Grid Architecture for Scheduling and Load Balancing – An Assessment

B.Priya MCA, M.Tech
Asst Prof , MCA Dept,
Sri Sai Ram Engineering College, Chennai.
priya.mca@sairam.edu.in

Dr.T.Gnanasekaran, Ph.D,
Professor and Head,Department of IT,
R.M.K College of Engineering and Technology,
Chennai

Abstract-Grid Computing emerged as a wide scale distributed system to offer dynamic coordinated resources sharing and high performance computing. Grid coordinates the resources that are not subject to centralized control. It uses standard open, general purpose protocols and its interfaces. Grid delivers non-trivial qualities of service such as the response time, throughput, availability and security. Grid computing is being adopted in various areas from academic, industry research to government use. Grids are becoming platforms for high performance and distributed computing. Grid computing is the next generation IT infrastructure that promises to transform the way organizations and individuals compute, communicate and collaborate. The goal of Grid computing is to create the illusion of a simple but large and powerful self-managing virtual computer out of a large collection of connected heterogeneous systems sharing various combinations of resources. Scheduling is a set of rules and policies that control the order by which various jobs are executed in a system. Load balancing deals with the way by which the various tasks are assigned to the resources thereby improving the system performance. Scheduling of various tasks to the resources in a Grid environment is an active research area. Resources are dynamic in nature so the load of resources varies with change in configuration of Grid so the Load Balancing of the tasks in a Grid environment can significantly influence Grid's performance. A poor scheduling policy may leave many processors idle while a clever one may consume an unduly large portion of the total CPU cycles. In order to utilize the resources efficiently and to satisfy the requirement of the users, several scheduling algorithms have been proposed by researchers for various applications. This paper deals with the overview of the Grid Computing concepts, the various standards associated with it, the terms associated with scheduling and load balancing and the tools associated with Grid Scheduling. The paper also deals with the review of the various grid scheduling algorithms along with the proposed model for scheduling,

Keywords-Computational Grid, Grid Scheduler (GS), Load balancing, OGSA, XML

I.INTRODUCTION

Grid is defined as "A type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous and heterogeneous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements"[1]. A grid is a system that coordinates resources that are not subject to centralized control using standard, open, general purpose interfaces and protocols to deliver non-trivial qualities of service. Grid Computing is an emerging computing model that treats all resources as a collection of manageable entities with common interfaces to such functionality as lifetime management, discoverable properties and accessibility via open protocols.

Grid computing appears to be a promising trend for three reasons:

1. Its ability to make more cost-effective use of a given amount of computer resources,
2. As a way to solve problems that cannot be approached without an enormous amount of computing power, and
3. It suggests that the resources of many computers can be cooperatively and perhaps synergistically harnessed and managed as a collaboration toward a common objective.

The rest of the paper is organized as follows: Section 1.1 to 1.3 illustrates the types, application areas and advantages of using Grid. Section 2 deals with the various standards associated with the Grid. Section 3 deals with the various Grid Scheduling concepts. Section 4 illustrates the Load Balancing mechanism in Grid. Section 5 deals with the some of the work associated with Scheduling and Load Balancing in Grid.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

A. Types of Grids

Ideally, a grid should provide full-scale integration of heterogeneous computing resources of any type: processing units, storage units, communication units, and so on. However, as the technology hasn't yet reached its maturity, real-world grid implementations are more specialized and generally focus on the integration of certain types of resources. As a result, nowadays we have different types of grids, which we describe as follows:

1. **Computational grid** A computational grid is a grid that has the processing power as the main computing resource shared among its nodes. This is the most common type of grid and it has been used to perform high-performance computing to tackle processing-demanding tasks.
2. **Data grid** Just as a computational grid has the processing power as the main computing resource shared among their nodes, a data grid has the data storage capacity as its main shared resource. Such a grid can be regarded as a massive data storage system built up from portions of a large number of storage devices.
3. **Network grid** This is known as either a network grid or a delivery grid. Such a grid has as its main purpose to provide fault-tolerant and high-performance communication services. In this sense, each grid node works as a data router between two communication points, providing data-caching and other facilities to speed up the communications between such points.

