International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
National Conference on Advances in Engineering and Technology
(AET- 29th March 2014)

RESEARCH ARTICLE OPEN ACCESS

Dynamic Load Balancing Approach For Grid Environment

Suchitra*, Sandip Kumar Goyal**
*(Department of Computer Science & Engineering, M. M. University, Mullana
Email: er.suchitra27june@gmail.com)
** (Department of Computer Science & Engineering, M. M. University, Mullana
Email: skgmmec@gmail.com)

ABSTRACT
Next generation computing involves the sharing and aggregation of geographically distributed resources which results in large-scale load balancing problems. In a grid system, Load Balancing has been an increasingly important issue for handling computational intensive jobs. By developing methodologies that can allocate such jobs to resources in a way that balance out the load, the total processing time will be reduced. In this paper, an algorithm based on the queue length of each resource and waiting time of each task, is proposed for achieving job allocation with load balancing. The simulation results show that proposed algorithm yields better performance when compared with other traditional approach.
***Keywords* -** Grid System, Job allocation, Load Balancing.

I. INTRODUCTION

The availability of low cost powerful computers coupled with the popularity of the Internet and high-speed networks have led the computing environment to be mapped from classical distributed to grid environments. To improve the global throughput of these environments, effective and efficient load balancing algorithms are fundamentally important. Emerging as new distributed computing environments, computational grids [1] provide an opportunity to share a large number of resources among different organizations. Grid computing involves coupled and coordinated use of geographically distributed resources for purposes such as large-scale computation and distributed data analysis [2] [3]. Performance enhancement is one of the most important issues in such grid systems. However, in many situations, poor performance is due to uneven load distribution among the nodes in the system. Therefore, to fully exploit the computing power of such grid systems, it is crucial to employ a judicious load balancing strategy for proper allocation and sequencing of tasks on the computing nodes. A lot of research has already been done in the field of load balancing.

II. RELATED WORK

In [4], authors proposed a dynamic load balancing algorithm which considers CPU length, CPU and memory utilization, network traffic as load metric. In [5], authors presented a load metric "Tendensive Weight" and this metric consider both CPU and Input-Output utilization including memory access rate. An algorithm is developed for calculation of the Tendensive Weight of each task to be distributed to the node. The workload estimation of each device for LB using fuzzy system is implemented in [6]. It used run-queue length and CPU utilization as the input variables for fuzzy sets and a set of membership function is defined. This scheme focuses only on run queue length and CPU utilization factors, other factors such as cost of migration, reliability etc. are not considered. In [7], authors proposed a decentralized load-balancing algorithm for a Grid environment. The presented method does not consider the actual cost for a job transfer. In [8], authors addresses several issues that are imperative to Grid environments such as handling resource heterogeneity and sharing, communication latency, job migration from one site to other, and load balancing. Two job migration algorithms, which are MELISA (Modified ELISA) and LBA (Load Balancing on Arrival) are proposed. The algorithms differ in the way load balancing is carried out and is shown to be efficient in minimizing the response time on large and small-scale heterogeneous Grid environments, however, no provision is provided for fault tolerance. In [9], authors presented an algorithm for dynamic load balancing in distributed systems with multiple supporting nodes by exploiting the interrupt service.

III. SYSTEM MODEL

The main objective is to propose a dynamic load balancing algorithm that can handle load in efficient way.

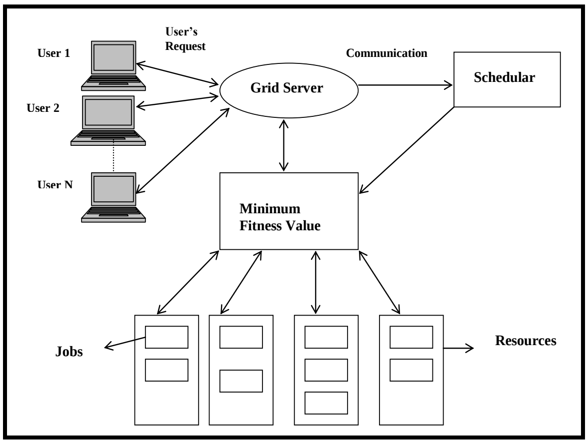

Figure 1: Computational Grid Model

Presented model make use of a centralized server called Grid server, in order to dynamically allocate and balance the load to different servers available or registered in the Grid environment as shown in Fig. 1. In this computational Grid model, the main role factors and performers are Grid Server, Users, Resources and Scheduler. Each resource is a computational unit with different processing power. All the resources and users register their information and update their status time to time to the Grid Server. Scheduler schedules the tasks to the resources according to the load divided by the grid server.

