Grid Computing and Grid Scheduler

Vaishali Gupta
Department of Computer Science and Engineering
Geeta Institute of Management & Technology, Kurukshetra, Haryana, India

Pankaj Dev Chadha
Department of Computer Science and Engineering
Geeta Institute of Management & Technology, Kurukshetra, Haryana, India

Abstract— The term "Grid computing", is a new model for distributed and resource oriented computing, enables the selection, sharing and aggregation of geographically a parted heterogeneous resources for solving large-scale problems in commerce, engineering and science. Grid computing has been a paradigm in information technology since past few years. Grid scheduling environment is the arrangements of the machines in the course of find it fast. To find a resource in geographically distributed heterogeneous computing systems and making scheduling decisions, taking into consideration eminence of service. Allocation of tasks to a large number of resources in a grid computing environment presents more difficulty than in network computational environments. This paper presents the key requirements that a Grid system needs to support for effective task allocation.

I. INTRODUCTION TO GRID COMPUTING

GRID technology is used for coordinating large scale resource sharing and problem solving among various autonomous groups. Grid computing are currently distinct from other major technical trends such as internet [2], enterprise distributed networks and peer to peer computing. It has some embracing issues in data management, QoS, resource allocation, scheduling accounting and performance.

Various user communities built the grid environment to offer a good infrastructure which helps the members to solve their specific problems which is called a grand challenge problem. Grid computing is a collaboration of computers that is infrastructure involving, databases & network resources available, to perform manipulation of large and intensive scale data set problems [1]. The hike in the complexities of computational problems in modern era of science and technology forced the engineers and scientists to cross the organizational boundaries to get desired data manipulation.

A grid network is consisted of different types of resources owned by different and typically independent organizations which results in heterogeneity of resources. Because of this reason, the grid based services and applications experience different resource behaviours.

Selection and sharing of resources in grid computing which can be represented by Fig.1 [4]:

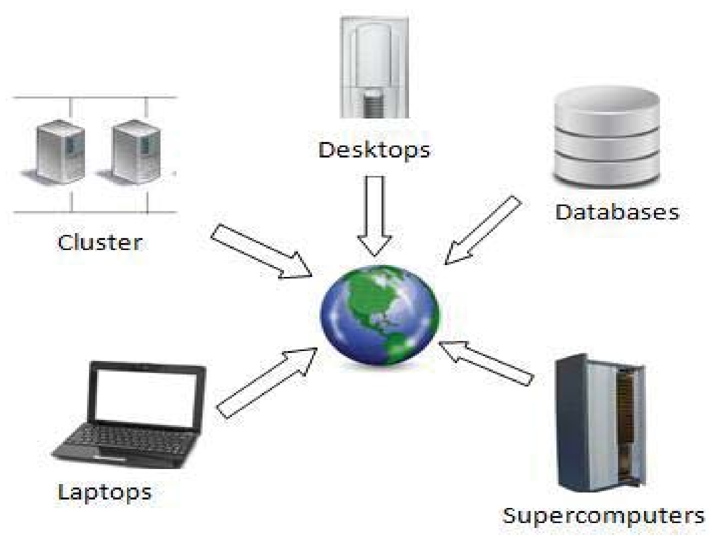

Fig. 1 Grid computing

Types of grid:

On the basis of usage grid computing, it can be divided into different types:
Collaboration grid: With the advancement of network hardware resources and internet services, demand for better collaboration is increased. So, desired collaboration is best possible with these kinds of grids [5].
Computational grids: These types of grids provide secure access to computational resources, these are sufficient to perform processing of computational problems which requires high computing power machines.
Utility Grid: Here, not only CPU cycles are shared, but also other soft wares and special peripherals like sensors are shared.
Data grid: Data and computation over that data – these are two things. Data grid provides the support for data storage and other data related services like handling, data discovery, publication etc.
Network grid: With poor network communication one can't utilize machines optimally even if we have computational machines with enough computational power as a part of grid. Network grid provides high performance communication using data caching among nodes and speed-up communication with one another cache nodes acting as router.

