

REVIEW ARTICLE

Available Online at www.jgrcs.info

A REVIEW ON DIFFERENT APPROACHES FOR LOAD BALANCING IN COMPUTATIONAL GRID

Mr.Gaurav Sharma^{*1}, Jagjit Kaur Bhatia²

^{*1} Assistant Professor, Department Of CSE, JMIT, Radaur, Yamunanagar, India
gaurav13@gmail.com, 8295411954¹

² Jagjit Kaur Bhatia, Department OF CSE (M-tech), JMIT, Radaur, Yamunanagar, India
jiitjagjit@gmail.com,9729330287²

Abstract: In heterogeneous environment like Grid, which allow the use for geographically widely distributed and multi-owner resources to solve large-level application. So for keep maintaining the balance of workload between emerged infrastructures like grid, use of load balancing algorithms was important. Jobs are needed to be equally balanced between various computing nodes. To fully utilized the resource in heterogeneous network and increasing computation and overall improve the throughput of system is aim of load balancing algorithm. The purpose of this paper is to review various different load balancing algorithms for the heterogeneous network like grid and to identify various metric and identify gaps between them. Many load balancing algorithms are already implemented which works against various issues like heterogeneity, scalability etc. Different load balancing algorithms for the grid environment woks on various metrics such as make span, time, average resource utilization rate, communication overhead, reliability, stability, fault tolerance.

Key words: Grid Computing, Load Balancing, Distributed Computing, Resource Management.

INTRODUCTION

Grid computing is emerging as a wide scale distributed infrastructure which allows large scale resource sharing and synchronized problem solving in dynamic a heterogeneous network. Numbers of resources are interconnected and work independently by cooperating with each other. Workload represents the amount of work to be performed where all resources have different processing speed. A grid environment can offer a resource balancing results by scheduling grid jobs properly [1]. An improved scheduling and efficient load balancing algorithm across the grid may lead to improve the overall system performance with less execution time. Load balancing is required to fairly distribute the tasks across various resources so as to increase the computation and minimum task execution time. In a grid some nodes may be heavily loaded while some may be idle or say under loaded. So a better load balancing algorithm is about to prevent from the condition where some resources are over burdened with work and some are not fully utilized or say free. Grid environment involves various issues [2] as follows.

Heterogeneity:

Heterogeneity refers to the use of different technologies and management policies that exists in both of computational and network resources.

Autonomy:

It refers to autonomous because the multiple owned organizations that share Grid resources, a site are viewed as an autonomous computational entity.

Scalability:

Problems involved when an grid grow from few of resources to millions. Better fault tolerant service and quality capability required.

Dynamicity:

Resource failure is possible it can be due to some hardware of software problem or connection disturbance. So to adopt a dynamic behavior to deal with such circumstances is important.

Resource balancing:

Balance the workload on millions of resources itself a challenge. Fair distribution and proper migration policies needs to be implemented.

Reliability and Management:

To keep the data in reliable form there are other issues involved that need to be handled.

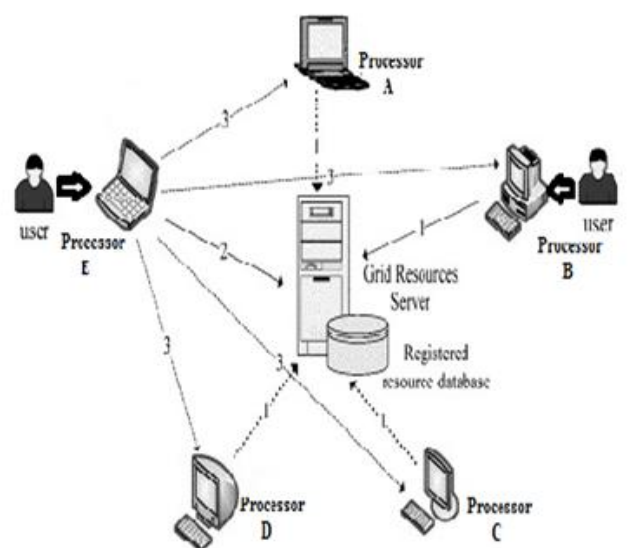


Figure 1.1 [Computational Grid Environment]

A load balancing algorithm is used to fully exploiting the unused resources, and has the possibility substantially increasing the Efficiency of resources usage, to enhance the

performance and speed of system with no wastage of time [3]. They are also important for the purpose of sharing of computational results, fulfilling the periodic computational needs and overall to meet the goal of balancing the workload among resources.

