e-ISSN: 2349-9745
p-ISSN: 2393-8161
Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Optimal Pricing and Load Balancing Approach for Computational Grid

Dr. P. Suresh[1],Dr. P. Keerthika[2], S. Nandhini[3]
[1]*Information Technology, Kongu Engineering College*
[2]*Computer Science and Engineering, Kongu Engineering College*
[3] *Information Technology, Kongu Engineering College*

Abstract— Grid computing is the federation of computer resources from multiple administrative domains to reach a common goal. The main challenging task of grid is managing resources and cleverly pricing them on computing systems. Resource sharing insists careful load balancing for an efficient and effective deployment of resources in the system when it is needed. The main goal of this paper is to integrate load balancing and optimal pricing for improving the resource utilization rate and user satisfaction. The system assumes heterogeneous resources with dynamically varying prices to determine load balancing and the load is balanced among the resources with their minimal price. It provides optimum solution so that it reduces the response time and overall cost charged for job execution. Through simulations the performance of the proposed system is evaluated.

Keywords- load balancing; pricing; resources; Grid Computing; scheduling; computational grid.

I. INTRODUCTION

1.1.Grid Computing

Grid computing enables access to computing resources distributed in different locations for users. Such an environment has distinctive characteristics which distinguish them from other distributed systems and also conventional parallel processing systems [1] Grid computing targets at gathering resources and combines the power of both parallel computing and distributed computing. Resource sharing is supported by distributed computing and computing power is supported by parallel computing. Grid can be categorized as computational grid and data grid [2]. A computational grid is a federation of heterogeneous resources in a network for applying to the single problem at the same time. The data grid enables efficient storage and distributions of data. It is usually to solve a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. The main emphasis on grid is resource management and job scheduling. Resource management is an efficient and effective deployment of resources in the system when it is needed. Job scheduling is a decision process by which application components are assigned to available resources to optimize various performance metrics. The need for scheduling arises from the requirement for most modern systems to perform multitasking i.e., executing more than one process at a time and load balancing i.e., distribution of workloads among the processors [3] [8].

1.2. Grid Computing Issues and Challenges

The resources available in the grid environment are basically owned and managed by multiple organizations [8]. Various factors like resources sharing and load balancing are considered for the efficient operation of computational grid. The primary challenges that should be taken into account for grid system are follows:

- Administration & Security
- Resource Management
- Information services
- Data management

1.3. Load Balancing

Load balancing distributes workloads across multiple computing resources [3]. It aims to improve resource use, reduce response time, maximize throughput and avoid overload. The load-balancing feature must always be integrated into any system in order to avoid processing delays and over-commitment of resources. Load balancing applications can be built in connection with schedulers and resource managers. The workload can be pushed outbound to the resources based on the availability state of the resources. This level of load balancing involves job partitioning, resource identifications and job queuing. Load balancing algorithms can be broadly classified into two categories: static and dynamic [8]. In static load balancing algorithm, distribute the work among processors prior to the execution of the algorithm, where estimation of resource requirements occurs. In dynamic load balancing algorithm, distribute the work among processors during the execution of the algorithm.

II. RELATED WORK

The purpose of load balancing is to improve the performance of the system by allocating the load properly on the resources. Usage of load balancing in various processes is explained in the following works. Intelligent Colonies of ants discuss about the meta-heuristic technique that is implemented for grid load balancing. An intelligent ant has been used to improve the performance of the system, and a new concept called Ant level load balancing is implemented. When the system is overloaded, a new ant is created [7]. So it allocates more memory based on the decision making algorithms and the efficiency of the system decreases due to the creation of many new ants.

A genetic based load balancing algorithm [1] for heterogeneous distributed systems in which half of the processors have double the speed of the others. In this method Least Expected Response Time for generic jobs Maximum Wait (LERT-MW) for dedicated jobs is used, it broadly classify the jobs into two categories and allocate the first set of jobs to speedy processors and the other set can be allocated to any resource i.e. generic. When the number of tasks increased, the efficiency of the system decreased.

Hybrid Particle Swarm Optimization (HPSO) method is used to solve the Task Assignment Problem (TAP) which is a non-deterministic polynomial time-hard problem [13]. It has been developed to dynamically schedule heterogeneous tasks on to heterogeneous processors in a distributed environment. The nature of the tasks is independent and non-pre-emptive. The HPSO yields a better result than the normal Particle Swarm Optimization (PSO) when applied to the task assignment problem and it also compared with Genetic Algorithm (GA).