B.Grid Application Areas

Applications with heavy use of computing resources (e.g., simulations, number crunching, the so-called grand challenge applications), applications using large information resources (e.g., multimedia databases), applications using special sub applications (e.g., visualization), and applications using special devices (e.g., expensive scanners, laboratory equipment) are candidates for Grids. Especially we mention the following important application areas :

1. Medical Applications: In diagnostics huge amounts of data are generated at one place by specialized devices. These data have to be transported to the specialists, possibly located at several locations, while the patient might be at a third location. The task of a Grid in this scenario is to prepare and transport the medical data, so that they are available at the right location at the right time [2].
2. Support for multinational enterprises: Multinational enterprises work at several locations in several time zones. Data, e.g., multimedia data from inspections, must be pre-processed and forwarded to specialists who can take decisions. Several Multimedia Applications make use of a Grid for processing media streams. Within multimedia QoS control is very important. Applications often include the handling of Digital Rights Management. E.g., multimedia data can be watermarked scrambled etc.
3. Applications from bio-informatics, seismology, meteorology, etc. are data – and computing-intensive, and need often other information resources.
4. E-governance :E-governance is the application of information and accountability in Government related tasks. It enables citizens to make the best use of automated administration processes that are accessible on-line. Grid computing is an ideal solution to these type of applications [6].
5. E-Learning : Scheduling of the various activities in an Intelligent Tutoring System (ITS) can be made possible by grid.

C. Grid Advantages

Some of the advantages of Grid Computing are listed below [3]:

1. Seamless and secure access to large number of geographically distributed resources.
2. Reduction in average job response time may occur but an overhead of limited network bandwidth and latency exists.
3. Provides users around the world with dynamic and adaptive access to unparalleled levels of computing.
4. With the infrastructure provided by the Grid, scientists are able to perform complex tasks, integrate their work and collaborate remotely.
5. Grids can lead to savings in processing time.
6. Efficient, effective, and economic utilization of available resources.
7. Increased availability and reliability of resources.
8. Shared access (by multiple users) to large amounts of data.
9. Improved methods for collaborative work.
10. Unprecedented Price-to-Performance ratio.

Implementing an e-governance solution will lower the cost of developing , deploying, managing government solutions and providing better services to the citizens. A Case study has been proposed by sumathi

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

et al[6] , which depicts the advantage of using Grid for a scenario of a citizen applying for a new water connection.

The design features that are required by a Grid to provide users with a perfect computing environment are[12]:

1. Multiple Administrative Domains and Autonomy
2. Heterogeneity
3. Scalability
4. Adaptability.

The steps necessary to realize a Grid include[12,13]:

1. The integration of individual software and hardware components in to a combined networked resource.
2. The Deployment of Low-level middleware to provide a secure and transparent access to resources.
3. The deployment of User-level middleware and tools for application development and the aggregation of distributed resources.
4. The development and optimization of distributed applications to take advantage of the available resources and infrastructure.

II.STANDARDS FOR GRID ENVIRONMENT

The various standards for Grid Environment are:

1. Open Grid Services Architecture (OGSA) [11]
2. Open Grid Services Interface (OGSI)
3. Data Access and Integration (OGSA – DAI)
4. Web Services Resource framework (WSRF)
5. Web Services Related Standards
   i. Extensible markup Language (XML)
   ii. Web Services Description Language (WSDL)
   iii. Service Oriented Access Protocol (SOAP)
   iv. Universal Description Discovery and Integration (UDDI)

III.SCHEDULING IN COMPUTATIONAL GRIDS

A task in a grid is associated with the parameters: CPU/ Memory Size, Deadline to complete the task, Priority etc. Scheduling is the processes of ordering tasks on computational resources and ordering communication between tasks. Scheduling is carried out for:

1. Shorten the Job Completion time
2. Enhance the system throughput.

A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [4]. A computational Grid environment behaves like a virtual organization consisting of distributed resources.

