Volume 7, Issue 2, February 2017 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Grid Computing: An Insight into Heterogeneous Computing Environment

Rameez Mushtaq Dar
Department of Computer Science
IUST, Awantipora, Pulwama, J&K, India

Romana Riyaz
Department of Computer Science
IUST, Awantipora, Pulwama, J&K, India

DOI: 10.23956/ijarcsse/V7I2/0123

Abstract -Duringthe last decade there has been a significant increase incomputational capability of computers, data storage and the way communication takes place. However, with such advancement, there still exist someproblems which are quite complex in nature and require a very high computation power to solve them. The extensive use and increased popularity of the Internet and the availability of high-speed communicational networks have significantly changed the way we do computing. These technologies have enabled the coordination and cooperative use of a wide variety of geographically distributed resources to come together as a single more powerful computer. Such a new method of pooling resources to solve large-scale complex problems is called as grid computing. This paper describes the concepts and components underlying the grid computing.

Keywords: Grid components, Complexity levels, fabric layer, Rehosting, OGSA, OGSI.

I. INTRODUCTION

The main purpose of distributed computing is to provide users with a platform which allows them a low cost, easy, and transparent method of access to a wide variety of heterogeneous resources. Such an integration of multiple different resources over a network is called Metacomputing. Metacomputing done at lowest level on local area networks (LAN) is termed as Cluster Computing whereas integration of vide variety of heterogeneous resources done on wide area networks (WAN) is termed as Grid Computing.

Grid computing is a virtualized distributed computing environment that aims at allowing the dynamic selection, sharing, and integration of distributed heterogeneous resources based on the availability, capability, performance and cost of these computing resources. It was actually in the mid-1990s the word grid was applied to computing by the extension and advancement of concept "computer Time-sharing". Fundamentally, the concept of grid computing is quite straightforward such that with grid computing an organization can seamlessly integrate, access and share dispersed heterogeneous resources like, hosts, servers, storage systems, data, and networks into one collaborative, in order to solve a specific task or achieve a specific goal[1].

In this paper we have presented detailed description of grid computing concepts, its characteristics and other important concepts which help in understanding grid computing architecture. Section II describes various characteristics of a computational grid, Section III describes the Grid Components and Complexity levels, Section IV provides a detailed grid architecture, Section V provides a detailed comparison of different approaches of grid computing, section VI presents descriptions of various standards used in grid computing. Finally section VII presents the conclusion of the study.

II. CHARACTERISTICS OF A COMPUTATIONAL GRID

There are multiple features and characteristics required by gird in order to provide a user with grid efficient, low cost, and transparent computing environment. Some of the important characteristics that exist at the core of grid computing are listed as: [2],[3]

1. **Heterogeneity:** The grid must involve a vast numbers of resources that are quite varied in nature and can encompass a large geographical distance through various domains.
2. **Scalability**: The grid should be flexible and tolerant to addition of resources or functionality should. It should be able to handle a large number of nodes without any performance degradation.
3. **Fault Tolerant**: A grid should be tolerant of unexpected computational aborts, hardware or software faults. Such faults are generally high in distributed environment and are generally handled by Resource Managers
4. **Security**: All the user participating computers should be protected from any malicious manipulations or interventions.
5. **Reliability**: The grid must be reliable in case any nodes break down or in case of any software and hardware faults. It must be able to continue provide it services in case of such faults. This is can be easily achieved if a grid has large number vide variety of resource at disposal.

III. GRID COMPONENTS AND COMPLEXITY LEVELS

A. Grid Components:

The grid systems involve a number of components distributed at three levels. Some of the important components that are required form a grid are listed as[4]: Fig. 1

1. **Application level components:** This level involves the Applications and High level Interfaces. Applications can be highly varied and includes large variety of problems ranging from small to complex fields. The high level interfaces allow the applications and users to interact and access the middleware services.
2. **Middleware level components:** This level involves services like Resource search, Resource discovery, Resource scheduling and allocation, fault tolerance, security mechanisms and load balancing. It also provides the users a transparent view of the resources available.
3. **Resource level**: This layer typically provides local services that involve computational resources like CPU cycles, storage, computers, Network infrastructure, software etc.

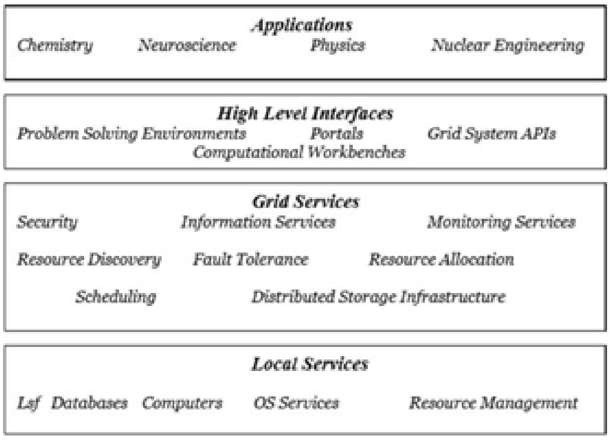

Figure 1. Grid Components

B. Complexity levels of grid:

Grids can be built ranging from just a few processors to large groups of processors organized as a hierarchy that may cover a continent or the globe. [5]

- **Local grid:** At initial level the simplest grid consists of just a few processors, all of which have the same hardware architecture and utilize the same operating system. These processors are connected in a data centre on a LAN or storage area network. Fig.2
- **Intra grid:** A next level of complexity is reached when one includes heterogeneous processors in the grid ensemble. These grids are also referred to as an intragrids or enterprise grids. As the term implies, processors participating in the enterprise grid may include devices owned and maintained by multiple departments, but still within one firm Fig.3
- **Intergrid:** Another level of grid is called as a pure grid, or an intergrid. An intergrid is used to collaborate on "large" projects of common interest. The high levels of security are usually required in this environment. Fig.4

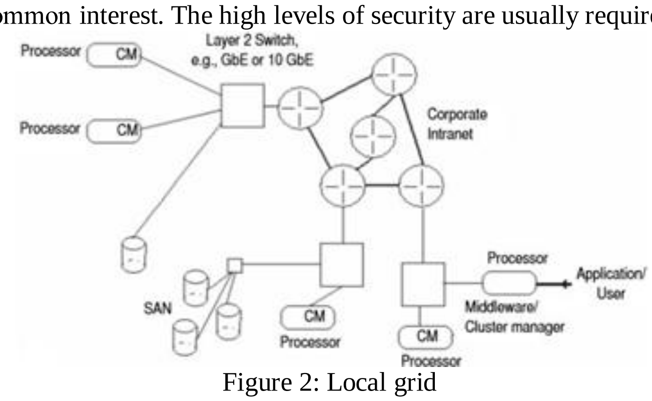

Figure 2: Local grid

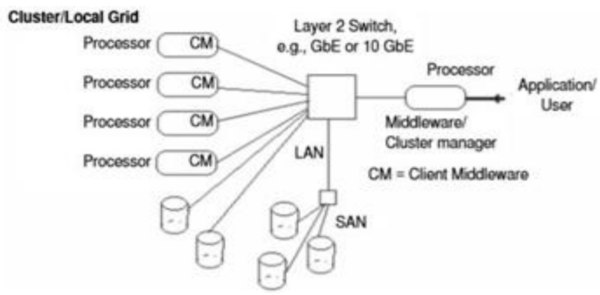

Figure 3: Intra grid

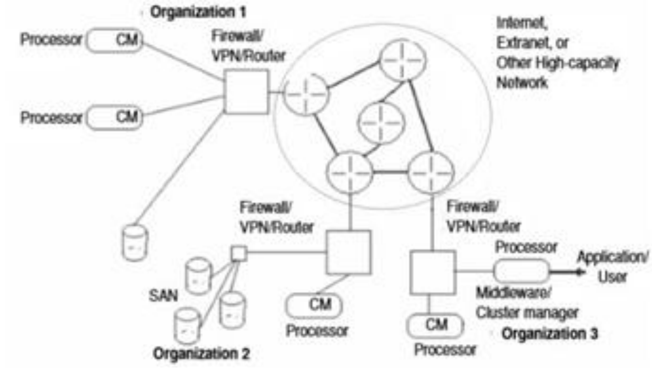

Figure 4: Inter grid

IV. GRID ARCHITECTURE

The architecture of the Grid is usually described in terms of layers, each providing a specific function. In grid architecture, the higher layers are concerned with the user whereas the lower layers are more emphasised on computers and networks[1]. Fig 5.

- **Fabric layer**: The Grid *Fabric* layer provides the resources which are shared on the basis of well defined grid protocols.Its components involve local resource related operations that occur as a result of sharing operations at higher levels. There is thus a close interdependence between the functions implemented at this level and the sharing operations supported, on the other level. More the Fabric functionality more sophisticated will be sharing operations at the same time[6].
- **Connectivity Layer:** The middleware layer provides the tools that enable the various elements such as servers, storage, networks etc. to participate in a unified Grid environment. The *Connectivity* layer defines main communication and authentication protocols which are 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 secure mechanisms for verifying the identity of users and resources.
- **Resource Layer:** The resource layer is made up of the actual resources that are part of the Grid, such as computers, storage systems, network components etc which can be connected directly to the network. Resource and connectivity protocols handle all grid specific network transactions between different computers and other resources on the Grid. Resource layer protocols can be distinguished primarily into two classes, which are Information Protocols and Management Protocols. While the former one is used to obtain the necessary information about the structure and the state of the resource and latter one is used to negotiate the access to the shared resources. Fig. 6

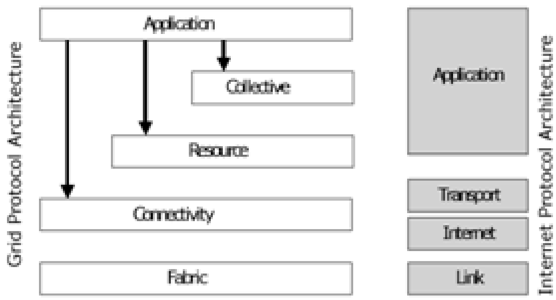

Figure: 5 Grid Architecture

- **Collective Layer**: This layer Deals with 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. [7]

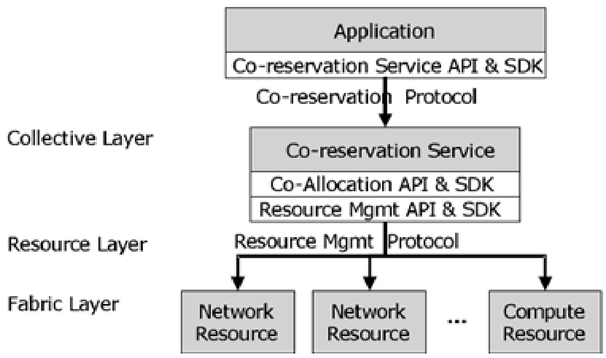

Figure 6: Collective and Resource layer protocols

- **Application Layer:** The highest layer of the grid architecture is the application layer, which includes all different user applications (science, engineering, and business, financial), portals and development toolkits supporting the applications. This is the layer that users interact with. Applications are developed in terms of services defined at any layer and called accordingly. At each layer, there are well-defined protocols providing access to some useful services like resource management, data access, resource discovery etc.[8]

V. COMPARATIVE ANALYSIS OF GRID COMPUTING APPROACHES

A. *Grid vs Clusters*

- The distinction between clusters and grids relates to the way resources are managed. In the case of clusters, the resource allocation is performed by a centralized resource manager and scheduling system. In addition to this, nodes cooperatively work together as a single unified resource whereas in the case of grids, each node has its own resource manager and does not aim at providing a single system view.[1]
- When local hosts are aggregated in tightly coupled configuration they tend to form the cluster parallel-based computing type but when geographically dispersed hosts are collaborated in distributed computing configurations, they tend to form the grid computing type.
- Like clusters and distributed computing, grids bring computing resources together. But Unlike clusters and distributed computing, which are physically proximate and have similar operations , grids can be geographically distributed and heterogeneous[1]

B. *Grid vs Web*

- The Web is mainly focused on communication whereas grid computing enables resource sharing and collaborative resource bind towards common goal.
- Web services provide standard infrastructure for data exchange between two different distributed applications, whereas grids provide an infrastructure for aggregation of high-end resources.
- While most Web services involve static processing and moveable data, many grid computing mechanisms involve static data and moveable processing.[1]

C. *Grid vs Virtualization*

- Grid computing also differs from virtualization. Virtualization has somewhat more of an emphasis on local resources, whereas grid computing has more of an emphasis on geographically distributed inter-organizational resources.
- Like virtualization technologies, grid computing enables the virtualization of IT resources. But, Unlike virtualization technologies, which virtualize a single system, grid computing enables the virtualization of broad-scale and disparate IT resources[9]

D. *Grid vs Peer-to-Peer:*

- In case with Peer-to-Peer environments, grid computing allows users to share files but unlike P2P, grid computing allows many-to-many sharing.
- With grid computing the sharing is not only in reference to files but other resources as well. The grid community generally focuses on aggregating distributed high-end machines such as clusters, whereas the P2P community concentrates on sharing low end systems such as PCs connected to the Internet [10]

E. *Grid vs Rehosting:*

- Grid computing, although potentially related to a Rehosting, but is not only that. Rehosting implies the reduction of typically a large number of servers to a smaller set of more powerful and more modern servers. It is a way to preserve the expensive investments in business logic and business data by moving to an open and more extensible architecture[1].There are savings associated with Rehosting. Also, applications are still assigned specific servers. Grid computing, on the other hand, permits the true virtualization of the computing function and applications are not reassigned a server, but the dynamic assignment is made based on real-time conditions[10]

VI. STANDARD USING GRID COMPUTING

The fundamental purpose of a grid computing is to make use of broadly distributed computing power across any kind of network but without standards one is actually limiting rather than extending one's ability utilize spare computer power on remote computers .Without industry-wide standards, it is a quite big technical challenge to achieve highly effective interactions among resources.

- **Open Grid Services Architecture (OGSA):**OGSA standard describes the way in which are service are provided to different layers of grid system by application layers components through a set of communication protocol over a network. It was developed within the Open Grid Forum, called the Global Grid Forum (GGF) at the time around 2002-2206. It is a distributed interaction and computing standard assuringinteroperability on heterogeneous systems so that different types of resources can communicate and share information.[11], [12]
- **Open Grid Services Infrastructure (OGSI):**The Open Grid Services Infrastructure (OGSI)This standard was published by the Global Grid Forum (GGF) as a proposed recommendation in June 2003. It was intended to

provide an infrastructure layer for the Open Grid Services Architecture (OGSA). OGSI takes the statelessness issues by extending Web services to accommodate grid computing resources that are both transient and stateful.[13]

VII. CONCLUSION

In this paper, we have given the insight into grid computing with some comparison to some conventional approaches. Grid Computing can has wide potential to deal with large and complex problem involving high computation. We can deal with large projects by integrating and making the different and vide variety resources to work together in the form of grid computing. An encouraged effort for merging resources can eventually lead to a whole new world where the Gridcomputing turns out to be a part of the everyday life of all computer users.

REFERENCES

[1] Daniel minoli,"A networking approach to grid computing", *John wiley& sons, inc*., publication isbn 0-471-68756-1

[2] M.A. Baker, R. Buyya, and D. Laforenza, "The Grid: International Efforts in Global Computing", *SSGRR 2000 The Computer & eBusiness Conference*, Aquila, Italy July 31. 2000 - August 6. 2000

[3] Cecile Germain, Vincent Neri, Gilles Fedak and Franck Cappello Xtrem, "Web : building an experimental platform for global computing Grid" :December 2000, *IEEE* Press.

[4] Dorian C. Arnold, Sathish S. Vadhiyar, and Jack Dongarra,"On the Convergence of Computational and Data Grids",*Parallel Processing Letters*, Volume 11, Numbers 2 and 3, June and September 2001, pp 187-202. ISSN 0129-6264.

[5] V. Berstis, "Fundamentals of Grid Computing," *Redbooks Paper*, 2002, IBM Corporation, 1133 Westchester Avenue, White Plains, New York 10604.

[6] I. Foster, C. Kesselman, S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations*". Intl. J. Supercomputer Applications*, 2001.

[7] V. Berstis. "Fundamental of Grid Computing". *IBM Redbooks paper* 2002.

[8] F. Berman, A. J. G. Hey, G. C. Fox, "*Grid Computing, Making the Global Infrastructure a Reality". Wiley Series in Communications Networking & Distributed Systems*, 2003.

[9] "Grid Computing using .NET and WSRF.NET Tutorial," GGF11, Honolulu, June 6, 2004

[10] Hewlett-Packard Company, "HP Virtualization: Computing Without Boundaries or Constraints, Enabling an Adaptive Enterprise," *HP white paper, 2002*, Hewlett-Packard

[11] M. Chetty and R. Buyya, "Weaving Computational Grids: How Analogous Are They With Electrical Grids?" IEEE Computing in Science and Engineering, July/August 2002

[12] J. C. R. Licklider and R. W. Taylor, "The Computer as a Communication Device," *Sci. Technology*., 21–31, April (1968).

[13] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", *Globus Project*, 2002.