Hybrid load balancing technique dealt with combination of both the static and dynamic load balancing for addressing the problem of resource allocation. The metric of update interval for reducing the delay and deadlock is used [6]. It reduces the waiting time of the jobs and providing them with priority, leading to the reduction of execution time. On the other hand consideration of processing elements capacity and site efficiency has to be done for more efficiency.

Load Balanced Min-Min (LBMM) algorithm [5] is used to reduce the makespan and increases the resource utilization. It execute Min-Min algorithm and the tasks are rescheduled to use the unutilized resources effectively. The system reduces the makespan and consumes more time to balance the load and results in high cost.

Capacity based load balancing proposed load balancing policy of two-level [11] for the multi-cluster grid system where computational resources are dispersed in different administrative domains or clusters which are located in different local area networks. Minimization of overall response time and maximization of system utilization and throughput are achieved through the consideration of the processing element's capacity and hence achieve an appropriate load balance. But the improvement ratio is decreased when the work load increases.

The proposed work is different from these existing works. The objective of the proposed system is to minimize time and minimize cost. The system prefers the resources with short response time without considering their prices, which results in high cost and the user is not satisfied with the system [12].

The proposed load balancing technique provides solutions to it. Along with these, the existing work have some problems like decreasing the improvement ratio and efficiency of the system, more overhead time and high cost which are also considered in the proposed system.

III. PROBLEM FORMULATION

Grid resources are dynamic in nature and applying the concept of load balancing is a most arduous task. So an efficient load balancing is required. Budget is a major constraint [10] because the user may have a maximum total cost to pay for the resources. If the whole cost is higher than the budget, user might not use the resources. These problems can be solved by considering the following parameters,

- Resources utilization
- Makespan
- Cost

Let N number of tasks has to be scheduled with the M number of resources in the grid system. Consider set of tasks and set of resources.

$$\Pi = \{T_1, T_2, \ldots, T_N\}$$

$$\Gamma = \{R_1, R_2, \ldots, R_M\}$$

Then scheduling is defined as mapping of tasks to resources.

i.e., $\Pi \longrightarrow \Gamma$

IV. PROPOSED SYSTEM

The main objective of the proposed system is to improve the resource utilization and to implement the optimal pricing method where the resources are efficiently utilized by applying the strategy load balancing and the pricing method is implemented based on the user's expected cost. So, the system integrates the load balancing and pricing method to allocate jobs to the resources with minimum cost and minimum time.

In the proposed system, the grid resources are heterogeneous and their prices vary dynamically [4]. The user submits the jobs to grid scheduler. The submitted jobs are scheduled based on the user's expected cost, number of jobs at particular time and resources capability. In the system, load balancing is to distribute the workload among available resources and all the resources involve in the grid utilized equally as much as possible. A load is the number of jobs in the waiting queue and can be low, moderate and heavy according to their work.

4.1. Grid Environment Creation

The primary components of a grid system are Grid User, Grid Resources, Resource Broker and Grid Information Service (GIS). Initially, the grid user interacts with the Resource Broker and sends their task to computation. So, create number of gridlets i.e., jobs submits by the user. The gridletscontains all the information about jobs like length of the job, file size and output size. Then resource broker discovers the resources for scheduling strategies and task processing. So, create the number of heterogeneous resources with dynamically varying price, availability of the processor, processor speed and memory of the resources and register those resources to GIS, where GIS worked as an agent and collects all the relevant information such as resource availability, node capacity and provide it to the resource broker to make the scheduling decision.

4.2. Load Balancing Strategy

Load balancing strategy is implemented while the jobs are scheduled. The strategy is done by considering the current load of a resource [10]. A load is the number of jobs in the waiting queue. Current load at each resource is calculated and resources are classified based on the load, whether it is under loaded, over loaded and normally loaded. Then the resources are selected in the under loaded

list and the jobs are allocated to a resource that satisfies user's cost.

The current load of the resource is calculated by all the length of the jobs submitted to the particular resource with their Million Instructions Per Second (MIPS) rating and Availability Time (AT) of the resource. The formula is illustrated in the Equation (1)

$$load_i = \frac{\sum_{j=1}^{n} length}{MIPS_i \times AT_i} \quad (1)$$

Where, i-Resource

j-job

n-number of jobs that are allocated to the resource i

The average of the load is calculated to know whether the load is low, moderate or heavy by using the Equation (2).

$$Avg.load = \frac{\sum_{i=1}^{n} load}{n} \quad (2)$$

Expected Execution Time (EET) is calculated to know how long the job will take to complete. It is calculated based on length of the jobs and MIPS rating of the resource using Equation (3).

$$EET(i, j) = \frac{length_j}{MIPS_i} \quad (3)$$

Expected Cost (EC) is calculated to take scheduling decisions because the decision is taken based on the cost, jobs at particular time and load of the resources. EC is calculated in the pricing method.