A Virtual Organization is a set of individuals and institutions defined by a definite set of sharing rules like what is shared, who is allowed to share, and the conditions under which the sharing takes place. A number of Virtual Organizations exist like the application service providers, storage service providers, *etc.*, but they do not completely satisfy the requirements of the Grid. The needs of the Grid range from client-server to peer-to-peer architecture, from single user to multi-user systems, from sharing files to sharing resources, *etc.* and all these in a dynamic, controlled, and secured manner. As Grid computing focuses on dynamic and cross-organizational sharing, it enhances the existing distributed computing technologies.

There are many issues in using computational grid such as:

1. How to appropriately and efficiently assign resources to tasks, generally called job scheduling.
2. Workflow Management of the applications[16]
3. Scheduling algorithms for handling Fault tolerance.

The various challenges for scheduling in Grid are:

1. All resources reside within a single administrative domain.
2. To provide a single system image, the scheduler controls all of the resources.
3. The resource pool is invariant.
4. Contention caused by the incoming applications can be managed by the scheduler according to some policies, so that its impact on the performance that the site can provide to each application can be well predicted.
5. Computations and their data reside in the same site or data staging is a highly predictable process, usually from a predetermined source to a predetermined destination, which can be viewed as a constant overhead.

There are various types of scheduling such as Static and Dynamic scheduling. The advantage of dynamic load balancing over static scheduling is that the system need not be aware of the run-time behavior of the application before execution. It is particularly useful in a system where the primary performance goal is maximizing resource utilization, rather than minimizing runtime for individual jobs [14].The taxonomy of scheduling algorithms as depicted in [19] is given below:

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

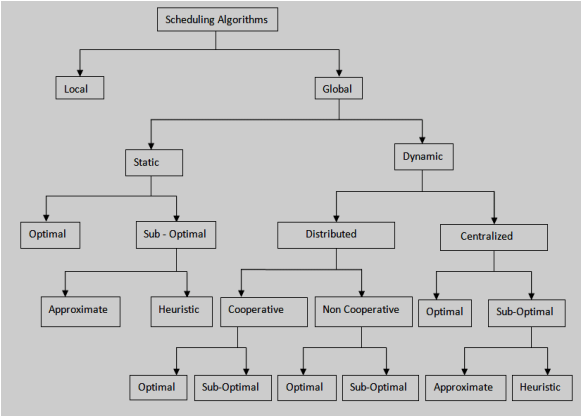

Figure 1: Taxonomy of scheduling algorithms

According to how the dynamic load balancing is achieved, there are four basic approaches [15]:

1. Unconstrained First-In-First-Out (FIFO, also known as First-Come-First- Served)
2. Balance-constrained techniques
3. Cost-constrained techniques
4. Hybrids of static and dynamic techniques

IV.LOAD BALANCING IN GRID

Load balance is also an important issue in grid environment. The purpose of load balancing is to balance the load of each resource in order to enhance the resource utilization and increase the system throughput. The main objective of load balancing methods is to speed up the execution of applications on resources whose workload varies at run time in unpredictable way. Hence, it is significant to define metrics to measure the resource workload. Every dynamic load balancing method must estimate the timely workload information of each resource. This is key information in a load balancing system where responses are given to following questions:

1. How to measure resource workload?
2. What criteria are retaining to define this workload?
3. How to avoid the negative effects of resources dynamicity on the workload; and
4. How to take into account the resources heterogeneity in order to obtain an instantaneous average workload representative of the system?

Several load indices have been proposed in the literature, like CPU queue length, average CPU queue length, CPU utilization, etc. The success of a load balancing algorithm depends from stability of the number of messages (small overhead), support environment, low cost update of the workload, and short mean response time which is a significant measurement for a user. It is also essential to measure the communication cost induced by a load balancing operation.

V. TOOLS ASSOCIATED WITH GRID SCHEDULING

The various tools associated with Grid Scheduling are:

A.OptorSim:

In the search for an optimal scheduling and replication strategy, the grid simulator OptorSim[17] was developed as part of the European DataGrid project. Simulations of various high energy physics (HEP) grid scenarios have been undertaken using different job scheduling and file replication algorithms, with the emphasis being on physics analysis use-cases. OptorSim which has the structure of EDG, includes the following elements to achieve a realistic simulated environment. These include storage resources where data can be kept, computing resources to which jobs can be sent, scheduler to decide to which resource the job has to be sent, the network which connects the sites and finally replica management.