RELATED WORK

Many load balancing algorithms has been proposed in this field. It is difficult to achieve load balancing in grid systems than in traditional distributed computing environment because of various issues and its dynamic nature. Many schemes presented are based on centralized structure. All of them suffered from significant deficiencies, such as scalability issues. Static and dynamic load balancing techniques are the mostly used techniques for task allocation in grid environment [6]. Several works have been done on dynamic load balancing approach A load balancing model based on tree representation of a grid is proposed. In [5] Yagoubi et al presented a hierarchical load balancing strategy which uses a task-level load balancing. In [6], Menno et al evaluated the effectiveness of dynamic load balancing and job replication by means of trace-driven simulations.

They proposed a solution to users of parallel application and distributed environment that weather to use DLB or JR. Agent-based approaches have tried to provide load balancing in cluster of machines [7]. Junwei Cao, et al concerned on load balancing while developing parallel and distributed computing applications. & when the issues of cross-domain and large-scale scheduling comes then emergence in computational grid extends the problem. So In this author proposed work with an agent-based grid management infrastructure which is coupled with a performance-driven task scheduler that has been developed for local grid load balancing. In [8], Shah et al represented two job migration algorithms, which are MELISA (Modified ELISA) and LBA (Load Balancing on Arrival).

These were differ in the way load balancing is carried out and has proved its efficiency in minimizing the response time on large and small-scale heterogeneous Grid environments authors proposed a decentralized grid model as a collection of clusters and then introduced a dynamic load balancing algorithm (DLBA) which performs intra-cluster and inter-cluster (grid) load balancing. Yajun Li et al in presented an hybrid strategy [9] for load balancing in grid environment which takes the advantage of two approaches average based, instantaneous approach by merging them. A new decentralized algorithm [10] proposed algorithm at Meta scheduler and cluster or resource level. Jasma et al proposed a fault optimal load balancing algorithm [11] by recognizing the available challenges in grid environment. To ensure the reliability in distributed grid environment, fault tolerance should be high.. In a grid like environment where thousands of computing nodes connected to each other, reliability of each and every resource cannot be guaranteed. Hence, it's necessary in order to eliminate the probability of failure is in grid computing. Main goal is to prevent, from condition where some processors are overloaded with a set of tasks while others are less loaded or free.

LOAD BALANCING APPROACHES

- a. **Static load balancing**-In static the number of processors is fixed, it is assumed that some priori information exist, but if any change occur in problem size, the fixed number of processors may not be sufficient and in some environments all processors cannot be used all the time. So it required some strategy which deal with such circumstances and overcome this problem. Round-robin, simulated annealing, randomized are some of techniques for static load balancing. It leads to use of dynamic load balancing [1].
- b. **Dynamic load balancing**- algorithms make changes to the distribution of work among computing nodes at run-time. They make use of current and recent load information when making distribution decisions and they continually monitor the load on all the processors and when the load imbalance reaches some predefined level, the redistribution of work is done [2].

The Static approach is attractive because of its simplicity and minimized runtime overhead. However it has disadvantage which assumes that the characteristics of the computing resources and communication network are all known in advance and will be constant. Such assumptions cannot be applied to grid environment.

- c. **Parameters**-There are basically three important parameters which determine that which load- balancing strategy will be employed
 - a) who takes the decision for load balancing?
 - b) What type of information is required for making the load balancing decision?
 - c) Where the decision about load balancing is made?