II. ARCHITECTURE

Grids started off in the mid-90s to address large-scale computation problems using a network of resource-sharing commodity machines that deliver computation power affordable by supercomputers and large dedicated clusters at that time.
The major motivation was that these high performance computing resources were hard to get access to and expensive so the starting point was to use the federated resources that could comprise compute, storage and network resources from multiple geographically distributed institutions, and such resources are generally heterogeneous and dynamic [7].
In grid computing infrastructure resources belong to and come from physically scattered administrative domains to collectively provide various resources (data, computing, and network) to the users. In a grid, computing nodes might not be placed at common physical location but may be independently operated from different locations [6]. Each computer on the grid is a distinct node. Collection of servers clustered together to work out a common problem form a grid network. The computers joined to form a grid network may have different hardware and operating systems. Grid consists of a layered architecture model presented by Fig.2 [14].

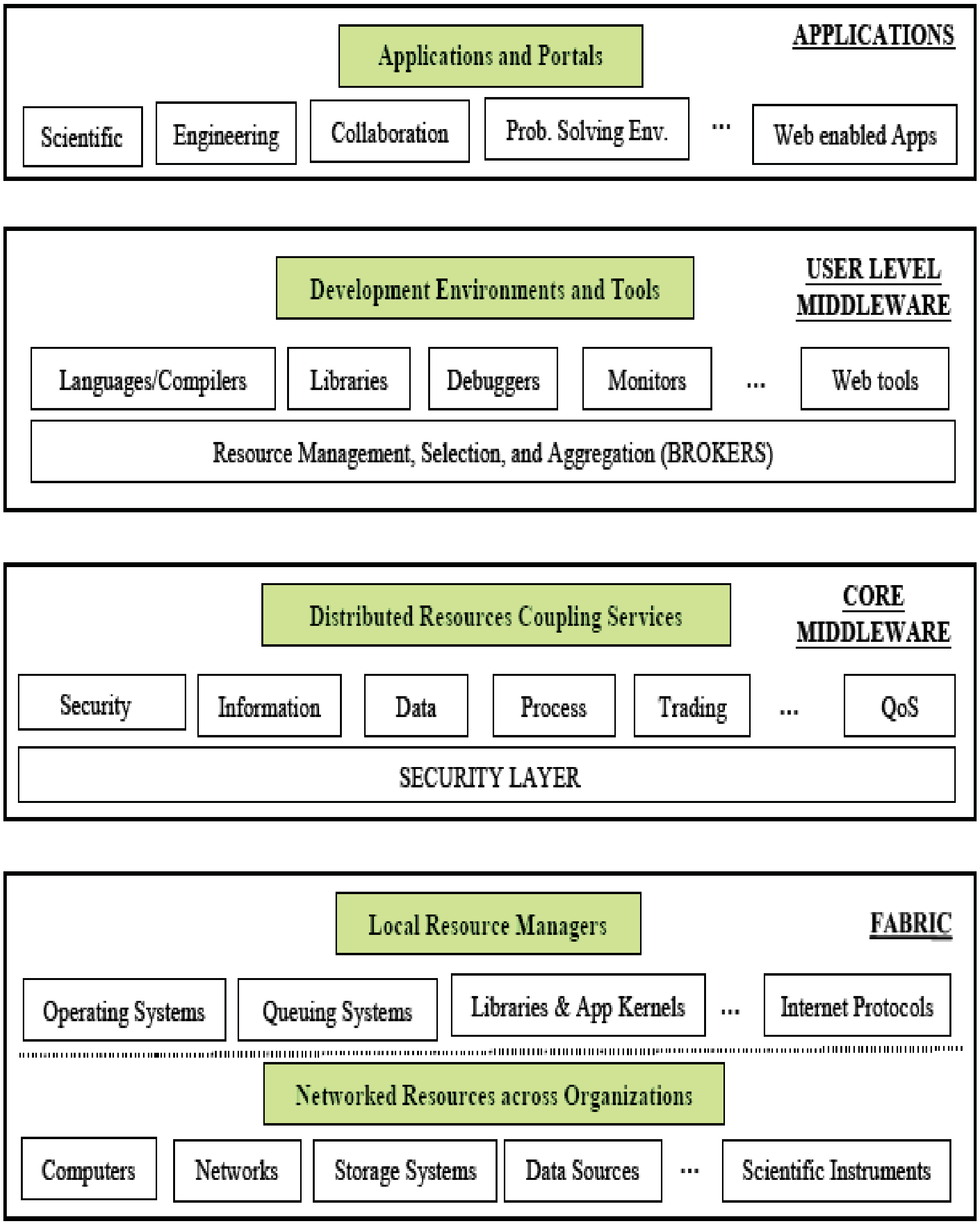

Fig. 2 Grid computing Architecture

Grid Fabric: This consists of all the globally distributed resources that are accessible from everywhere on the Internet. These resources can be computers (such as PCs, SMPs, clusters) running a variety of operating systems (such as UNIX or Windows), as well as resource management systems such as PBS (Portable Batch System) or SGE (Sun Grid Engine), LSF (Load Sharing Facility), storage devices, Condor, databases and special scientific instruments such as a particular heat sensor or radio telescope.
Core Grid Middleware: This offers core services like storage access, remote process management, information registration and discovery, co-allocation of resources, security and aspects of Quality of Service (QoS) such as trading and resource reservation.
User-Level Grid Middleware: It includes programming tools, application development environments and resource brokers for managing resources and scheduling application tasks for execution on global resources.
Grid Applications and Portals: Grid applications are typically developed using Grid-enabled utilities and languages like MPI (message-passing interface) or Nimrod parameter specification language. An example

application, such as a grand-challenge problem or a parameter simulation would require access to remote data sets, computational power and may need to interact with scientific instruments. Grid portals offer Web-enabled application services, where the users can collect and submit results for their jobs on remote resources through the Web.

III. GRID SCHEDULER

Grid environment is a distributed environment consists of different processors with numerous capabilities [8]. One of the most common issues in resources is the problem of job scheduling. In most of the works accomplished in this field is done to find an appropriate scheduling which minimizes the total execution time. To select schedulers that minimizes the mean length of queues and also the mean waiting time of processes in queues. The efficient scheduling of jobs on Grid systems is clearly critical because long wait time or queue's long length leads to grate waste of computational resources and also leads to finalization of deadline of some processes [4].

This technology makes it possible to share, select and aggregate resources that are geographically distributed like super computers, databases, data sources and specialized devices belonging to different organizations. However, there are certain issues in the area of grid computing that need to be addressed [9]. Allocation of resources to a large number of jobs in a grid computing environment presents more difficulty than in LAN computational environments.

The grid scheduler's aim is to allocate the tasks to the available resources. In large scale computing systems such as grid computing systems, there are often large amounts of resources available to be used for computing jobs. Since these resources can cost up to millions of dollars maximizing their utilization is an important problem. Scheduling in a grid computing system is not as simple as scheduling on a many machine because of several factors. These factors include the fact that grid resources are sometimes used by paying customers who have interest in how their jobs are being scheduled. Also, grid computing systems usually operate in remote locations so scheduling tasks for the clusters may be occurring over a network. It is because of these reasons that looking at scheduling in grid computing is an interesting and important problem to examine [3].

Another significant factor is the load balancing of available resources in computational grids. Applications differ in their characteristics and their demand for resources. Hence the most significant technical challenges in the Grid are efficient and application adaptive management of resources and scheduling. A lot of applications that are beyond the scopes of distribution and resource sharing now use grid computing. Scheduling of grid resources alone renders the distributed resources useful. To achieve high performance grid computing, we need to make use of optimal schedulers as opposed to poor schedulers that gives contrasting performance [10]. A variety of constraints on different problems are to be satisfied by the scheduling chosen. Specific knowledge of the underlying grid infrastructure is put to use by the existing approaches for grid scheduling by queuing systems and ad hoc schedulers in order to achieve an efficient allocation of resources.

The grid scheduler finds out the better resource of a particular task and submits that job to the selected systems. The grid scheduler does not have control over the resources and also on the submitted jobs. The selection of resource is based on the prediction of the computing power of the resource. So, lots of problems are needed to be solved in this area. The grid scheduler must allocate the tasks to the resources efficiently. There are some policies in which resource can be managed:

- *Location Policy* is the policy that affects the finding of a suitable device for migration. The common technique followed here is polling, on a broadcast, random, nearest-neighbour or roster basis.
- *Transfer policy* is that which determine whether a device is suitable for participating in a process-migration [13]. One common technique followed is the threshold policy, where a device participates in a negotiation only when its load is less than (in destination-initiated algorithm) or greater than (in sender-initiated algorithm) a threshold value.
- *Selection policy* is the policy that deals with the selection of the process to be migrated. The common factors which must be considered are the cost of migration (communication time, memory, computational requirement of the process, etc.) and the expected gain of migration (overall speedup of the system, etc.).
- *Information policy* is that component of the algorithm that decides what, how and when the information regarding the state of the other devices in the system is gathered and managed. They can be grouped under demand-driven, periodic, or state-change-driven policies.

IV. GRID APPLICATIONS

Grid makes a utilized use of resources, optimizing the CPU cycles which otherwise would have been wasted. With this the users can get extra computation resource and thus can process their large-scale computational problems thus solving a complex problem to the computational level of a supercomputer. Grid Resources can be used to solve complex problems in many areas like high-energy physics, biophysics, nuclear simulations, weather monitoring and prediction, financial analysis, chemical engineering etc.

Various areas in which grid computing may be used are:

Engineering Design and Automation: Computational aerodynamics, artificial intelligence and automation, finite-element analyses, remote sensing applications, pattern recognition, computer vision, image processing, etc. [12].

Medical, Military and Basic Research: Polymer chemistry, medical imaging, nuclear weapon design, problem of quantum mechanics, etc.

Predictive Modelling and Simulation: Flood warning, socio-economic and government use, numerical weather forecasting, astrophysics (Modelling of Black holes and Astronomical formations), semiconductor simulation, Oceanography, human genome sequencing, etc.

Energy Resource Exploration: Plasma Fusion power, seismic exploration, nuclear reactor safety, reservoir modelling, etc.

Visualization: computer-generated graphics, films and animations, data visualization, etc.

Some advantages of grid computing are:

- Seamless and secure access to large number of geographically distributed resources and reduction in average job response time may occur but an overhead of limited network bandwidth and latency exists.
- Grid provides users around the world with dynamic and adaptive access to unparalleled levels of computing. With the infrastructure provided by the Grid, scientists are able to perform complex tasks, integrate their work and collaborate remotely [11].
- Grids can lead to savings in processing time and it is responsible for efficient, effective, economic utilization of available resources and increased availability and reliability of resources.
- It provides shared access (by multiple users) to large amounts of data, improved methods for collaborative work and parallel processing capability. Grid also reduces the time of result.

V. CONCLUSION

In this grid and its architecture has been studied briefly and the requirements that a grid system needs to support for effective task allocation has been discussed. In future an effective task allocation algorithm will be designed to allocate the tasks to the grid resources efficiently.

REFERENCES

[1] Congfeng Jiang, Cheng Wang, Xiaohu Liu and Yinghui Zhao, “A survey of Job Scheduling in Grid”, 2007.

[2] Luis Ferreira, Fabiano Lucchese, Tomoari Yasuda, Chin Yau Lee, Carlos Alexandre Queiroz, Elton Minetto, Antonio Mungioli First Edition (April 2005). Grid Computing in Research and education.

[3] D.Maruthanayagam, Dr.R.Umarani, “Enhanced Ant Colony Algorithm for Grid Scheduling” Int.J.Comp.Tech.Appl, Vol 1 (1) 43-53

[4] Poonam Dabas, Anoopa Arya, “Grid Computing: An Introduction” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013.

[5] Phillipa Sessini, “Scheduling in Grid Computing Systems”, CPSC 531 Term Project, University of Calgary, Alberta,Canada: August 10, 2005.

[6] Shruti N. Pardeshi, Chitra Patil,Snehal Dhumale, “Grid Computing Architecture and benefits” International Journal of Scientific and Research Publications, Volume 3, Issue 8, August 2013 1 ISSN 2250-3153.

[7] “Introduction to Grid Computing” by Bart Jacob Michael Brown, Kentaro Fukui, Nihar Trivedi

[8] Manish Parashar, Craig A Lee, member, IEEE”Grid Computing: Introduction and Overview” I. Platform computing,”The politics of Grid: Organisational politics as a barrier to implementing grid computing”, 2004.

[9] M. L. Bote-Lorenzo, Y. A. Dimitriadis and E. Gómez-Sánchez (2004)

[10] Grid Characteristics and Uses: A Grid Definition. Grid Computing.(Ed.). Pgs. 291-298. Springer Berlin / Heidelberg.

[11] I. Foster (2003) The Grid: A New Infrastructure for 21st Century Science. Grid Computing. G. F. Fran Berman, Tony Hey (Ed.). Pgs. 51-63.

[12] Volker Sander, “Networking Issues for Grid Infrastructure”, GFD-I.037, Nov, 22, 2004.

[13] F. Xhafa, J. Carretero, L. Barolli, and A. Durresi.“Immediate mode scheduling in grid systems” International Journal of Web and Grid Services, 3(2):219–236, 2007.

[14] H. Shan, L. Oliker, and R. Biswas, “Job Superscheduler Architecture and Performance in Computational Grid Environments,” in Proceedings of ACM/IEEE Conference on Supercomputing, pp. 44-48, November 2013.