B. GridSim:

The GridSim toolkit provides a comprehensive facility for simulation of different classes of heterogeneous resources, users, applications, resource brokers, and schedulers. It can be used to simulate application schedulers for single or multiple administrative domains distributed computing systems such as clusters and Grids. Application schedulers in the Grid environment, called resource brokers, perform resource discovery, selection, and aggregation of a diverse set of distributed resources for an individual user. This means that each user has his or her own private resource broker and hence it can be targeted to optimize for the requirements and objectives of its owner. In contrast, schedulers, managing resources such as clusters in a single administrative domain, have complete control over the policy used for allocation of resources. [3]

C.SimGrid:

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of distributed and parallel application scheduling on distributed computing platforms ranging from simple network of workstations to Computational Grids. Henri Casanova in [18] has used Simgrid for the study of scheduling algorithms for distributed application.

VI.RELATED WORK

Some of the Grid related projects are Globus, Condor and Nimrod – G. Globus[7] uses open source toolkit for building gird systems and applications. It allows sharing of computing power, databases and other tools securely online. It provides facilities for Resource monitoring, discovery, security and file management. Nimrod – G is a tool to manage the execution of the parametric studies across distributed computers.

They have been various scheduling algorithms developed for various applications. Ramya et al[5] proposed an Optimized Hierarchical Load Balancing Algorithm(OHLBA) which dynamically creates an optimal schedule to complete the jobs within minimum makespan. The Average Computational Power (ACP) for each cluster is found out to select the resources. The Average load of each resource (ALC) is found for load analysis. A threshold value is compared with the ALC to submit a job to the cluster which is in underload.

Sumathi et al[6] proposed a Hybrid Core Scheduling algorithm for Workflow management in Grid. The Hybrid Core algorithm dynamically schedules tasks of workflows to grid sites based on the performance of the sites when running previous jobs from the same workflow for E-Governance applications.

Salman Meraji et al[8] proposed a new scheduling algorithm to improve task assignment performance by schedulers through minimizing makespan amounts and maximizing resource utilization percentage and matching proximity.

Keerthika et al[9] proposes a Bicriteria Scheduling algorithm (BSA) that considers user satisfaction along with fault tolerance. The algorithm calculates the total completion time and then the fitness value based on failure rate. Deadline hit count is another new metric introduced in this work. This represents the number of tasks successfully completed within the user deadline. This is a measure of user satisfaction which is the key requirement in grid system.

Joshua Samuel Raj et al[10] proposed an enhancement of Hierarchical Load Balancing Algorithm by evaluating the cluster imbalance. This proposed Enhanced Hierrarchical Load Balancing algorithm(ALHBA), the idle condition of the cluster is evaluated while scheduling. The alogrithm then computes the Average Load of cluster and the Expected Computing Power (ECP) of each job for scheduling. The queue length of each cluster is found out. When the queue length is greater, jobs are stolen and allocated to free clusters.

VII.PROPOSED MODEL

The proposed model for scheduling and load balancing proposed by the author termed as Grid Architecture for Scheduling and Load balancing (GASLB) is shown in Figure 2:

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

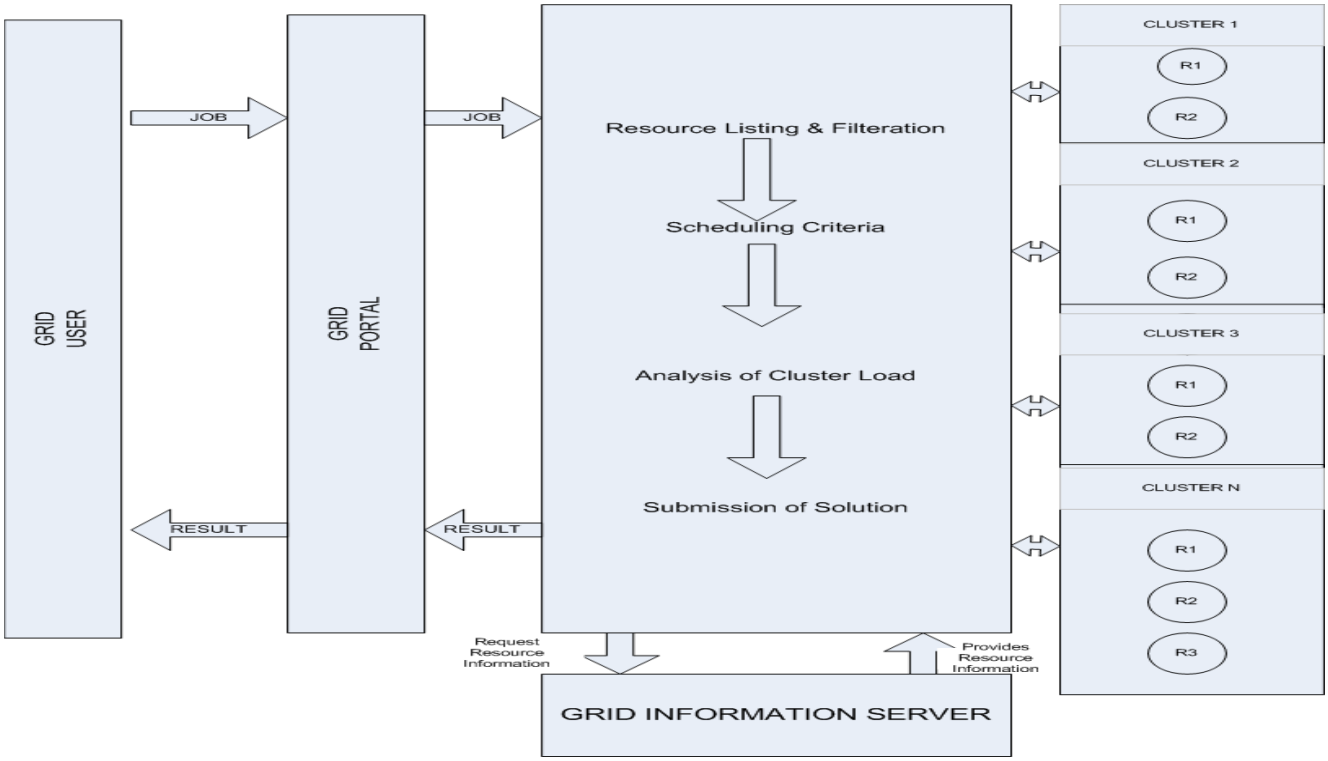

Figure 2: Grid Architecture for Scheduling and Load Balancing (GASLB)

The details of the various Grid resources are updated in a Grid Information Server(GIS). The GIS collects and predicts the resource state information such as

1. CPU Capacities
2. Memory Size
3. Network Bandwidth
4. Software Availabilities and
5. The load of a site in a particular period.

The Grid scheduler (GS), also termed as Meta Scheduler receives applications from Grid users, selects feasible resources for these applications according to acquired information from the Grid Information Service module, and finally generates application-to-resource mappings, based on certain objective functions and predicted resource performance such as:

1. User Demand
2. Communication time
3. Failure handling mechanisms and
4. Reduced Makespan (the difference between the start and end of a job)

The GS works in three phases. They are :

1. Resource Listing and Filtering
2. Resource Selecting and Scheduling according to certain objectives (i.e Resource Allocation)
3. Analyzing the Cluster Load
4. Job Submission.(i.e Job Execution)

The third phase deals with the Load balancing concept.

VIII. CONCLUSION AND FUTURE WORK

In this paper the state of the art of the Grid scheduling is reviewed. It is intended to propose a scheduling algorithm to improve the performance in an E-Governance application for effective scheduling of the various tasks.