IV. PROPOSED METHODOLOGY

The proposed algorithm will balance the load based on the queue length of each resource and waiting time of each task. It transfers the job to the resource having minimum computation value of queue length and waiting time. The proposed algorithm for dynamic load balancing in grid environment is as follows:

i) Input the value of number of Users (NU) and Resources (NR).
ii) Create the Expected Time For Computation [ETC] Matrix for each Resource.
iii) Get the availability of all registered Resources.
iv) Request for the queue length and waiting time of each resource.
v) Initialize the NextUser = 0 and QueueLength of each resource= 0.
vi) Find the resource ' R ' with minimum computed Queue Length (ql) and Waiting time(wt).
vii) Allocate the job of NextUser to R which is having minimum fitness value (F) given by :
F=0.5 (ql) + 0.5 (wt) .
viii) NextUser = NextUser + 1.
ix) QueueLength_R = QueueLength_R + 1.
x) Check for the arrival of any job ' J ' from Resource R' after completing the execution.
xi) If no resource arrival exist then goto step xiv).
xii) QueueLength_R' = QueueLength_R' – 1.
xiii) Make Span= max time computed from the available resources.
xiv) If NextUser <= NU then goto step vi).
xv) Print ExecutionTime_J of all jobs.
xvi) Print which resource has been allocated to which User.

V. SIMULATION RESULTS

Table 1 show the parameters used during simulation of Proposed Methodology (PM).

Table 1: Parameters Used

No. Resources	3 – 40
No. of Users	10 – 50
Type of Resources	Heterogeneous

The execution time of jobs corresponding to different users using PM and queue based algorithm is shown in Table 2, Table 3, Table 4 and Fig. 2, Fig. 3, Fig. 4 respectively. The graph shows that the execution time of jobs under random queue length based algorithm is more than that of execution time of jobs proposed.

Table 2: Make span value of various users when number of resources = 8

Make Span Value		
Users	Proposed Method (PM)	Queue Based Method (QBM)
15	132.6	136.3
25	252.3	257.3
35	321	327.3
40	330	335
50	446.6	456.6

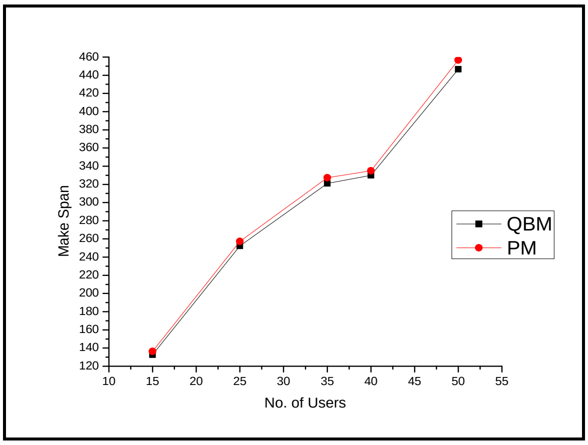

Figure 2: Make span value of various users when number of resources = 8

Table 3: Make span value of various resources when number of users =100

Make Span Value		
Resources	Proposed Method (PM)	Queue Based Method (QBM)
10	657	662.3
15	455.3	465.3
20	332	336.3
30	259.3	268
40	196.6	204

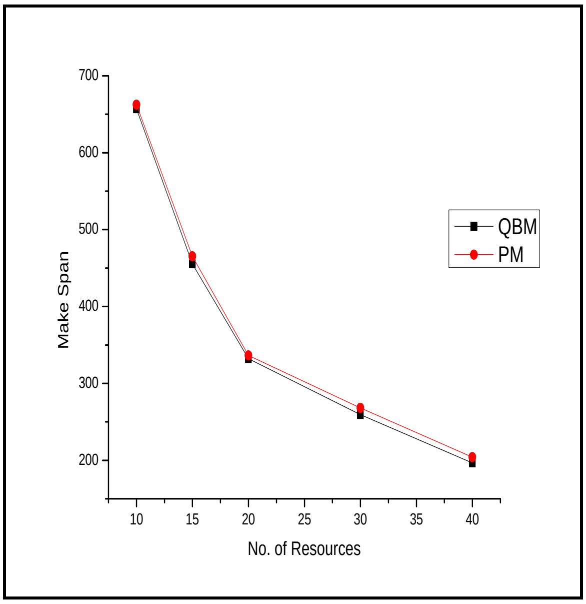

Figure 3: Make span value of various resources when number of users=100

Table 4: Make span value at various stages when sers and resources both vary.