STRATEGIES FOR LOAD BALANCING

- a) **Centralized & Decentralized Strategy:**
 - a. **Centralized Strategy**-In centralized approach [6], only one node in the distributed system acts as the main or central controller. This main node has global view on the load information of all nodes connected to it, and decides how to allocate jobs to each of the nodes. And rest of the nodes act as slaves.
 - b. **Decentralized**- all nodes in the distributed system are involved in making the load balancing decision. It is commonly agreed that distributed algorithms are more scalable and have better fault tolerance
- b) **Sender-Initiated & Receiver-Initiated Strategies:-**
 - a. **Sender-initiated**: In sender initiated strategy, Congested nodes attempt to move work towards under-loaded nodes. Sender-initiated policy works well than the receiver-initiated strategy at low to moderate system loads. The reasons behind is that, the probability of finding a lightly-loaded node is higher than that of finding a heavily-loaded node.
 - b. **Receiver-initiated**- In this type, Less-loaded nodes look for heavily-loaded nodes from which work may be received similarly, at high system loads; the receiver initiated policy works better since it is much easier to find a heavily-loaded node.

c) **Global & Local Strategies:-**

- a. **Global Strategy:-** The load balancer uses the performance profiles of all available nodes. Global or local policies both replies the question for what information will be used to make a load balancing decision in global policies. For global schemes, balance load speed is faster compared to a local scheme since all workstations are considered at the same time.
- b. **Local Strategy-**In local scheme workstations are partitioned into different groups. The benefit in a local scheme is that performance profile information is only exchanged within the group..

d) **Co-operative & Non-co-operative:-**

- a. **Co-operative strategy-** is one in which load is shared by other node, In other words nodes co-operate with each other. & on the other side, if they don't reflects **non co-operative** strategy behavior. It takes their decision own to balance load. These are the main strategies used in load balancing mechanism.

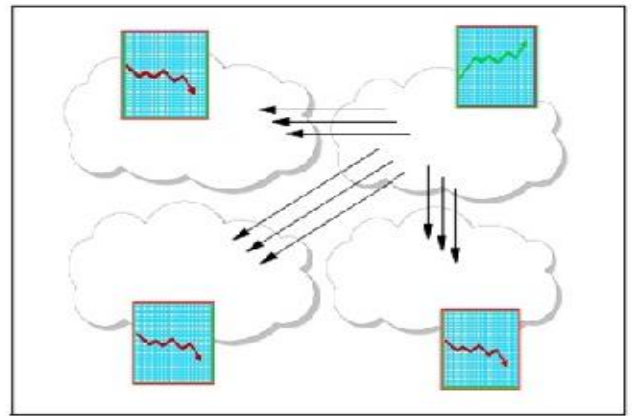


Figure: 2 [Job migration Process]

POLICIES OF LOAD BALANCING

A good load balancing algorithm is defined by some basic policies which are Transfer policy, Selection policy, Location policy and Information policy, resources type policy, triggering policy [1].

- a. **Information policy:** specifies what workload information to be collected, updated, When it is to be collected and from where.
- b. **Triggering policy:** determines the appropriate period to start a load balancing operation.
- c. **Resource type policy:** classifies a resource as server or receiver of tasks according to its availability status. Whether resource is available to retrieve the task from most overloaded resource.
- d. **Location policy:** uses the results of the resource type policy to find a suitable partner for a server or receiver
- e. **Selection policy:** defines the tasks which should be migrated from busiest resources (source) to most idle resources (receiver). A selection policy considers several factors in selecting a task, for e.g. transfer of small task will take less overhead.

On the basis of Location policy, the dynamic load balancing algorithms can be further classified into sender initiated algorithms, receiver initiated algorithms and symmetrically-initiated algorithms

LOAD BALANCING STEPS

There are some main steps that almost all load balancing algorithms have in common [4].

- a. **Monitoring:** - Monitoring resource load state.
- b. **Synchronization-** Exchanging load and state information between resources.
- c. **Rebalancing Criteria-** Calculating the new work distribution and making work moment decision.
- d. **Job Migration-** Its actual movement of data.

It provides when a system decides to export a process. It decides whether to create it on local site or create it on a remote processing site.

VARIOUS ISSUES

Dynamic load balancing may consider following issues, however it need to collect and maintain information about available nodes [1].

- a. **Process transfer issue:** It takes care to determining whether to execute a process locally or remotely.
- b. **State information exchange issue:** It determines how to get exchange the collected load information among various nodes.
- c. **Load estimation issue:** This policy specify the issue regarding to estimate the workload of a particular node of the system.
- d. **Migration issue:** Main job of this policy to migrate the load from one state to another. It determines total number of times a process is migrating.