IX. REFERENCES

[1] M. Baker, R. Buyya and D. Laforenza, Grids and Grid Technologies for Wide-Area Distributed Computing, in J. of Software-Practice & Experience, Vol. 32, No.15, pp: 1437-1466, December 2002.
[2] Ian Foster, Carl Kesselman, Jeffrey M. Nick and Steven Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

[3]Liang Fang, Aleksander Slominski, and Dennis Gannon ,Computer Science Dept, Indiana University , Web Services Security and Load Balancing in Grid Environment.
[4] Y. H. Lee, S. Leu and R. S. Chang, "Improving job scheduling algorithms in a grid environment", Future generation computer systems, (2011)May.
[5] Ramya R and Shalini Thomas. Article: An Optimal Job Scheduling Algorithm in computational Grids. IJCA Special Issue on International Conference on Communication, Computing and Information Technology ICCCMIT(1):12-16, February 2013.
[6] Constructing a Grid Simulation for E-Governance Applications Using GridSim P. Sumathi, M. Punithavalli, Journal of Computer Science 4 (8): 674-679, 2008 ISSN 1549-3636 © 2008 Science Publications.
[7] Globus website: www.globus.org
[8] Salman Meraji, M. Reza Salehnamadi, "A Batch Mode Scheduling Algorithm for Grid Computing" J. Basic. Appl. Sci. Res., 3(4)173-181, 2013, © 2013, TextRoad Publication
[9] P.Keerthika and N. Kasthuri, "An Efficient Grid Scheduling algorithm with Fault tolerance and User Satisfaction", Mathematical problems in Engineering, Volume 2013(2013), Article ID 340294.
[10] Joshua Samuel Raj, Hridya K. S and V. Vasudevan, Augmenting Hierarchical Load Balancing with Intelligence in Grid Environment , International Journal of Grid and Distributed Computing Vol. 5, No. 2, June, 2012
[11] Clovis Chapman1, Paul Wilson2, "Condor services for the Global Grid:Interoperability between Condor and OGSA", Proceedings of the 2004 UK e-Science All Hands Meeting, ISBN 1-904425-21-6, pages 870-877, Nottingham,UK,August2004http://www.cs.wisc.edu/condor/doc/condor-ogsa-2004.pdf
[12] Rajkumar Buyya, David Abramson and Srikumar Venugopal, 2005. The Grid Economy, Special Issue on Grid Computing, Proceedings of the IEEE, Manish Parashar and Craig Lee(Eds). IEEE Press, New York, USA., March 2005. PP 618-714
[13] International Journal of Grid and Distributed Computing Vol. 4, No. 3, September, 2011, A Taxonomy of Grid Resource Selection Mechanisms Adil Yousif, Abdul Hanan Abdullah, Muhammad Shafie Abd Latiff and Mohammed Bakri Bashir Faculty of Computer Science & Information System Universiti Teknologi Malaysia UTM, Malaysia
[14] B. Jacob, L. Ferreira, N. Bieberstein, C. Gilzean, J. Girard, R. Strachowski, S. Yu,Enabling applications for Grid Computing with Globus, Redbook, IBM, March 2003.
[15] H. El-Rewini, T. Lewis, and H..Ali, Task Scheduling in Parallel and Distributed Systems, ISBN: 0130992356, PTR Prentice Hall, 1994.
[16] "A Taxonomy of Workflow Management Systems for Grid Computing", Jia Yu, Rajkumar Buyya, Journal of Grid Computing, September 2005, Volume 3, Issue 3-4, pp 171-200
[17] David G.Cameron, Ruben Carvajal-Schaffino, A.Paul Millar, Caitriana Nicholson, Kurt Stockinger, Floriano Zini, "Evaluating Scheduling and Replica Optimisation Strategies in OptorSim," International Conference on Grid Computing, Proceedings of the 4th International Workshop on Grid Computing, 2003, page 52.
[18] Henri Casanova, "Simgrid: a Toolkit for the Simulation of Application Scheduling," Proceedings of the first IEEE/ACM InternationalSymposium on Cluster Computing and the Grid (CCGRID), 2001.
[19] T.Casavant, and Kuhl, "A Taxonomy of Scheduling in General – purpose Distributed Computing Systems," IEEE Trans. on Software Engineering, vol. 14, no. 2, pp. 141 – 154, Feb. 1988.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE