# Different Strategies of Load Balancing In Grid Computing

**Mr. Ramesh Prajapati, Dr. Samrat Khanna**

*Abstract*— **Grid computing is the collection of computer resources from various area or locations to achieve a common goal. Grid computing is a distributed computing that includes sharing of data and storage, computational power and resources over network across dynamic organizations. Workload and resource management are two main functions by Grid. 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. To achieve the promising potentials of tremendous distributed resources, effective and efficient load balancing are fundamentally important Load balancing enables in effective technique of resources to improve the overall performance of the system. With the increase in system size, the probability of occurrence the main goal of load balancing is to provide a distributed, low cost, scheme that balances the load across all the processors. The purpose of this paper is to review various different load balancing for the grid based distributed network and identify gaps between them. Many load balancing algorithms are already implemented which works against various issues like scalability, heterogeneity.**

***Index Terms***— Grid computing, Grid Architecture, Resource Management, Load balancing

## I. INTRODUCTION

Grid computing applies the resources of many computers in a network for a single problem at the same time - usually to a scientific or technical one that requires a large number of computer processing cycles or access to large amounts of data[1]. All Grid is a system which coordinates resources that are not subject to centralized control, using standard, open, general purpose protocols and interfaces to deliver nontrivial qualities of service[3].With the increased popularity of internet and availability of high performance computers and high speed networks as low cost commodity, it has become possible to use networks of computers as a single unified computing resource. With the growth of grid technologies, more and more companies are moving from large scale, centralized databases to databases that reside on grid-based systems. Grid technology allows organizations to use numerous computers to solve problems by sharing
computing resources. The problems to be solved might involve data processing, network bandwidth, or data storage. Workload represents the amount of work to be performed where all resources have different processing speed.

Resource management and scheduling is the strongest key grid services, but to achieve scheduling and efficient grid resource management, load balancing and task scheduling is one of the key issues that must be addressed. The technique of load balancing is to distribute workload across two or more computing nodes, in order to get maximum throughput, optimal resource utilization, minimize response time, and avoid overload. Two main aspects that have to be considered in implementing any load balancing algorithms are scalability and adaptability [6].

Grid computing is enabled by relatively high-performance computers, robust computer networks, grid management software, and the divisibility of difficult scientific problems. Together these allow a job to be subdivided and distributed to thousands or even millions of computers to calculate a solution. In Grid computing, individual users can access computers and data, transparently, without having to consider location, operating system, account administration, and other details. Grids tend to be more loosely coupled, heterogeneous, and geographically distributed. In Grid computing details are abstracted, and the resources are virtualized [2].

A typical distributed system will have a number of interconnected resources that can work independently or in cooperation with each other [13].Key characteristic of grids is to share resources (e.g. CPU cycles and network capacities)among numerous applications, therefore number of resources available to

Manuscript received July 13, 2014

**Mr. Ramesh Prajapati**, Dept. of Computer Science Engineering Rai University, Center for Research & Development Saroda, Dholka, India

**Dr. Samrat Khanna**, Dept. of Information Technology Istar, Sardar patel centre for science & Tech. V.Vnagar, India

any given application highly fluctuates over time. In this scenario load balancing plays key role for those applications which are grid enabled. To minimize this unbalancing situation (time needed to perform all tasks), the workload should be evenly distributed over all resources based on their processing speed. Essential objective of load balancing consists primarily in optimizing

the average response time of applications, which often means maintaining the workload proportionally equivalent on the whole resources of a system.

This work focuses on load balancing in a grid environment. Grid application performance is critical in grid computing environment. So to achieve high performance we need to understand the factors that can affect the performance of an application like load balancing, which is one of most important factors that influence the overall performance of application. 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. Recent research on these topics has led to the emergence of a new paradigm known as Grid computing [2].

Section II describes the Literature Survey. In Section III, we talk about Load balancing. Approaches of Load Balancing are presented in Section IV. In section V, we present the load balancing strategies. Challenges describe in section VI.we present the conclusion in section VII.

## II. LITERATURE SURVEY

Grid computing is the collection of computer resources from various locations to achieve a common goal. Grid computing is a type of parallel system which enables the dynamically selection, aggregation and distribution of geologically resources at run time depending on their user quality of self service requirement, availability, performance, cost, capability. Load balancing is very effective technique to reduce response time and to improve resources utilization, exploiting through proper distribution of the application assuming homogeneous set of nodes linked with homogeneous and fast networks, various load balancing algorithms were developed. Analyzing the past results and to improve the performance and throughput, efficient algorithms with better scheduling policies [5].

Grid Computing has emerged as a new and important field and can be used to increase the performance of Distributed Computing. Grid computing [6] has large and powerful applications of self-managing virtual computer out of a large collection of heterogeneous systems that sharing various resources which lead to the problem of load balance. The main goal of load balancing is to provide a distributed, low cost, scheme that balances the load across all the processors.

Grid Computing is emerging as wide scale distributed infrastructure that promises to support resource sharing and synchronized problem solving in dynamic and heterogeneous environment. Grid Computing is becoming a generic platform for high performance and distributed computing due to which it is being adopted in various areas like academic, industry, and research use[7]. Grid Computing has progressed a lot, yet, there are some areas of concern, like resource management, resource scheduling, load balancing and security over which, research is still in progress. Load balancing is a great challenge in dynamic and heterogeneous environment like Grids. Load balancing is a technique to enhance resources, utilizing parallelism, improve throughput, and to cut response time through an appropriate distribution of the applications. The main goal of load balancing is to provide a distributed, low cost, scheme that balances the load across all the processors. An effective and efficient load balancing algorithm is required to balance the load in Grid environment, which is a tedious task due to basic nature of Grid environment.

Load Balancing [8] is a technique to improve resources, utilizing parallelism, exploiting throughput managing and to reduce response time through proper distribution of the application. Load balancing strategies is always used for the emergence of Distributed systems. Generally there are three type of phases related to Load balancing i.e. Information Collection, Decision Making, Data Migration. Grid computing is a replica of distributed computing that uses geographically and disperses resources. To increase performance and efficiency, the Grid system needs competent load balancing algorithms for the distribution of tasks. Load balancing algorithms is of two types, static and dynamic.

In the provision of a Grid service[9], a provider may have heterogeneous clusters of resources offering a variety of services to widely distributed user communities. Within such a provision of services, it will be desirable that the clusters will be hosted in a cost effective manner. Hence, an efficient structure of the available resources should be decided upon these clusters. A static structure, adopted in classical distributed systems, where a single master node controls all resources and decides where incoming jobs should be executed, is not efficient for Grid computing. For this purpose, we propose a dynamic tree-based model to represent Grid architecture in order to manage workload. This model is characterized by three main features: (i) it is Hierarchical; (ii) it supports heterogeneity and scalability; and, (iii) it is totally independent from any physical Grid architecture. Over

the proposed model, we develop a load balancing strategy suitable for large scale, dynamic and heterogeneous environments. The proposed strategy is based on a neighborhood load balancing whose goal is to decrease the amount of messages exchanged between Grid resources. As a consequence, the communication overhead induced by task transfer and workload information flow is reduced, leading to a high improvement in the global throughput of a Grid. The first experiment results of our strategy are very

promising. In effect, we have obtained a significant improvement of the mean response time with a reduction of the communication cost. The popularity of the Internet [10] 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. Recent research on these topics has led to the emergence of a new paradigm known as Grid computing. To achieve the promising potentials of tremendous distributed resources, effective and efficient load balancing algorithms are fundamentally important. Unfortunately, load balancing algorithms in traditional parallel and distributed systems, which usually run on homogeneous and dedicated resources, cannot work well in the new circumstances.

## III. LOAD BALANCING

Load balancing is a technique to enhance resources, utilizing parallelism, exploiting throughput improvisation, and to cut response time through an appropriate distribution of the applications [14]. To minimize the decision time is one of the objectives for load balancing which has yet not been achieved. Job migration is the only efficient way to guarantee that submitted jobs are completed reliably and efficiently in case of process failure, processor failure, node crash, network failure, system performance degradation, communication delay; addition of new machines dynamically even though a resource failure occurs which changes the distributed environment [11].

Load balancing mechanisms can be broadly categorized as centralized or decentralized, dynamic or static, and periodic or non-periodic [12]. All load balancing methods are designed such as, to spread the load on resources equally and maximize their utilization while minimizing the total task execution time. Selecting the optimal set of jobs for transferring has a significant role on the efficiency of the load balancing method as well as grid resource utilization. This problem has been neglected by researchers in most of previous contributions on load balancing, either in distributed systems or in the grid environment [15].

As following Figure 1 load balancing feature can prove invaluable for handling occasional peak loads of activity in parts of a larger organization. These are important issues in Load Balancing:

• An unexpected peak can be routed to relatively idle machines in the Grid.

• If the Grid is already fully utilized, the lowest priority work being performed on the Grid can be temporarily suspended or even cancelled and performed again later to make room for the higher priority work.

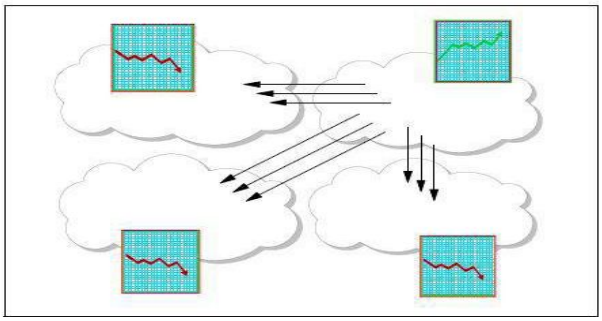

Figure 1 Job Migration [3]

## IV. LOAD BALANCING APPROACHES

Load balancing problem has been discussed in traditional distributed systems literature for more than two decades. Various algorithms, strategies and policies have been proposed, implemented and classified [16].Algorithms can be classified into two categories: static or dynamic.

### *(a) Static Load Balancing Algorithm*

It allocate the tasks of a parallel program to workstations based on either the load at the time nodes are allocated to some task, or based on an average load of our workstation cluster. The decisions related to load balance are made at compile time when resource requirements are estimated. This algorithm is the simplicity in terms of both implementation as well as overhead, since there is no need to constantly monitor the workstations for performance statistics. However, static algorithms only work well when there is not much variation in the load on the workstations.

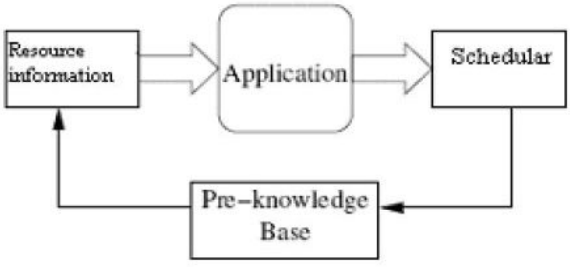


Figure 2 Static Load Balancing [17]

***A static load balancing techniques are:***

- Round robin algorithm - the tasks are passed to processes in a sequential order; when the last process has received a task the schedule continues with the first process (a new round)
- Randomized algorithm: the allocation of tasks to processes is random
- Simulated annealing or genetic algorithms: mixture allocation procedure including optimization techniques.

***Static load balancing still have some problems:***

- It is very difficult to estimate a-priori [in an accurate way] the execution time of
- Sometimes there are communication delays that vary in an uncontrollable way
- For some problems the number of steps to reach a solution is not known in advance

### *(b) Dynamic Load Balancing Algorithm*

Dynamic load balancing algorithms make changes to the distribution of work among workstations at run-time; they use current or recent load information when making distribution decisions. Multicomputers with dynamic load balancing allocate/reallocate resources at runtime based on no a priori task information, which may determine when and whose tasks can be migrated.

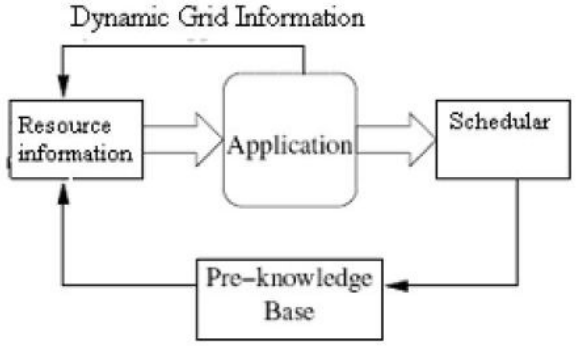


Figure 3 Dynamic Load Balancing [17]

Dynamic load balancing algorithms can provide a significant improvement in performance over static algorithms. However, this comes at the additional cost of collecting and maintaining load information, so it is important to keep these overheads within reasonable limits.

## V. LOAD BALANCING STRATEGIES

There are three major parameters which usually define the strategy a specific load balancing algorithm will employ [9]. These three parameters answer three important questions: Who makes the load balancing decision? What information is used to make the load balancing decision, and where the load balancing decision is made?

### *(a)Sender-Initiated vs. Receiver-Initiated Strategies*

The question of who makes the load balancing decision is answered based on whether a sender-initiated or receiver- initiated policy is employed [41]. In sender-initiated policies, congested nodes attempt to move work to lightly- loaded nodes. In receiver-initiated policies, lightly-loaded nodes look for heavily-loaded nodes from which work may be received.

### *(b) Global vs. Local Strategies*

Global or local policies answer the question of what information will be used to make a load balancing decision in global policies, the load balancer uses the performance profiles of all available workstations. In local policies workstations are partitioned into different groups. The benefit in a local scheme is that performance profile information is only exchanged within the group. The choice of a global or local policy depends on the behavior an application will exhibit. For global schemes, balanced load convergence is faster compared to a local scheme since all workstations are considered at the same time. However, this requires additional communication and synchronization between the various workstations; the local schemes minimize this extra overhead. But the reduced synchronization between workstations is also a downfall of the local schemes if the various groups exhibit major differences in performance. If one group has processors with poor performance (high load), and another group has very fast processors (little or no load), the latter will finish quite early while the former group is overloaded.

### *(c) Centralized vs. De-centralized Strategies*

A load balancer is categorized as either centralized or distributed, both of which define where load balancing decisions are made [17-19]. In a centralized scheme, the load balancer is located on one master workstation node and all decisions are made there. Basic features of centralized approach are: a master node holds the collection of tasks to be performed, tasks are sent to the execution node when a execution process completes one task, it requests another task from the master node.

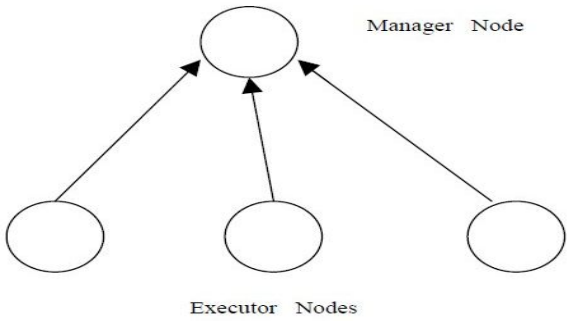


Figure 4 Centralized Strategies

## VI. CHALLENGES OF LOAD BALANCING

Challenges of Load Balancing[7] in Grid Computing There are many challenges related to load balancing in grid environment. Some are discussed further:

### A. Resource Heterogeneity

There are two types of resources in computational grid, first one are network resources and second one are computational resources in which heterogeneity exists. In networks resources it may be in terms of bandwidth and used network protocols. In case of computational resource there may be different hardware, architecture, no. of resources, physical memory size, CPU speed and so on. Heterogeneity results in differing capability of processing, which is the main cause for load balancing in heterogeneous environment.

### B. Site Autonomy

Grid with their distributed ownership and cross-domain organization gives a different type of problem in grid environment. As a direct result of distributed ownership, resource owners take management decision regarding local resources, often based on local information and local requirements. If the whole system would be shared there, then security of the system may leak. An application from the unauthorized user can't be run on any system. To efficiently manage the resources available in grid systems to meet the needs of an ever-changing and diverse user community, an automated, agile, and adaptive dynamic control system with grid-wide perspective is needed [3].

### C. Dynamic Behavior

In grid environment, there are more and more problems due to the heterogeneity of resources. Any time any resource may be available and can be unavailable due to machine failure or connection problems. This dynamic behavior always gives headache to the user of grid environment [3].

### D. Resource Non-Dedication

Due to non dedication of resource, resource may join the many grid systems simultaneously. A contention arises there when requests from many users come there. So due to resource non dedication resource usage contention is the major issue in grid environment.

### E. Application Diversity

In grid environment there is different types of users having different type of applications and has different requirements. For example some applications may have set of dependent jobs, other may have independent jobs, and on other side some application requires sequential execution. Keeping all aspects of jobs in mind, designing a general purpose load balancing system is extremely difficult [20]

## CONCLUSION

Grid Computing is definitely a promising tendency to solve high demanding applications and its related problems. Main objective of the grid environment is to achieve high performance. Dynamic nature and complexity of Grid make load balancing very complex and vulnerable to faults. To maintain entire load of nodes is very hard due to dynamic nature of resources in a Grid environment. There are a number of factors, which can affect the grid application performance like load balancing, heterogeneity of resources and resource sharing in the Grid environment. In future work on load balancing and presents factors due to which load balancing is initiated, compares existing load balancing algorithms and finally proposes an efficient load balancing algorithm for Grid environment. It also presents Job Migration technique for balancing load in Grid environment.

