Volume 3, Issue 11, November 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Survey Paper on Grid Computing

Priyanka C A[1]
Mtech (Software Engineering),
VTU, India

***Abstract**—The last decade has seen a substantial increase in commodity computer and network performance, mainly as a result of faster hardware and more sophisticated software. Nevertheless, there are still problems, in the fields of science, engineering, and business, which cannot be effectively dealt with using the current generation of supercomputers. In fact, due to their size and complexity, these problems are often very numerically and/or data intensive and consequently require a variety of heterogeneous resources that are not available on a single machine. A number of teams have conducted experimental studies on the cooperative use of geographically distributed resources unified to act as a single powerful computer. This new approach is known by several names, such as meta computing, scalable computing, global computing, Internet computing, and more recently Grid computing. In this article, we define grid computing, first, we review the "Need of Grid computing". Next, we present an extensible and open Grid architecture, in which protocols, services, application programming interfaces, and software development kits are categorized according to their roles in enabling resource sharing. Finally, we discuss benefits and application of grid computing.*

***Keywords**— Grid Computing, Need of Grid Computing, Grid Architecture, Distributed Computing, Benefits of Grid Computing and Applications of Grid Computing*

I. INTRODUCTION

The popularity of the Internet as well as the availability of powerful computers and high-speed network technologies as low-cost commodity components is changing the way we use computers today. These technology opportunities have led to the possibility of using distributed computers as a single, unified computing resource, leading to Grid computing. A computational grid is a hardware and software infrastructure that provides high speed and inexpensive access to high end computational facilities. Grid Computing is being extensively used in various areas like science, business, governments etc. The areas and scope of application is increasing at a very fast pace. Grid computing was introduced in 1970 by Foster and Kesselman. The term Grid Computing originated in the early 1990s as a metaphor for making computer power as easy to access as an electric power grid. CPU Scavenging and Volunteer became popular in 1997 by distributed.net and later in 1991 by SETI@home to harness the power networked PCs worldwide, in order to solve CPU-intensive research problems [Berman, et al, 2003]. The concept of grid computing was the brainchild of Ian Foster, Carl Kesselman and Steve Tuecke, and it was made known in one of their seminal presentations, "The Grid: Blueprint for a new computing infrastructure." Their efforts created the Globus Toolkit, which contains computation management, storage management, security management, monitoring and other related services. They are widely acclaimed as the "father of the grid". A Grid system is a virtual organization comprising several independent autonomous domains [Akinyemi, etal, 2007]. The Grid community often refers to the notion of a "virtual organization" (VO). Grid enables people to be members of many VOs which give one access to various computational, instrument-based data and other types of resources. VO makes its resource much more useful and accessible for their users [Joseph, et al, 2004].

II. NEED OF GRID COMPUTING

Computational approaches to problem solving have proven their worth in almost everywhere. Computers are used for modelling and simulating complex scientific and engineering problems, diagnosing medical conditions, controlling industrial equipment, forecasting the weather, managing stock portfolios, and many other purposes. Yet, although there are certainly challenging problems that exceed our ability to solve them, computers are still used much less extensively than they could be. To pick just one example, university researchers make extensive use of computers when studying the impact of changes in land use on biodiversity, but city planners selecting routes for new roads or planning new zoning ordinances do not. Yet it is local decisions such as these that, ultimately, shape our future. There are a variety of reasons for this relative lack of use of computational problem-solving methods, including lack of appropriate education and tools. But one important factor is that the average computing environment remains inadequate for such computationally sophisticated purposes.

III. GRID ARCHITECTURE

Our goal in describing our Grid architecture is not to provide a complete enumeration of all required protocols (and services, APIs, and SDKs) but rather to identify requirements for general classes of component. The result is an extensible, open architectural structure within which can be placed solutions to key VO requirements.