COMPARISON OF DIFFEENT LOAD BALANCING ALGORITHM IN GRID BASED ON VARIOUS METRIC/ISSUES

A comparison has been shown in the following table for different load balancing algorithms based on various metric such as communication overhead means to message traffic while communicating , load balancing time, scalability, heterogeneity etc;

Algorithm/ Metric	Agent based approaches for load balancing	Fault-tolerant optimal neighbour load-balancing	Dynamic load balancing algorithm in grids	Decentralized load balancing algorithm in grid
Communication Overhead	More	More	Less	More
Makespan	Less	Average	More	Less
Load balancing time	Less	Less	More	Less
Scalability	Scalable	Scalable	Scalable	Scalable
Avg. Resource Utilization rate	Average	Improved	More	More
Fault Tolerance	Integrated	Very high	Integrated	Integrated
Reliability	Integrated	High	Integrated	Integrated

Figurw 1.2- (Comparison of load balancing algorithm)

CONCLUSION

This paper presents a comparative survey of load balancing algorithms in grid environment. The accepted techniques of load balancing in grid environment with their importance, combinations and variations have been discussed. Grid application performance remains a challenge in dynamic grid environment. Resources submitted to Grid and can be withdrawn from Grid at any moment. Main objective of load balancing algorithm is to achieve high performance in grid environment by optimal usage of geographically distributed and heterogeneous resources. So such an algorithm which efficiently manage and balance the workload also according to working capacity of processor and minimized the execution time and increase the global throughput of system, is required in such an unpredictable environment of grid. However, accepting the importance of all the aforesaid areas, to put forward a future direction of work, this research would next focus on finding optimal approach for better performance of applications running in grid.

REFERENCES

- [1]. Lalitha Hima, Venkatesan, Karunya university, coimbatore, India "Perspective Study on Resource level Load balancing in Grid Computing Environments" 2011 IEEE.
- [2]. Miguel L. Bote-Lorenzo, Yannis A. Dimitriadis, "Grid Characteristics and its usage Springer-Verlag LNCS 2970, Spain, 2009.
- [3]. M. Baker, R. Buyya, and D. Laforenza, "Grids and Grid Technologies for Wide Distributed Computing", International Journal of Software: Practice and Experience (SPE), Vol. 32, no. 2010.
- [4]. Neeraj Rathore, Chana "A Cognitive Analysis of Load Balancing and Job Migration Techniques" World Congress on Information and Communication Technologies, January IEEE 2011.
- [5]. B. Yagoubi, and M. Meddeber, "A load balancing model for grid environment", Proceeding of 22nd International Symposium on Computer and Information Sciences (ISCISC 2007), pp. 1-7, 7 November 2007.
- [6]. M. Dobber, R. Mei, and G. Koole, "Dynamic Load Balancing and Job Replication in a Global-Scale Grid Environment: A Comparison", IEEE Transaction on Parallel and Distributed Systems, Vol. 20, no. 2, pp. 207-218, February 2009.
- [7]. Junwei Cao "agent based using performance driven task scheduling" C&C Research Laboratories, NEC Europe Ltd., Sankt Augustine, Germany IEEE.
- [8]. Shah, Veeravalli, "On the Design of Adaptive and Decentralized Load-Balancing Algorithms with Load Estimation for Computational Grid Environments" IEEE Transactions On Parallel And Distributed Systems, 2007.
- [9]. "A Hybrid Load balancing Strategy of Sequential Tasks for Computational Grids" Yajun Li, Yuhang Yang Rongbo Zhu Electronic Engineering Department Shanghai Jiaotong University, China 2009.
- [10]. Rajkumar Rajavel, Kongu Engineering College, Perundurai, Erode, India. December, 2010 "Decentralized Load Balancing for the Computational Grid Environment".
- [11]. Jasma, Nedunchezian Raju Department of Computer Science and Engineering, Atria Institute of Technology, "A Fault Tolerance Optimal Neighbour Load Balancing Algorithm for Grid Environment", 2010.