Make Span Value		
Resources, Users	Proposed Method (PM)	Queue Based Method (QBM)
3,10	253	261
10,35	260	267
15,45	196	203
18,50	197	201

The results show that our Proposed Method (PM) is better than the Queue Based Method (QBM) algorithm in all scenarios.

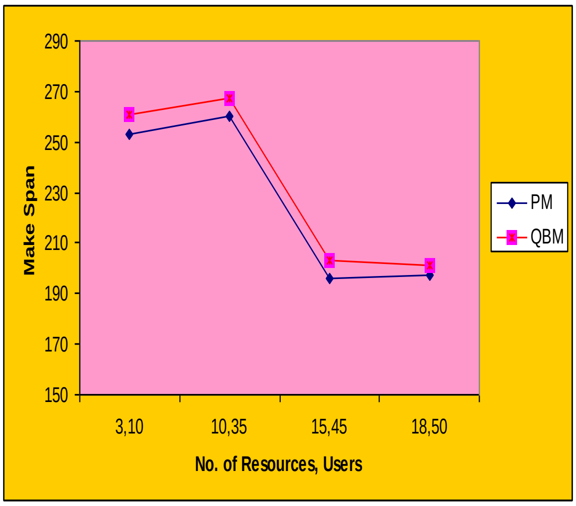

Figure 4: Make span value of various resources and users when both vary.

VI. CONCLUSION

Main objective of the grid environment is to achieve high performance computing by optimal usage of geographically distributed and heterogeneous resources. Grid application performance remains a challenge in dynamic grid environment. Resources can be submitted and can be withdrawn from at any moment. This characteristic makes Load Balancing one of the critical features.

There are a number of factors, which can affect the grid performance like load balancing, heterogeneity of resources and resource sharing in the Grid environment. In this paper author focused on Load Balancing and proposed an efficient load balancing approach for grid environment and also analyzed existing (QBM) method as well as proposed a new method (DLBA) which more efficiently balances load as illustrated by simulation results.

REFERENCES

[1] Yajun Li, Yuhang Yang, and Rongbo Zhu," A Hybrid Load balancing Strategy of Sequential Tasks for Computational Grids", *International Conference on Networking and Digital Society (ICNDS), 9(1), 2009,112-117.*

[2] N. Yamamoto, O. Tatebe, and S. Sekiguche, "Parallel and Distributed Astronomical Analysis on Grid Datafarm", *Proceeding of 5th IEEE/ACM International Workshop on Grid Computing*, 2004, 461-466.

[3] A. Touzene, S. Al-Yahai, H. AlMuqbali, A. Bouabdallah, and Y. Challal,"Performance Evaluation of Load Balancing in Hierarchical Architecture for Grid Computing Service Middleware", *International Journal of Computer Science Issues, 8(2), 2011, 213-223.*

[4] P. Werstein, H. Situ, and Z. Huang, "Load Balancing in a Cluster Computer", *Proceeding of 7th International Conference on Parallel and Distributed Computing Applications and Technologies, 2006, 569-577.*

[5] K. Shahu, K. Chand, Y. Kumar , and A. Babu, "A Novel Approach for Effective Dynamic Load Balancing by using Tendensive Weight", *International Journal of Computer Science and Network Security*, *8(6)*, 2008, 42-48.

[6] Yu-kwong, and Lap-sun," A new fuzzy-decision based load balancing system for distributed object computing"**,** *Journal of Parallel and Distributed Computing*, *64(2)*, 2004, 238-253.

[7] S. Das, M. Arora, and R. Biswas, "A De-Centralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid Environments", *Proceeding of International Conference on Parallel Processing Workshops*,3(4), 2002, 499-505.

[8] R. Shah, V. Bhardwaj, and M. Misra, "On the Design of Adaptive and Decentralized Load-Balancing Algorithms with Load Estimation for Computational Grid Environments", *IEEE Transaction on Parallel and Distributed System*, *18(12)*, 2007, 1675-1686.

[9] P. Jain, and D. Gupta, "An Algorithm for Dynamic Load Balancing in Distributed Systems with Multiple Supporting Nodes by Exploiting the Interrupt Service", *Proceeding of International Journal of Recent Trends in Engineering*, *1(1)*, 2009, 232-236.