Practical grids are generally described in terms of layers (see Fig 1). The lowest layers (the 'platform') comprise the hardware resources, including computers, networks, databases, instruments, and interface devices. These devices, which will be geographically distributed, may present their data in very different formats, are likely to have different qualities of service (e.g. communication speeds, bandwidth) and are likely to utilize different operating systems and processor architectures. A key concept is that the hardware resources can change over time - some may be withdrawn, upgraded or replaced by newer models, others may change their performance to adapt to local conditions.

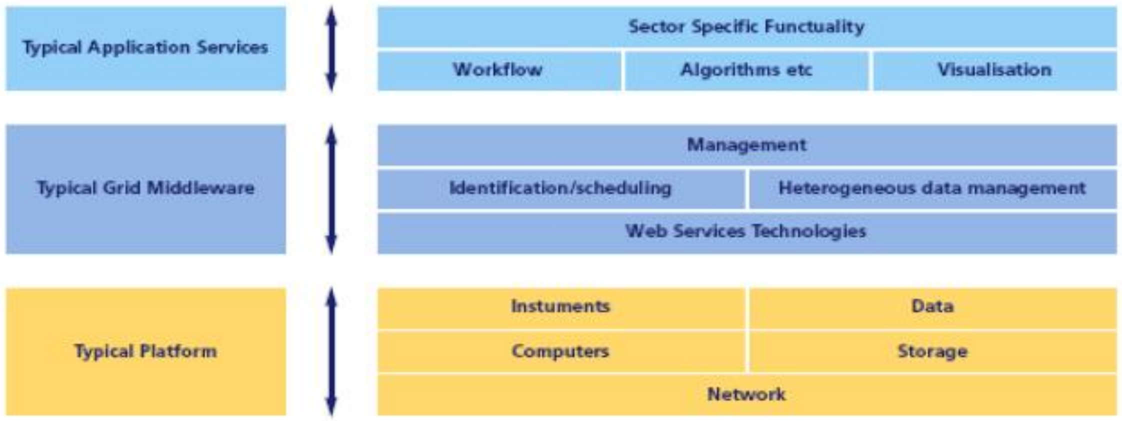

Figure 3.1: Layered Grid Architecture

The middle layers (sometimes referred to as 'middleware') provide a set of software functions that 'buffer' the user from administrative tasks associated with access to the disparate resources. These functions are made available as services and some provide a 'jacket' around the hardware interfaces, such that the different hardware platforms present a unified interface to different applications. Other functions manage the underlying fabric, such as identification and scheduling of resources in a secure and auditable way. The middle layer also provides the ability to make frequently used patterns of functions available as a composed higher-level service using workflow techniques. The highest layers contain the user 'application services'. Pilot projects have already been carried out in user application areas, such as life sciences (e.g. computational biology, genomics), engineering (e.g. simulation and modelling, just in time maintenance) and healthcare (e.g. diagnosis, telematics). These services could include horizontal functions such as workflow (the linkage of multiple services into a single service), web portals, data visualization and the language/semantic concepts appropriate to different application sectors.

Components within each layer (in figure 3.1) share common characteristics but can build on capabilities and behaviour's provided by any lower layer. In specifying the various layers of the Grid architecture, we follow the principles of the "hourglass model". The neck of the hourglass defines a fundamental set of core abstractions and protocols, onto which many different high-level behaviours can be mapped (the top of the hourglass), and which themselves can be mapped onto many different underlying technologies The architecture of grid computing is explained based on the hourglass. The neck of the hourglass consists of *Resource* and *Connectivity* protocols, which facilitate the sharing of individual resources. Protocols at these layers are designed so that they can be implemented on top of a diverse range of resource types, defined at the *Fabric* layer, and can in turn be used to construct a wide range of global services and application-specific behaviours at the Collective layer

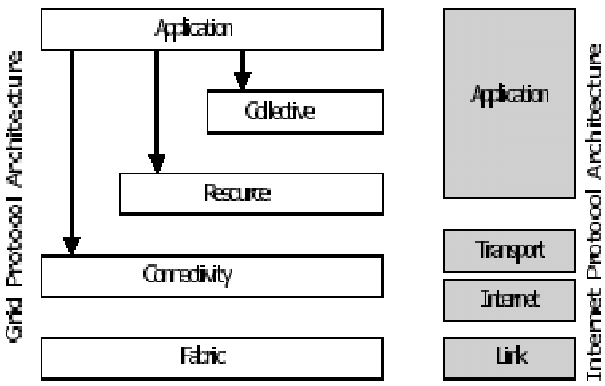

Figure3.2 The layered Grid architecture and its relationship to the Internet protocol architecture.

A. Fabric: Interfaces to Local Control

The Grid *Fabric* layer provides the resources to which shared access is mediated by Grid protocols: for example, computational resources, storage systems, catalogues, network resources, and sensors. A "resource" may be a logical entity, such as a distributed file system, computer cluster, or distributed computer pool.

Fabric components implement the local, resource-specific operations that occur on specific resources (whether physical or logical) as a result of sharing operations at higher levels. There is a tight and subtle interdependence between the functions implemented at the Fabric level, on the one hand, and the sharing operations supported, on the other. Richer Fabric functionality enables more sophisticated sharing operations; at the same time, if we place few demands on Fabric elements, then deployment of Grid infrastructure is simplified. For example, resource-level support for advance reservations makes it possible for higher-level services to aggregate resources in interesting ways that would otherwise be impossible to achieve. However, as in practice few resources support advance reservation "out of the box," a requirement for advance reservation increases the cost of incorporating new resources into a Grid.

B. Connectivity : Communicating Easily and Securely

The *Connectivity* layer defines core communication and authentication protocols required for Grid-specific network transactions. Communication protocols enable the exchange of data between Fabric layer resources. Authentication protocols build on communication services to provide cryptographically secure mechanisms for verifying the identity of users and resources.

Communication requirements include transport, routing, and naming. While alternatives certainly exist, we assume here that these protocols are drawn from the TCP/IP protocol stack: specifically, the Internet (IP), transport (TCP, UDP), and application (DNS etc.)

With respect to security aspects of the Connectivity layer, we observe that the complexity of the security problem makes it important that any solutions be based on existing standards whenever possible.

C. Resource :Sharing Single Resource

The Resource layer builds on Connectivity layer communication and authentication protocols to define protocols for the secure negotiation, initiation, monitoring, control, accounting, and payment of sharing operations on individual resources. Resource layer implementations of these protocols call Fabric layer functions to access and control local resources. Resource layer protocols are concerned entirely with individual resources and hence ignore issues of global state and atomic actions across distributed collections.

D. Collective : Coordinating Multiple Resources

While the Resource layer is focused on interactions with a single resource, the next layer in the architecture contains protocols and services (and APIs and SDKs) that are not associated with any one specific resource but rather are global in nature and capture interactions across collections of resources. For this reason, we refer to the next layer of the architecture as the *Collective* layer. Because Collective components build on the narrow Resource and Connectivity layer "neck" in the protocol hourglass, they can implement a wide variety of sharing behaviours without placing new requirements on the resources being shared.

E. Application layer

The final layer in our Grid architecture comprises the user applications that operate within a VO environment. (Figure3.3) illustrates an application programmer's view of Grid architecture. Applications are constructed in terms of, and by calling upon, services defined at any layer. At each layer, we have well-defined protocols that provide access to some useful service: resource management, data access, resource discovery, and so forth. APIs are implemented by software development kits (SDKs), which in turn use Grid protocols to interact with network services that provide capabilities to the end user. Solid lines represent a direct call; dash lines protocol interactions

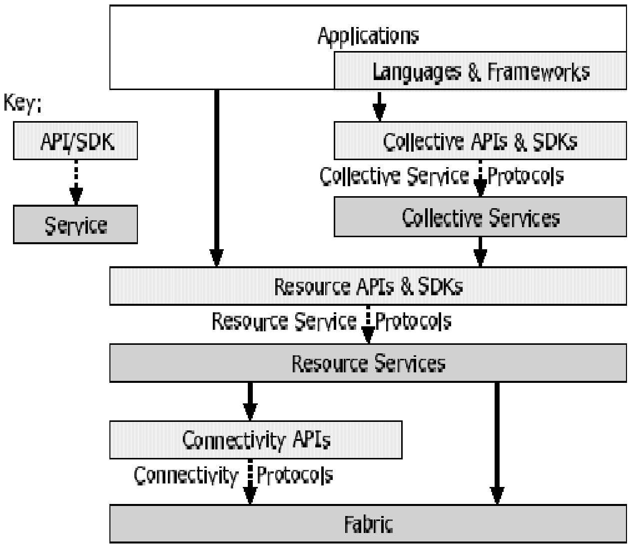

Figure3.3: illustrates an application programmer's view of Grid architecture

IV BENEFITS OF GRID COMPUTING

Grid computing can provide many benefits not available with traditional computing models:

• *Better utilization of resources* — Grid computing uses distributed resources more efficiently and delivers more usable computing power. This can decrease time-to-market, allow for innovation, or enable additional testing and simulation for improved product quality. By employing existing resources, grid computing helps protect IT investments, containing costs while providing more capacity.

• *Increased user productivity* — By providing transparent access to resources, work can be completed more quickly. Users gain additional productivity as they can focus on design and development rather than wasting valuable time hunting for resources and manually scheduling and managing large numbers of jobs.
• *Scalability* — Grids can grow seamlessly over time, allowing many thousands of processors to be integrated into one cluster. Components can be updated independently and additional resources can be added as needed, reducing large one-time expenses.
• *Flexibility* — Grid computing provides computing power where it is needed most, helping to better meet dynamically changing work loads. Grids can contain heterogeneous compute nodes, allowing resources to be added and removed as needs dictate.

V APPLICATIONS OF GRID COMPUTING

Grid Computing is being extensively used in various areas like science, business, governments etc. The areas and scope of application is increasing at a very fast pace. In this section a brief discussion on the application of Grid Computing in various domains is done. This is intended to give only an idea of the application of Grid Computing and represents only the tip of the iceberg. The fig. 5.1 shows some of the possible application areas.

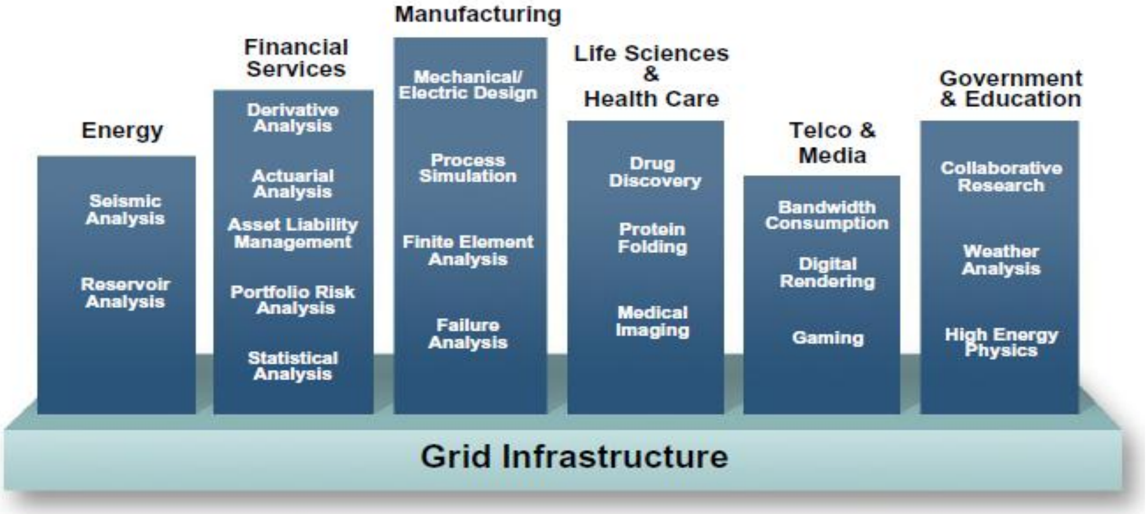

Figure5.1: Grid Computing Applications(source: Watkins 2006)

A. Energy

Grid Computing have been extensively used in analysis of seismic data coming from under earth's surface. Seismic data from earth movements are being used for understanding, modelling and forecasting of earthquakes. The discovery and evaluation of oil and gas reservoirs have long required powerful computers and processing of very large amounts of data. Offshore oil and gas exploration requires immense levels of information processing, seismic imaging is the key technology in discovering these oil and gas fields. Grid Computing has been of immense help in processing these huge volumes of data at a very high speed.

B Financial Services

With the emergence of a competitive market force, customer satisfaction, and reduction of risk are the most competitive areas financial communities continually strive to achieve. The requirements related to sophistication, accuracy, and faster executions are among the more salient objectives across financial communities. These objectives are now achieved by real-time access to the current and historical market data, complex financial modelling based on the respective data, and faster response times to user queries.

C Manufacturing

The business and industry sectors need mechanisms to capture data, speed up the analysis on the data, and provide faster responses to market needs Grid Computing systems provide a wide range of capabilities that address these kinds of analysis and modelling activities.

D Life Science & Health Care

Life Science

The life sciences sectors have to provide rapid changes in the way that drug treatment and drug discovery efforts. The complexity of processing data, there needs to be additional requirements surrounding data security, secure data access, secure storage, privacy, and highly flexible integration. The Grid Computing systems can provide a common infrastructure for data access, and at the same time, provide secure data access mechanisms while processing the data. Today, life science utilizes the Grid Computing systems to execute sequence comparison algorithms and enable molecular modelling using the above-collected secured data.

Health Care

One of the important applications of grid computing is in medicine. Grid computing has been useful in a variety of situations like producing interactive medical simulations like heart simulation and in analysing and managing medical images. Grids have also been successfully used in supporting virtual collaboration in Hospitals (a virtual network of hospitals that delivers medical training, e-surgery, and medical analysis services).

E Telecom & Media

Media

The major challenge facing media applications is the production, broadcasting, delivery and play out of interactive media content (audio, video, and image) in real time. Grid solutions have been developed for this; these include, the Grid Visualization Kernel (GVK) which allows the visualization pipeline to be ported to grid resources, and also handles the communication between the simulations.

Collaborative Games

There are collaborative types of Grid Computing disciplines that are involving emerging technologies to support online games, the resources are selected based on the requirements, often involving aspects such as volume of traffic and number of players, rather than centralized servers and other fixed resources. Grid computing gaming environments are capable of supporting such virtualized environments for enabling collaborative gaming (Maad et al., 2005).

F Government & Education

Government

The Grid Computing environments in government focus on providing coordinated access to massive amounts of data held across various agencies in a government. This provides faster access to solve critical problems, such as emergency situations, and other normal activities. These key environments provide more efficient decision making with less turnaround time. Grid Computing enables the creation of virtual organizations, including many participants from various governmental agencies (e.g., local, state and central). (Maad et al., 2005).

Education

There is a big difference in education sector between first world and developing world. This difference is based on digital learning resources and computing power. Unfortunately all of these resources are geographically distributed all over the world. Grid is the technologies that can integrate all of these resources of knowledge's and produce super-computing power from those geographically distributed computers to access those knowledge's without sacrificing local autonomy(GF/117). For example, the Indian Institute of Information Technology and Management, Kerala is heading a Grid project known as Kerala Educational Grid that envisages the linkage of colleges and universities to resource centers that will supply education materials on demand and increase cooperation and networking among the affiliated academics (Sherly, 2005).

Environment

Environment applications involve parallel execution of hundreds of programs corresponding to large scale air pollution, nuclear waste storage, pollution and climate models. Various grid *solutions* have been offered. For a particular large scale air pollution model, a national VO was formed involving researchers and resources from 7 UK institutions. At an international level, a similar VO was formed involving 21 EU institutions within the CrossGrid consortium (CrossGrid).

Weather Forecasting

Another field with huge data generation and processing requirements is weather forecasting. Both local weather data stations and satellites collect and transmit large volumes of data for analysis. Grid Computing has been very extensively used for forecasting of weather and other natural catastrophic events.

Research Collaboration

Research-oriented organizations and universities undertaking advanced research collaboration require the analysis of tremendous amounts of data. Grid computing discipline provides mechanisms for resource sharing by forming one or more virtual organizations providing specific sharing capabilities

Astronomy

The major *challenge* facing the field of astronomy is the analysis of tera-bytes of astronomical image data generated by telescopes. Moreover, astronomical image capturing devices can generate several images, each of hundreds of Mbytes, per single shot (Yamamoto, 2004). This necessitates: data intensive computation; scalable file I/O in the order of GB/s; replica management; and parallel / distributed processing of files. Grid computing is essential and has been successfully used for such data intensive.

G Engineering and Design

The enormous competitive pressure in the business and industry sectors today afford most engineering and design far less turnaround time. They need mechanisms to capture data, speed up the analysis on the data, and provide faster responses to market needs. These complexities fall into several areas of solutions in Grid Computing that span across industry sectors all over the world. These complexities are described the following areas:

- The analysis of real-time data to find a specific pattern within a problem
- The parametric studies to verify different aspects of the systems
- The modelling experiments to create new designs
- The simulation activities to verify the existing models for accuracy

Grid computing systems provide a wide range of capabilities that address the above kinds of analysis and modelling activities. These advanced types of solutions also provide complex job schedulers and resource managers to deal with computing power requirements.

VI. CONCLUSION

Grid Computing has emerged as a powerful tool for enhancing the computer power. This technology also called 'poor man's supercomputer' uses the idle capacities of various geographically spread resources. These idle capacities can be used to perform calculations and activities which otherwise would take long time to perform. Grid Computing has been briefly explained here. The benefits of Grid Computing and various applications have also been discussed.

REFERENCES

[1] Watkins, R. 2006. IBM Grid Computing & Virtualization, GRID @ Asia Conference, Seoul, Korea, December 11-13

[2] Armstrong, R., Gannon, D., Geist, A., Keahey, K., Kohn, S., McInnes, L. and Parker, S.Toward a Common Component Architecture for High Performance Scientific Computing. In *Proc. 8th IEEE Symp. on High Performance Distributed Computing*,1999.

[3] Akinyemi, I.O., Daramola, J.O. and Adebiyi, A. A.(2007), Grid-Enabled e-learning Framework for NigeriaEducational Institutions, Nigeria Computer Society, 21st National Conference Proceedings, Vol. 18, pp. 91-98.

[4] Berman, F., Anthony J. G. H. and Geoffrey C. F. (2003). Grid Computing: Making the global infrastructure a reality, ISBN 0-12-742503-9.

[5] Joseph, J., Ernest, M. and Fellenstein, C. (2004), Evolution of Grid Computing Architecture and Grid Adoption Models, IBM Systems Journal, Vol. 43, No. 4, pp. 624 – 645.

[6] Maad S., Coghlan B., John R., Eamonn K., Watson R., and Pierantoni G. 2005b. 'The Horizon of the Grid For Eovernment'. Proceeding eGovernment'05 Workshop, Brunel, United Kingdom.

[7] Sherly, E. 2005. Data Grid Architecture for a Distributed Data Management system: Kerala Educational GridPerspective. CSI Communications, July, 2005.

[8] Yamamoto N., Tatebe O., and Sekiguchi S. 2004. "Parallel and Distributed Astronomical Data Analysis on Grid Datafarm", Proceedings of 5th IEEE/ACM International Workshop on Grid Computing (Grid 2004), 461-466.

