ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 5, Issue 2, pp: (24-29), Month: April - June 2017, Available at: www.researchpublish.com

Grid Computing: A Survey

Manjot Kaur Bhatia

Professor, Department of Information Technology, Jagan Institute of Management Studies, Delhi

Abstract: **A single parallel system, such as a cluster, with restricted number of resources may not be able to deliver appropriate computing power to applications demanding high computation power. One solution to this problem is to collect and utilize distributed computing resources that are owned and maintained by different institutions, labs or computing centres. The distributing computing power when connected transparently to meet the requirement of computation intensive applications has raised a new approach termed as grid computing. Grid computing is broadly considered as a concept of enormous potential in both industry and academia. Grid computing environment consists of geographically distributed resources like processors, storage systems and high speed networks. Implementation of grid computing environment involves several issues like security, resource management, data management, information services and scheduling. Some of which are of immediate concern and some are long term issues.**

Keywords: **Grid, Distributed Computing, Grid Computing.**

1. INTRODUCTION

With the emerging development of internet world, high speed networks and powerful computers can be easily used by the scientists and engineers for complex computation. It also helps the people in managing large amount of information. Scientists can utilize processing capability of the available computing resources to extend their experiments on multiple parameters and carry out simulations with large data set. The results of experiments and simulations can be shared immediately anywhere in the world using high speed networks. The two essential and important requirements of high performance and parallel computing environment are more computational power than we have at any given point and the simplest, yet most complete way to use the available resources. These resources can be utilized effectively by means of grid. This environment allows processes to share distributed resources, services and data. The environment that shares resources and data from different administrative domains raises many challenges related to data and resources at all levels- conceptual and implementation models, managing, distribution, application formulation and development, programming system, processing and storage, infrastructures and services, network and security. These challenges have encouraged the researchers to create a new paradigm known as “Grid”. This is similar to the electric power grid that provides consistent, pervasive, dependable, transparent access to electric power irrespective of its source. The popularity of the Internet and the availability of powerful computers and high-speed networks as low-cost commodity components are changing the way we use computers today. These technical opportunities have led to the possibility of using geographically distributed and multi-owner resources to solve large-scale problems in science, engineering, and commerce. The research in this area emerges with a new concept termed as grid computing.

2. GRID COMPUTING

Computers were initially used for numeric computations and then for non numeric computations. As the computers are connected by wireless or wired networks, the idea of utilizing computational capacity of the idle CPUs was raised in 1970s. The invention of web by Tim Berners-Lee in 1989 was a key revolution in computing as an easy way of sharing information. World Wide Web with help of some protocols provided the means of accessing information online crossing the limits of locations. Scientific and engineering computational applications need high performance computing environment. Today, biologists have calculated the structures of complex assemblies of molecules and screen thousands of drug candidates. High performance computing run advanced application programs efficiently on thousands of powerful computers using parallel computing. Parallel computing is the simultaneous use of multiple computing resources to solve a complex computational problem. The accessible computational resources, workstations and high-speed networks can be used as products for solving high performance computing (HPC) applications using parallel processing. With distributed computing, multiple computers at remote locations communicate and work together to achieve a goal. Hence, parallel and distributed computing made it possible to aggregate and use computing power from numerous computing resources. However, these advances cannot match constantly increasing computational demands. Emerging applications, such as high resolution protein structure prediction (Bradley, 2005 and Socolich, 2005) used in protein folding, and molecular dynamic simulation (Rapaport, 2004) used in computational physics, constantly demand high processing capability. Though, a single parallel system, such as a cluster, with restricted number of resources may not be able to deliver sufficient computing power to those computation-intensive applications. One solution to this problem is to collect and utilize distributed computing resources that are owned and maintained by different institutions, labs or computing centres. The distributing computing power when connected transparently to meet the requirement of computation intensive applications has raised a new approach termed as grid computing. A grid computing environment facilitates the sharing, selection and aggregation of a wide variety of resources distributed across multiple organizations. Resources include computing devices, clusters, storage systems, specialized devices and networks devices. Generally, grid computing allows its users to divide and spread large computations across multiple machines, access to distributed storage systems and provides association possibilities within virtual organizations (Humphrey and Thompson, 2002). Grid allows the simultaneous use of large numbers of resources from multiple administrative domains with dynamic requirements, complex communication structures and strict performance requirements (Foster et. al., 1998). Resource sharing allows to combines the capacity of multiple resources in accomplishing a single goal and provides a mean to utilize the power of idle resources.

