)
- Grid Modeling and Simulation (GridSim)

Alchemi: Though scientific computing facilities have been heavy users of Unix-class operating systems, the vast majority of computing infrastructure within enterprises is still based on Microsoft Windows. Alchemi

was developed to address the need within enterprises for a desktop grid solution that utilizes the unused computational capacity represented by the vast number of PCs and workstation running Windows within an organization. Alchemi is implemented on top of the Microsoft .NET Framework and provides the runtime machinery for constructing and managing desktop Grids. It also provides an object-oriented programming model along with web service interfaces that enable its services to be accessed from any programming environment that supports SOAP-XML abstraction.

Libra: Libra is a cluster scheduling system that performs service-level agreements based allocation of resources. It guarantees a certain share of the system resources to a user job such that the job is completed by the deadline specified by the user provided he has the requisite budget for it. Jobs whose output is required immediately require a higher budget than those with a more relaxed deadline. Thus, Libra delivers utility value to the cluster users and increases their satisfaction by creating realistic expectations for the job turnaround times.

Grid Market Directory (GMD): It provides registry service where service providers can register themselves and publish the services they're providing and consumers can query to obtain the service that meets their requirements. Some of the attributes of a service are its access point, input mechanism and the cost involved in using it.

Compute Power Market (CPM): It is a market-based resource management and scheduling system developed over the JXTA platform. It enables trading of idle computational power over P2P networks. The CPM components that represent markets, consumers and providers are Market Server, Market Resource Agent, and Market Resource Broker (MRB). It supports various economic models for resource trading and matching service consumers and providers and allows plugging in of different scheduling mechanisms.

Accounting and Trading Services: GridBank is a Grid-wide accounting and micro-payment service that provides a secure infrastructure for Grid Service Consumers (GSCs) to pay Grid Service Providers (GSPs) for the usage of their services. The consumer is charged on the basis of resource usage records maintained by the provider and service charges that have been agreed upon by both parties in the beginning. GridBank can also be used as an authentication and authorization mechanism thereby ensuring access to the resources to

only those consumers with the requisite credit in their accounts.

Resource Broker: The Gridbus Resource Broker provides an abstraction to the complexity of Grids by ensuring transparent access to computational and data resources for executing a job on a Grid. It uses user requirements to create a set of jobs, discover resources, schedule, execute and monitor the jobs and retrieve their output once they are finished. The broker supports a declarative and dynamic parametric programming model for creating Grid applications.

The Gridbus broker has the capability to locate and retrieve the required data from multiple data sources and to redirect the output to storage where it can be retrieved by processes downstream. It has the ability to select the best data repositories from multiple sites based on availability of files and quality of data transfer.

Web Portals: G-monitor is a web-portal for monitoring and steering computations on global Grids. G-monitor interfaces with resource brokers such as Gridbus broker and Nimrod-G and uses their services to initiate and monitor application execution. It provides the user with up-to-date information about the progress of the execution at the individual job level and at the overall experiment level. At the end of the execution, the user can collect the output files through G-monitor.

Gridscape: It is tool for creation and generation of interactive and dynamic portals that enable users to view the status of the resources within the testbed and easily add new resources when required. It is also possible to customize the portal to reflect the unique identity of the organization managing the testbed.

Modeling and Simulation: The GridSim toolkit provides facilities for the modeling and simulation of resources and network connectivity with different capabilities, configurations and domains. It supports primitives for application composition, information services for resource discovery and interfaces for assigning application tasks to resources and managing their execution. It also provides a visual modeler interface for creating users and resources. These features can be used to simulate parallel and distributed scheduling systems such as resource brokers or Grid schedulers for evaluating performance of scheduling algorithms or heuristics.

Gridbus technology has been used in creating a range of e-Science and e-Business applications from reading

brainwaves and early detection of breast cancer to searching for cosmic particles and developing finance portfolio analysis. Some prominent usages include: (1) Osaka University, Japan has developed a Gridbus-based neuroscience application that helps in Brain Activity Analysis, (2) Friedrich Miescher Institute for Biomedical Research, Switzerland has used Gridbus technology in their distributed bioinformatics platform that helps in identification of patterns of transcription factors in the regulatory regions of mammalian genes, (3) Tier Technologies, USA has used in OCR data processing, and (4) CSIRO Land and Water Division and CRC for Catchment Hydrology in Australia have used our Gridbus technology in distributed processing of environmental simulation models.

6 Summary and Conclusion

To summarize, we have shown how Grid computing is becoming the preferred platform for next generation eScience experiments that require management of massive distributed data. We have covered some of the major Grid efforts around the world and discussed the Grid software stack with two sample technologies – the Globus Toolkit and the Gridbus middleware. A number of scientific and commercial applications have started harnessing Grids. It can be observed that while there has been a lot of development for Grid technologies for eScience, there is still more to be achieved in terms of Grids providing computing utilities in the same manner as power utilities supply electric power. Ultimately, this would require development of richer

services and applications on top of already existing ones so that Grid computing can move beyond scientific applications and into mainstream IT infrastructure.

References

- [1] T. Hey and A. E. Trefethen, "The UK e-Science Core Programme and the Grid," *Journal of Future Generation Computer Systems (FGCS)*, vol. 18, no. 8, pp. 1017-1031, 2002.
- [2] CERN LHC Computing Grid Project, <http://lhcgird.web.cern.ch/LHCgrid/>
- [3] I. Foster and C. Kesselman (editors), *The Grid: Blueprint for a Future Computing Infrastructure*, Morgan Kaufmann Publishers, San Francisco, USA, 1999.
- [4] F. Berman, G. Fox, A. Hey (editors), *Grid Computing: Making the Global Infrastructure a Reality*, Wiley Press, New York, USA, 2003.
- [5] M. Parashar and C. Lee (editors), *Proceedings of the IEEE : Special Issue on Grid Computing*, Volume 93, Issue 3, IEEE Press, New York, USA, March 2005.
- [6] M. Baker, R. Buyya, and D. Laforenza, Grids and Grid Technologies for Wide-Area Distributed Computing, *International Journal of Software: Practice and Experience (SPE)*, Volume 32, Issue 15, pp. 1437-1466, Wiley Press, USA, December 2002.
- [7] I. Foster, C. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, *International Journal of High Performance Computing Applications*, vol. 15, pp. 200-222, Sage Publishers, London, UK, 2001.
- [8] R. Buyya, D. Abramson, and J. Giddy, An Economy Driven Resource Management Architecture for Global Computational Power Grids, *The 2000 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2000)*, Las Vegas, USA, June 26-29, 2000.
- [9] Global Grid Forum, <http://www.ggf.org>
- [10] The Globus Project, <http://www.globus.org>
- [11] The Gridbus Project, <http://www.gridbus.org>

Dr. Rajkumar Buyya is a Senior Lecturer and the Storage Tek Fellow of Grid Computing in the Department of Computer Science and Software Engineering at the University of Melbourne, Australia. He is also serving as the Director of the Grid Computing and Distributed Systems Laboratory. He has authored/co-authored over 100 papers and technical documents that include three books—Microprocessor x86 Programming, Mastering C++, and Design of PARAS Microkernel. He received B.E, M.E, and Ph.D. degrees from Mysore, Bangalore, and Monash Universities respectively. He was awarded Dharma Ratnakara Memorial Trust Gold Medal for academic excellence in Mysore University. He is currently serving as Co-Chair of the IEEE Technical Committee on Scalable Computing and Associate Editor of the Journal of Future Generation Computing Systems, Elsevier Press, Holland.

Mr. Srikumar Venugopal is a Doctoral Candidate in the Department of Computer Science and Software Engineering, University of Melbourne, Australia. He is also a research assistant in the Grid Computing and Distributed Systems Laboratory. He is the lead researcher and developer of the Gridbus Grid Service Broker and has assisted in its usage in grid-enabling several scientific applications. His research interests include grid economy, data grids, scheduling and mobile agents. He received B.E degree from Cochin University of Science and Technology.