## RERERENCES

1. Foster, I., Kesselman, C. and Tuecke, S. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" International Journal of High Performance Computing applications, Vol. 15, No. 3, pp. 200-222,2001
2. Klein rock, L. "MEMO on Grid Computing", University of California,Los Angeles, 1969
3. Book - "Introduction to Grid Computing (Hardcover - 2009)" (url - http://www.flipkart.com/introduction-grid-computing- magoules-frederic/1420074067-eox3f9kfvb)
4. Jim Waldo, Geoff Wyant, Ann Wollrath, and Sam Kendall, "A Note on Distributed Computing", Sun Microsystems Laboratories 2550 Garcia Avenue Mountain View, CA 94043.
5. Lord, P.Alper, P.Wroe, C. and Goble, C., "Feta: A lightweight
6. Architecture for User Oriented Semantic Service Discovery", In Proceedings of The Semantic Web: Research and Applications: Second European Semantic Web Conference (ESWC 2005), Heraklion, Crete.
7. Sanjay P. Ahuja, Jack R. Myers Department of Computer and Information Sciences, University of North Florida, Jacksonville, "A Survey on Wireless Grid Computing", The Journal of Supercomputing, 37, 3-21, 2006.
8. Daniel Adler "Grid Computing in Science" Georg-August Universidad Gottingen Institute for Informatics Software Engineering for Distributed Systems Group 08. November 2007.
9. Manish Parashar, A.Lee, "Grid Computing: Introduction and Overview".
10. Ratnesh Kumar Nath, "Efficient Load Balancing Algorithm in Grid Environment", Thapar University, Patiala, May 2007.
11. Belabbas Yagoubi and Yahya Slimani, "Dynamic Load Balancing Strategy for Grid Computing",

World Academy of Science, Engineering and Technology 19, 2006.
12. Javier Bustos Jimenez, "Robin Hood: An Active Objects Load Balancing Mechanism for Intranet", Departamento de Ciencias de la Computacion, jbustos@dcc.uchile.cl, Universidad de Chile.
13. S. Iqbal, Load balancing strategies for parallel architectures, Ph.D. Thesis, Univeristy of Texas at Austin, May 2003
14. R. Buyya and J. Giddy and H. Stockinger, Economic Models for Resource Management and Scheduling in Grid Computing, in J. of Concurrency and Computation: Practice and Experience, Volume 14, Issue (13-15), Pages (1507-1542), Wiley Press, Dec. 2002.
15. Parag Kulkarni & Indranil Sengupta Department of Computer Science & Engg. Indian Institute of Technology, Kharagpur, "Load Token Policy" ICTA'07, April 12-14,
16. Shahzad Malik, "Dynamic Load Balancing in a Network of Workstations", 95.515F Research Report, November 29, 2000.
17. Miguel L. Bote-Lorenzo, Yannis A. Dimitriadis and Eduardo Gómez-Sánchez, "Grid Characteristics and Uses: a Grid Definition"
18. Kai Lu, Riky Subrata and Albert Y. Zomaya, Networks & Systems Lab, School of Information Technologies, University of Sydney "An Efficient Load Balancing Algorithm for Heterogeneous Grid Systems Considering Desirability of Grid Sites".
19. B. Yagoubi , Department of Computer Science, Faculty of Sciences, University of Oran and Y. Slimani , Department of Computer Science, Faculty of Sciences of Tunis, "Task Load Balancing Strategy for Grid Computing".
20. http://images.google.co.in/images.
21. P.Senthil Kumar, S.Renuka Devi HOD Of MCA Department Cherraan's Arts Science College, Kangayam. Research Scholar, Cherraan's Arts Science College, Kangayam " National Seminar on Bio Computing 2009".
22. Frederic Magoules, Jie pan, Kiat-An Tan, Abhinit Kumar, "Introduction to Grid Computing", CRC Press, A chapman and hall Book.