IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-2, Issue-11 E-ISSN: 2347-2693

Architecture and Layers in Grid Computing

V.Priya[1*], K.Yamunadevi[2] and P.Priyanga[3]

[1*]*Department of Computer Science and Applications*
[2,3]*Department of Computer Science*
Vivekananda College of Arts and Sciences for Women, Namakkal, TamilNadu, India

www.ijcaonline.org

Received: Oct/26/2014 Revised: Nov/08/2014 Accepted: Nov/22/2014 Published: Nov/30/2014

Abstract - The perfect architecture of Grid Computing System, analyzes security requirements. Some security problems existing in Grid Computing System, presents five-layer security architecture, defines a new set of security policy and gives the representation, introduces future work. In various ways we are trying to explain grid computing along with its architecture, infrastructure and the standards available for grid computing. Then at last we have discussed about the earlier and current activities in grid computing.

Keywords: Grid Computing, Grid Architecture, Application Layer, Open Grid Service Architecture (OGSA). Grid Application, Grid Infrastructure

I. INTRODUCTION

GRID computing is a technology for coordinating large scale resource sharing and problem solving among various autonomous group. Grid technologies are currently distinct from other major technical trends such as internet, enterprise distributed networks and peer to peer computing. Also it has some embracing issues in QoS, data management, scheduling, resource allocation, accounting and performance.

Grids are built by various user communities to offer a good infrastructure which helps the members to solve their specific problems which are called a grand challenge problem .A grid consists of different types of resources owned by different and typically independent organizations which results in heterogeneity of resources and policies. Because of this, grid based services and applications experience a different resource behavior than expected.

Similarly, a distributed infrastructure with ambitious service put more impact on the capabilities of the interconnecting networks than other environments. Grid High Performance Network works on network research, grid infrastructure and development.

Grid computing can mean different things to different individuals. The grand vision is often presented as an analogy to power grids where users (or electrical appliances) get access to electricity through wall sockets with no care or consideration for where or how the electricity is actually generated.

In this view of grid computing, computing becomes pervasive and individual users (or client applications) gain access to computing resources (processors, storage, data, applications, and so on) as needed with little or no knowledge of where those resources are located or what the underlying technologies, hardware, operating system, and so on.

Grid computing could be defined as any of a variety of levels of virtualization along a continuum. Exactly where along that continuum one might say that a particular solution is an implementation of grid computing versus a relatively simple implementation using virtual resource is a matter of opinion. But even at the simplest levels of virtualization, one could say that grid-enabling technologies.

II. GRID COMPUTI

Grid computing is the collection of computer resources from multiple locations to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. Grid computing is distinguished from conventional high performance computing systems such as cluster computing in that grid computers have each node set to perform a different task/application. Grid computers also tend to be more heterogeneous and geographically dispersed (thus not physically coupled) than cluster computers. Although a single grid can be dedicated to a particular application, commonly a grid is used for a variety of purposes. Grids are often constructed with general-purpose grid middleware software libraries.

Grid size varies a considerable amount. Grids are a form of distributed computing whereby a “super virtual computer” is composed of many networked loosely coupled computers acting together to perform large tasks. For certain applications, “distributed” or “grid” computing, can be seen as a special type of parallel computing that relies on complete computers (with onboard CPUs, storage, power

Corresponding Author: *V.Priya*

supplies, network interfaces, etc.) connected to a network (private, public or the Internet) by a conventional network interface, such as Ethernet. This is in contrast to the traditional notion of a supercomputer, which has many processors connected by a local high-speed computer bus. Grid computing combines computers from multiple administrative domains to reach **a common goal**, **to solve a single task**, and may then disappear just as quickly.

Grids are a form of distributed computing whereby a "super virtual computer" is composed of many networked loosely coupled computers acting together to perform very large tasks. This technology has been applied to computationally intensive scientific, mathematical, and academic problems through volunteer computing, and it is used in commercial enterprises for such diverse applications as drug discovery, economic forecasting, seismic analysis, and back office data processing in support for e-commerce and Web services.

III. EARLY GRID ACTIVITIES

Over the past several years, there has been a lot of interest in computational Grid Computing worldwide. We also note a number of derivatives of Grid Computing, including compute grids, data grids, science grids, access grids, knowledge grids, cluster grids, terra grids, and commodity grids. As we explore careful examination of these grids, we can see that they all share some form of resources; however, these grids may have differing architectures.

One key value of a grid, whether it is a commodity utility grid or a computational grid, is often evaluated based on its business merits and the respective user satisfaction. User satisfaction is measured based on the QoS provided by the grid, such as the availability, performance, simplicity of access, management aspects, business values, and flexibility in pricing. The business merits most often relate to and indicate the problem being solved by the grid. For instance, it can be job executions, management aspects, simulation workflows, and other key technology-based foundations.

Earlier Grid Computing efforts were aligned with the overlapping functional areas of data, computation, and their respective access mechanisms. Let us further explore the details of these areas to better understand their utilization and functional requirements.

Data

The data aspects of any Grid Computing environment must be able to effectively manage all aspects of data, including data location, data transfer, data access, and critical aspects of security. The core functional data requirements for Grid Computing applications are:

- The ability to integrate multiple distributed, heterogeneous, and independently managed data sources.
- The ability to provide efficient data transfer mechanisms and to provide data where the computation will take place for better scalability and efficiency.
- The ability to provide data caching and/or replication mechanisms to minimize network traffic.
- The ability to provide necessary data discovery mechanisms, which allow the user to find data based on characteristics of the data.

Computation

The core functional computational requirements for grid applications are:

- The ability to allow for independent management of computing resources
- The understanding current and predicted loads on grid resources, resource availability, dynamic resource configuration, and provisioning
- Failure detection and failover mechanisms
- Ensure appropriate security mechanisms for secure resource management, access, and integrity

Let us further explore some details on the computational and data grids as they exist today.

Computational and Data Grids

In today's complex world of high speed computing, computers have become extremely powerful as to that of (let's say) five years ago. Even the home-based PCs available on the commercial markets are powerful enough for accomplishing complex computations that we could not have imagined a decade prior to today.

The quality and quantity requirements for some business-related advanced computing applications are also becoming more and more complex. The industry is now realizing that we have a need, and are conducting numerous complex scientific experiments, advanced modeling scenarios, genome matching, astronomical research, a wide variety of simulations, complex scientific/business modeling scenarios, and real-time personal portfolio management. These requirements can actually exceed the demands and availability of installed computational power within an organization. Sometimes, we find that no single organization alone satisfies some of these aforementioned computational requirements. Thus the computational Grid Computing environment became a reality, which provides a demand-driven, reliable, powerful, and yet inexpensive computational power for its customers.

We can summarize the data requirements in the early grid solutions as follows:

- The ability to discover data
- The access to databases, utilizing meta-data and other attributes of the data
- The provisioning of computing facilities for high-speed data movement

- The capability to support flexible data access and data filtering capabilities

As one begins to realize the importance of extreme high performance-related issues in a Grid Computing environment, it is recommended to store (or cache) data near to the computation, and to provide a common interface for data access and management.

IV. CURRENT GRID ACTIVITIES

As described earlier, initially, the focused Grid Computing activities were in the areas of computing power, data access, and storage resources. The definition of Grid Computing resource sharing has since changed, based upon experiences, with more focus now being applied to a sophisticated form of coordinated resource sharing distributed throughout the participants in a virtual organization. This application concept of coordinated resource sharing includes any resources available within a virtual organization, including computing power, data, hardware, software and applications, networking services, and any other forms of computing resource attainment. This concept of coordinated resource sharing is depicted in

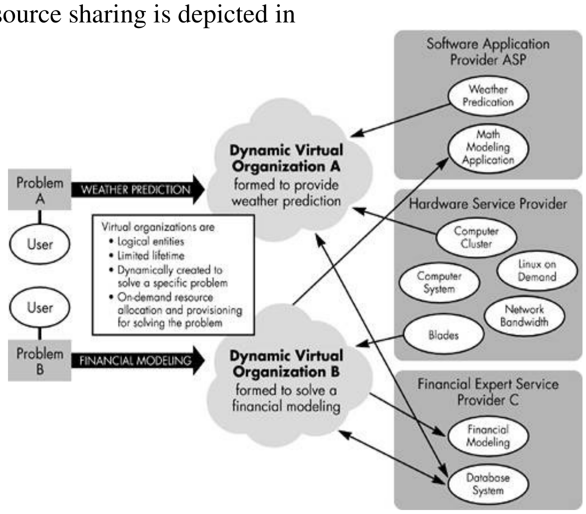


Figure 1 Grid activities

Through implementing this type of Grid Computing environment, these resources are immediately available to the authenticated users for resolving specific problems. These problems may be a software capability problem (e.g., modeling, simulation, word processing, etc.) or hardware availability and/or computing capacity shortage problems (e.g., processor computing resources, data storage/access needs, etc.). While on another level, these problems may be related to a networking bandwidth availability problem, the need for immediate circuit provisioning of a network, a security event or other event correlation issue, and many more types of critical environmental needs. Let us further explore this concept of "*virtualization*" by describing in more detail the usage patterns found within each of the virtual organizations.

- *A virtual organization for weather prediction*. For example, this virtual organization requires resources such as weather prediction software applications to perform the mandatory environmental simulations associated with predicting weather. Likewise, they will require very specific hardware resources to run the respective software, as well as high-speed data storage facilities to maintain the data generated from performing the simulations.
- *A virtual organization for financial modeling*. For example, this virtual organization requires resources such as software modeling tools for performing a multitude of financial analytics, virtualized blades to run the above software, and access to data storage facilities for storing and accessing data.

These virtual organizations manage their resources and typically will provision additional resources on an "as-needed" basis. This on-demand approach provides tremendous values toward scalability, in addition to aspects of enhanced reusability.

This approach is typically found in any "on-demand" environment. This capability is based upon a *utility* infrastructure, where resources are allocated as, and when, they are required. Likewise, their utility pricing scenarios are always based upon the capturing of usage metrics.

The following discussion introduces a number of requirements needed for such Grid Computing architectures utilized by virtual organizations. We shall classify these architecture requirements into three categories. These resources categories must be capable of providing facilities for the following scenarios:

- The need for dynamic discovery of computing resources, based on their capabilities and functions.
- The immediate allocation and provisioning of these resources, based on their availability and the user demands or requirements.
- The management of these resources to meet the required service level agreements (SLAs).

Virtual organization must be capable of providing facilities for:

- The formation of virtual task forces, or groups, to solve specific problems associated with the virtual organization.
- The dynamic collection of resources from heterogeneous providers based upon users' needs and the sophistication levels of the problems.
- The dynamic identification and automatic problem resolution of a wide variety of troubles, with automation of event correlation, linking the specific

problems to the required resource and/or service providers.

Users/applications typically found in Grid Computing environments must be able to perform the following characteristics:

- The clear and unambiguous identification of the problem(s) needing to be solved
- The identification and mapping of the resources required solve the problem
- The ability to sustain the required levels of QoS, while adhering to the anticipated and necessary SLAs

The above discussion helps us now to better understand the common requirements for grid systems. In the subsequent chapters in this section, and moreover throughout this book, we discuss the many specific details on the Grid Computing architecture models and emerging Grid Computing software systems that have proven valuable in supporting the above requirements.

V. BENEFIT OF GRID COMPUTING

Exploiting underutilized Resources

One of the basic uses of grid computing is to run an existing application on a different machine. The machine on which the application is normally run might be unusually busy due to a peak in activity. The job in question could be run on an idle machine elsewhere on the grid.

Parallel CPU capacity

The potential for massive parallel CPU capacity is one of the most common visions and attractive features of a grid. In addition to pure scientific needs, such computing power is driving a new evolution in industries such as the bio-medical field, financial modeling, oil exploration, motion picture animation, and many others.

Virtual resources and virtual organizations for collaboration

Another capability enabled by grid computing is to provide an environment for collaboration among a wider audience. In the past, distributed computing promised this collaboration and achieved it to some extent.

Access to additional resources

As already stated, in addition to CPU and storage resources, a grid can provide access to other resources as well. The additional resources can be provided in additional numbers and/or capacity.

Resource balancing

Grid federates a large number of resources contributed by individual machines into a large single-system image. For applications that are grid-enabled, the grid can offer a resource balancing effect by scheduling grid jobs on machines with low utilization.

Reliability

High-end conventional computing systems use expensive hardware to increase reliability. They are built using chips with redundant circuits that vote on results, and contain logic to achieve graceful recovery from an assortment of hardware failures.

Management

The goal to virtualize the resources on the grid and more uniformly handle heterogeneous systems will create new opportunities to better manage a larger, more distributed IT infrastructure.

It will be easier to visualize capacity and utilization, making it easier for IT departments to control expenditures for computing resources over a larger organization

VI. GRID ARCHITECTURE

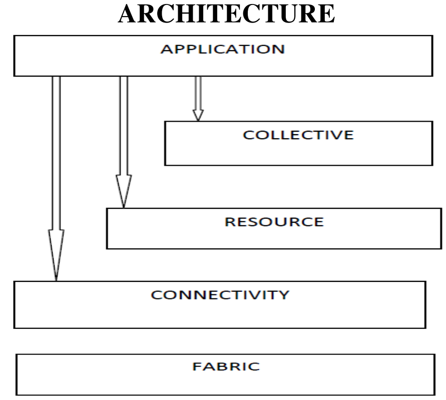
Descriptive: Provide a common vocabulary for use when describing Grid systems

Guidance: Identify key areas in which services are required

***Prescriptive*:** Define standard “Intergrid” protocols and APIs to facilitate creation of interoperable Grid systems and portable applications

Grids focused on integrating existing resources with their hardware, operating systems, local resource management, and security infrastructure. In order to support the creation of the so called “Virtual Organizations”—a logical entity within which distributed resources can be discovered and shared as if they were from the same organization, Grids define and provide a set of standard protocols, middleware, toolkits, and services built on top of these protocols.

VII. DIFFERENT LAYERS IN GRID ARCHITECTURE



Grids provide protocols and services at five different layers as identified in the Grid protocol architecture.

At the **Fabric layer**, Grids provide access to different resource types such as compute, storage and network resource, code repository, etc. Grids usually rely on existing fabric components, for instance, local resource managers.

Connectivity layer defines core communication and authentication protocols for easy and secure network transactions. The GSI (Grid Security Infrastructure) protocol underlies every Grid transaction.

The **Resource layer** defines protocols for the publication, discovery, negotiation, monitoring, accounting and payment of sharing operations on individual resources. The GRAM (Grid Resource Access and Management) protocol is used for allocation of computational resources and for monitoring and control of computation on those resources, and GridFTP for data access and high-speed data transfer.

The **Collective layer** captures interactions across collections of resources, directory services such as MDS (Monitoring and Discovery Service) allows for the monitoring and discovery of VO resources, Condor-G and Nimrod-G are examples of co-allocating, scheduling and brokering services, and MPICH for Grid enabled programming systems, and CAS (community authorization service) for global resource policies.

The **Application layer** comprises whatever user applications built on top of the above protocols and APIs and operate in VO environments.

VIII. OGSA

The Open Grid Services Architecture (OGSA) is a set of standards defining the way in which information is shared among diverse components of large, heterogeneous grid systems. It is a set of standards defining the way in which information is shared among diverse components of large, heterogeneous grid systems. In this context, a grid system is a scalable wide area network (WAN) that supports resource sharing and distribution. OGSA is a trademark of the Open Grid Forum.OGSA definitions and criteria apply to hardware, platforms and software in standards-based grid computing. The OGSA is, in effect, an extension and refinement of the service-oriented architecture (SOA). The OGSA addresses ongoing issues and challenges such as authentication, authorization, policy negotiation and enforcement, administration of service-level agreements, management of virtual organizations and customer data integration.

IX. GRID APPLICATION

Based on our earlier discussion, we can align Grid Computing applications to have common needs, such as what is described in (but not limited to) the following:

- Provisioning and distributing application codes to specific system nodes
- Application partitioning that involves breaking the problem into discrete pieces
- Discovery and scheduling of tasks and workflow
- Data communications distributing the problem data where and when it is required

Schedulers

Schedulers are types of applications responsible for the management of jobs, such as allocating resources needed for any specific job, partitioning of jobs to schedule parallel execution of tasks, data management, event correlation, and service-level management capabilities. These schedulers then form a hierarchical structure, with meta-schedulers that form the root and other lower level schedulers, while providing specific scheduling capabilities that form the leaves. There are schedulers that must provide capabilities for areas such as (but not limited to):

- Advanced resource reservation
- Service-level agreement validation and enforcement
- Rescheduling and corrective actions of partial failover situations

Resource Broker

The resource broker provides *pairing* services between the service requester and the service provider. This pairing enables the selection of best available resources from the service provider for the execution of a specific task scheduler for the resource execution task.
The pairing process in a resource broker involves allocation and support functions such as:

- Allocating the appropriate resource or a combination of resources for the task execution
- Supporting users' deadline and budget constraints for scheduling optimizations

Load Balancing

The Grid Computing infrastructure load-balancing issues are concerned with the traditional load-balancing distribution of workload among the resources in a Grid Computing environment. This load-balancing feature must always be integrated into any system in order to avoid processing delays and over commitment of resources.

Grid Portals

Grid portals are similar to Web portals, in the sense they provide uniform access to the grid resources. For example, grid portals provide capabilities for Grid Computing resource authentication, remote resource access, scheduling

capabilities, and monitoring status information. These kinds of portals help to alleviate the complexity of task management through customizable and personalized graphical interfaces for the users.
Some examples of these grid portal capabilities are noted in the following list:

- Querying databases or *LDAP* servers for resource-specific information
- File transfer facilities such as file upload, download, integration with custom software, and so on
- Manage job through job status feedbacks
- Allocate the resources for the execution of specific tasks
- Security management
- Provide personalized solutions

Integrated Solutions
These integrated solutions are a combination of the existing advanced middleware and application functionalities, combined to provide more coherent and high performance results across the Grid Computing environment.

X. GRID INFRASTRUCTURE

Grid computing infrastructure component must address several potentially complicated areas in many stages of the implementation. These areas are:

- Security
- Resource management
- Information services
- Data management

Let us further examine the significance of each of these above components.

Security

The heterogeneous nature of resources and their differing security policies are complicated and complex in the security schemes of a Grid Computing environment. These computing resources are hosted in differing security domains and heterogeneous platforms.The other security requirements are often centered on the topics of data integrity, confidentiality, and information privacy. The Grid Computing data exchange must be protected using secure communication channels, including *SSL*/TLS and oftentimes in combination with secure message exchange mechanisms such as WS-Security. The most notable security infrastructure used for securing grid is the Grid Security Infrastructure (*GSI*). In most cases, *GSI* provides capabilities for single sign-on, heterogeneous platform integration and secure resource access/authentication.

Resource Management
The tremendously large number and the heterogeneous potential of Grid Computing resources cause the resource management challenge to be a significant effort topic in Grid Computing environments. These resource management scenarios often include resource discovery, resource inventories, fault isolation, resource provisioning, resource monitoring, a variety of autonomic capabilities, and service-level management activities.

Information Services
Information services are fundamentally concentrated on providing valuable information respective to the Grid Computing infrastructure resources. These services leverage and entirely depend on the providers of information such as resource availability, capacity, and utilization, just to name a few. These information services enable service providers to most efficiently allocate resources for the variety of very specific tasks related to the Grid Computing infrastructure solution.

Data Management
Data forms the single most important asset in a Grid Computing system. This data may be input into the resource, and the results from the resource on the execution of a specific task. If the infrastructure is not designed properly, the data movement in a geographically distributed system can quickly cause scalability problems. It is well understood that the data must be near to the computation where it is used. The current advances surrounding data management are tightly focusing on virtualized data storage mechanisms, such as storage area networks (*SAN*), network file systems, dedicated storage servers, and virtual databases. These virtualization mechanisms in data storage solutions and common access mechanisms (e.g., relational SQLs, Web services, etc.) help developers and providers to design data management concepts into the Grid Computing infrastructure with much more flexibility than traditional approaches.

XI. CONCLUSION

So far we have been describing and walking through overview discussion topics on the Grid Computing discipline that will be discussed further throughout this book, including the Grid Computing architecture, the applications, and the infrastructure requirements for any grid environment.

In addition to this, we have discussed when one should use Grid Computing disciplines, and the factors developers and providers must consider in the implementation phases. With this introduction we can now explore deeper into the various aspects of a Grid Computing system, its evolution across the industries, and the current architectural efforts underway throughout the world.

The proceeding chapters in this book introduce the reader to this new, evolutionary era of Grid Computing, in a concise, hard-hitting, and easy-to-understand manner.

REFERENCES

[1] Foster and C. Kesselman, "The Grid: Blue print for a new computing infrastructure", Morgan Kaufmann Publications (1999).

[2] Foster, C. Kesselman, J. M. Nick and S. Tuecke, "The physiology of the Grid: An open grid services architecture for distributed systems integration", Grid Forum white paper, 2003.

[3] Volker Sander, "Networking Issues for Grid Infrastructure", GFD-I.037, Nov, 22, 2004.

[4] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A.Roy. "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation", Intl Workshop on Quality of Service, 1999.

[5] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, M. Wilde. "Falkon:a Fast and Light-weight tasK executiON framework",IEEE/ACM SuperComputing 2007.

[6] Foster, I. & Kesselman, C. (Eds). The Grid: Blueprint for a New Computing Infrastructure. Morgan-Kaufmann (1999).

[7] Foster, I. & Kesselman, C. Globus: A Toolkit-Based Grid Architecture. In ref. 2, pages 259-278. Morgan Kaufmann Publishers (1999).

[6] The Globus Security Team. "Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective," Technical Report, Argonne National Laboratory, MCS, 2005.

[9] *Technology and Strategy Perspectives* (Fellenstein, 2004) for further details and precision on important technologies, Grid Computing, and key strategy perspectives.

[10] Fellenstein (2004) for further details and precision on important technologies, Grid Computing, and key strategy perspectives.

[11] Fellenstein (2004) for further details and precision on important service provider technologies, Grid Computing, and key strategy perspectives on both topics.