2.1. Grid computing as computational grid:

Grid computing is defined as "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities." (Foster, Kesselman, 1999). Grid computing (Krauter et. al., 2002) combines computers from multiple administrative domains to reach a common goal and to solve a single task. It is analogous to the power grid (Chetty et. al., 2002). Buyya et. al., 2002 defined grid as 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 user's quality-of-service requirements. The resources, information and files are shared for solving complicated problems within controlled and secure conditions. The Grid computing environment is formed with dynamic grouping of individuals, groups or organisations and is called virtual organisation. Conditions and policies for sharing resources and files are defined by these virtual organisations. The concept of virtual organisation is the key to Grid computing. It is defined as a dynamic set of individuals and/or institutions defined around a set of resource-sharing rules and conditions (Foster, Kesselman, & Tuecke, 2001).

3. GRID COMPUTING ENVIRONMENT

The development of grid environment was a big challenge for the researchers and developers. Grid environment should be able to provide access to major computing resources independent of the geographic locations. It is the back bone for the success of various grid applications. This environment is a combination of number of activities and resources involved in the execution of activities.

i Security: Grid computing is a form of distributed computing in which resources are geographically distributed and owned by different individuals with different technologies. This distributed environment allows sharing of geographically distributed heterogeneous computers and resources. Protecting the security policies of different domains adds complexity to the security scheme of the grid environment. Security is important for data integrity and confidentiality, user authentication and authorization to access resources. It is difficult to run a centrally managed security system. The Grid Security Infrastructure (GSI) provides a single sign-on, run anywhere authentication service with support for local control over access rights and mapping from global to local identities.

ii Information services: Information services mainly concentrate on providing important information on the resources of grid computing infrastructure. The information provided by these services depends completely on the information providers and include information on availability, capacity, and utilization of the resource. This information of each resource depends on the representation and the characteristics of that specific resource. Higher level indexing or data collection services are required to change these resource-specific data into database of valuable information on resources for the end user.

iii Resource management and monitoring: In grid computing environment resources are geographically distributed and owned by different individuals with different technologies that raises the issue resource management in grid computing. Users can access and utilize the resources of multiple domains participating in the grid network. The resource management system includes functions for resource discovery, resource state, resource fault tolerance capacity, resource monitoring and various service-level management activities. The resource management area facilitates in the selection of the correct resource for the jobs submitted by the users satisfying service-level requirement of the jobs. Resource management system helps the job scheduling system in identifying and allocating suitable resources to the jobs. Example: One of the notable resource management systems is the Grid Resource Allocation Manager (GRAM) in the Globus toolkit. Globus toolkit is popular grid middleware that provides set of tools for constructing a grid, covering security measures, resource location, resource management, communications and so on.

iv Data management: Data is an important asset in a grid computing environment. The movement of data in a geographically distributed system need to be done using secure data transfer protocols. Data management component should provide secure, reliable, efficient data transfer to and from the respective resources, schemes for replication and caching facilities and the ability to register, locate, and manage multiple copies of datasets.

v Load Balancing: To avoid processing delays of jobs and over loading of resources, scheduling system used by grid schedulers for job scheduling should include the feature load balancing in association with and resource managers. The balancing of load between available resources involves partitioning of jobs, identifying the resources, and queuing of the jobs. It can also implement features for advance reservations of resources, as well as running multiple jobs in parallel. Load balancing can also handle failure detection and management and can reschedule the job to other resources if required.

vi Scheduling: Grid computing is a high performance computing environment to solve larger scale computational demands. Grid computing contains resource management, task scheduling, security problems, information management and so on. Task scheduling in an important aspect of distributed computing. As grid computing is a form of distributed computing with heterogeneous resources working in a shared environment with no central control. The main aim of grid scheduling is to increase the system throughput and to satisfy the job requirements from the available resources. From the point of view of scheduling systems, a higher level abstraction for the grid can be applied by ignoring some infrastructure components such as authentication, authorization, and resource discovery and access control.

4. GRID ARCHITECTURE

The grid computing environment needs components to provide the above listed functionalities required to carry out various scientific, engineering and business applications. To provide these services, the grid infrastructure must develop the necessary Software Development Kits (SDKs), Application Programming Interfaces (APIs) and protocols (Foster, 2001). The infrastructure should be open and build on top of available standards such as the Transmission Control Protocol/Internet protocol (TCP/IP) and the Lightweight Directory Access Protocol (LDAP), and it should not require the replacement of existing site policies, operating systems, or network protocols. It must not enforce programming paradigm, language, or tools. It should also protect site autonomy, not compromise existing security and be fault tolerant with no single point of failure (Baker et. al., 2002). To fulfil these requirements, the hour glass model represents a good example for both the Grid and the Internet (National Academy Press, 1994). In 2001 Foster proposed layered grid architecture as shown in Figure 1.1, in which various grid components are arranged into five layers:

(i) Application layer (ii) Collective Layer (iii) Resource Layer (iv) Connectivity Layer (v) Fabric Layer

Each layer in grid architecture builds on the capabilities and services provided by lower layer but components within same layer share common characteristics and interact with each other. Foster followed the principle of hourglass model to specify various layers of grid architecture. The neck of the hourglass includes Resource and Connectivity layers. The protocols at these layers facilitate the sharing of various types of resources defined at the Fabric layer and can also be used in application development at the Collective layer.

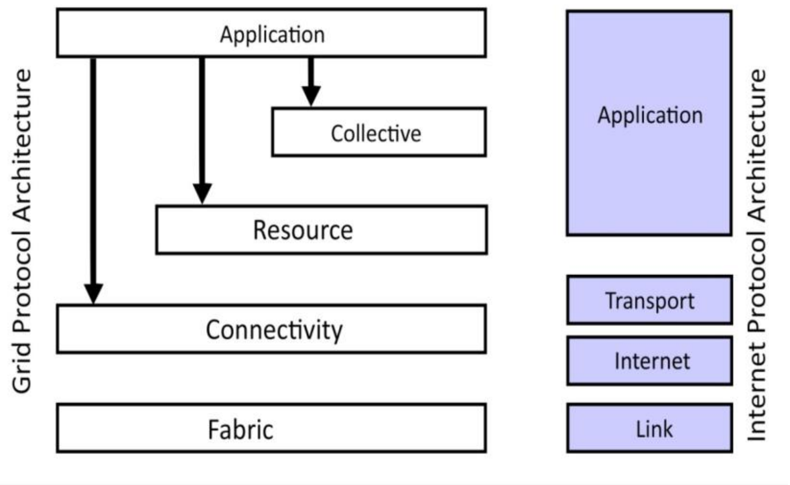

Figure1.1. Layered grid architecture (Foster et. al., 2001)

i Fabric layer: It includes various distributed resources such as computational resources, storage systems, network resources, catalogs and scientific instruments. The resource can be a logical entity like distributed file system, database systems or distributed computer pool. These resources are shared by the various users or applications with the help of grid protocols. The computational resources represent multiple architectures such as clusters, supercomputers, servers or ordinary PCs which run on variety of operating system like Unix or Windows (Buyya and Venugopal, 2005). Fabric layer components implement mechanism to permit resource-specific operations for specific local resources.

ii Connectivity layer: It defines core communication and authentication protocols. These protocols are required for grid-specific network transactions. Communication protocols facilitate the exchange of data between Fabric layer resources. Authentication protocols provide security mechanism used to validate the identity of users and resources. Communication requirements include transport, routing, and naming. Butler(2000) identified some important features that should be in authentication protocol of virtual environment such as: single sign on, delegation, user based trust relationships, Integration with various local security solutions. Grid security solutions should also provide flexible support for communication protection.

iii Resource layer: The resource layer based on the connectivity and authentication protocols control the access of resources. Resource layer defines protocols for the secure initiation, publication, discovery, negotiation, monitoring, control and account of sharing operations on individual resources. The resource layer calls the fabric layer functions to access and control local resources. Resource layer protocols are concerned completely with individual resources and do not consider issues of global collection. There are two types of resource layer protocols (Foster, 2001):

a. Information protocols: used to obtain information about the structure and state of a resource, for example, its configuration, current load, and usage policy.

b. Management protocols are used to negotiate access to a shared resource, specifying, for example, resource requirements (including advanced reservation and quality of service) and the operation(s) to be performed, such as process creation, or data access.

iv Collective layer: The main purpose of collective layer is to schedule and implement multiple resources as collections. It implements various functionalities and services for the shared resources. Foster(2001) explain these functionalities and services as:

1. Directory services: Services that allow grid users to discover the existence of Grid resources. This service allows its users to search for resources by name, type, availability or load.

2. Coallocation-allocation, scheduling, and brokering services: Grid users can request for the allocation of one or more resources for a specific purpose or for the scheduling of tasks on the appropriate resources. Examples include Condor (Raman et.al., 1998), Globus(Czajkowski,1998), NetSolve(Casanova et. al., 1999), Nimrod/G(Buyya et. al., 2000), AppLeS(Casanova et. al., 2000).

3. Monitoring and diagnostics services: are used to monitor resources for overloading, failure and external attack.

4. Data replication services: Helps to manage and improve the data access performance of storage resources.

5. Grid-enables programming systems: This system facilitates familiar programming models to be used in grid environments to deal with subjects like resource discovery, resource allocation, security and so on.

6. Software discovery services: Discover and select the best software implementation and execution platform based on the parameters of the problem being solved (Casanova, 1999).

7. Community authorization servers: These servers implement community policies leading to resource access and generate ability for the community members to access community resources.

8. Collaboratory services: These services help in coordinated exchange of information within the users of large communities.

5. CONCLUSION

High performance computing needs hardware and software with high end computational capabilities to meet the processing demands of complex scientific problems. A computational grid shares processing power as the main computing resource among its nodes and provides dependable, consistent, pervasive and inexpensive computational facilities to tasks. It provides computational power to process large scale jobs, to better utilize resources and to satisfy requirement for instant access to resources on demand. The development of grid computing environment raises certain issues such as security, reliability, resource management, resource aggregation, resource discovery, resource selection, task scheduling, task-schedule efficiency and more.

REFERENCES

[1] Bradley, P., Misura, K. and Baker, D. (2005) "Toward High-Resolution de Novo Structure Prediction for Small Proteins". Science, 309(5742):1868–1871.

[2] Rapaport, D. C.(2004). The Art of Molecular Dynamics Simulation. Cambridge University Press.

[3] Humphrey, M., & Thompson, M. R. (2002). Security implications of typical grid computing usage scenarios. Cluster Computing, 5(3), 257-264.

[4] Foster, I., Kesselman, C., Tsudik, G., & Tuecke, S. (1998, November). A security architecture for computational grids. In Proceedings of the 5th ACM conference on Computer and communications security (pp. 83-92). ACM.

[5] Foster, I., & Kesselman, C. (1999, Ed 2004), Chapter 4 of The Grid 2: Blueprint for a new computing infrastructure, Morgan-kaufman.

[6] Foster, I., Kesselman, C., & Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. International journal of high performance computing applications, 15(3), 200-222.

[7] Baker, M., Buyya, R., & Laforenza, D. (2002). Grids and Grid technologies for wide-area distributed computing. Software: Practice and Experience, 32(15), 1437-1466

[8] Buyya, R., & Venugopal, S. (2005). A Gentle Introduction to Grid Computing and Technologies. CSI Communications, 29 (1), Computer Society of India (CSI).

[9] Butler, R., Welch, V., Engert, D., Foster, I., Tuecke, S., Volmer, J., & Kesselman, C. (2000). A national-scale authentication infrastructure. Computer, 33(12), pp. 60-66.

[10] Raman, R., Livny, M., & Solomon, M. (1998, July). Matchmaking: Distributed resource management for high throughput computing. In High Performance Distributed Computing, 1998. Proceedings. The Seventh International Symposium on (pp. 140-146).

[11] Czajkowski, K., Foster, I., Karonis, N., Kesselman, C., Martin, S., Smith, W., & Tuecke, S. (1998, January). A resource management architecture for metacomputing systems. In Job Scheduling Strategies for Parallel Processing (pp. 62-82). Springer Berlin Heidelberg.

[12] Casanova, H., Berman, F., Obertelli, G., & Wolski, R. (2000, November). The AppLeS parameter sweep template: User-level middleware for the grid. In Supercomputing, ACM/IEEE 2000 Conference (pp. 60-60). IEEE.

[13] Buyya, R., Abramson, D., & Giddy, J. (2000, May). Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid. In High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International Conference/Exhibition on (Vol. 1, pp. 283-289). IEEE.

[14] Casanova, H., Dongarra, J., Johnson, C. and Miller, M.(1999). Application-Specific Tools. In Foster, I. and Kesselman, C. eds. *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, pp.159-180.

[15] Krauter, K., Buyya, R., & Maheswaran, M. (2002). A taxonomy and survey of grid resource management systems for distributed computing. Software: Practice and Experience, 32(2), 135-164.

[16] Buyya, R. (2002). “Economic-based Distributed Resource Management and Scheduling for Grid Computing”, Ph.D. Thesis, Monash University, Melbourne, Australia.

[17] Chetty, M., & Buyya, R. (2002). Weaving computational Grids: How analogous are they with electrical Grids?. Computing in Science & Engineering, 4(4), 61-71.