4.3. Pricing Method

Pricing method is implemented based on the user's expected cost [9]. The Expected Cost for execution of the job at each resource is calculated. Then Choose a resource that satisfies the user's expected cost. EC is calculated by EET and cost of the particular resource that vary dynamically. It is given in the Equation (4).

$$EC(i, j) = EET(i, j) \times \cos t_i \quad (4)$$

Then the resource is chosen that satisfies the user's expected cost. Grid scheduler scheduled the jobs to those resources which satisfies all the properties. Expected completion time (ECT) is calculated for knowing the makespan of the system. It is given in the Equation (5).

$$ECT(i, j) = \frac{\sum_{j=1}^{n} length}{MIPS_i} + EET(i, j) + DT(i, j) \quad (5)$$

Where,

$$DT(i,j) = \frac{length_j}{baudrate_i}$$

Overall execution of the system i.e., makespan is calculated by using the Equation (6)

$$makespan = \max(ECT_i) \quad (6)$$

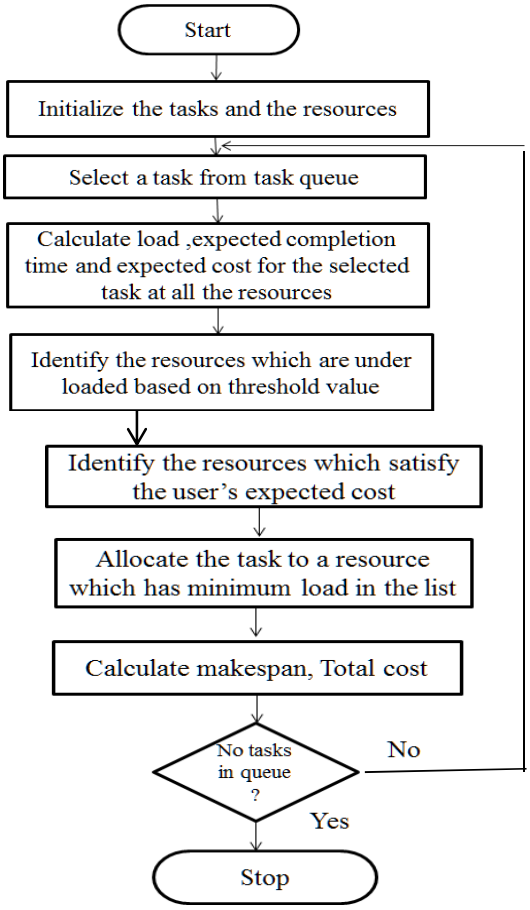
By using these strategies, the load balancing and pricing method can be integrated and the jobs can be efficiently allocate to their respective resources. The system performance will be improved and the user is also satisfied.

V. AN INTEGRATED LOAD BALANCING AND OPTIMAL PRICING ALGORITHM

In which load balancing and pricing are interrelated. The distribution of the tasks is depending upon the load, time and cost of the resources [9]. Both are dynamic variables. The pseudo code for the proposed algorithm is

STEP 1: BEGIN
STEP 2: GET grid resources
STEP 3: REGISTER grid resources to GIS
STEP 4: INPUT gridlets
STEP 5: SELECT the gridlets
STEP 6: CALCULATE load at each resources
STEP 7: CLASSIFY the resources whether it is under loaded, over loaded or normally loaded.
STEP 8: SELECT the resource which is under loaded
STEP 9: CALCULATE EET and EC of the resources
STEP 10: SELECT minimum EET and EC of the resource
STEP 11: ALLOCATE the gridlets those selected resources
STEP 12: CALCULATE makespan by calculating ECT
STEP 13: WRITE max (ECT)
STEP 14: IF task queue=0
STEP 15: END
STEP 16: ELSE
STEP 17: REPEAT steps 5 to 13 until the task queue is zero
STEP 18: END

The flow diagram of the proposed system is



VI. RESULTS AND DISCUSSION

6.1. Environment Creation

Table 1.Environment Creation Setup

Grid resources	5
Machines for each grid resources	3
PEs and MIPS rating for each resource	Random
Cost for each resource	Random
Gridlets	10
Length, file size, output size for each gridlets	Random

Table 1 shows the environment creation setup for grid system using GridSim. There are five resources in the grid system and each has three machines. The initial status of each resource is stored in Information Server. Grid user creates ten Gridlets. Job 1 arrives at job scheduler, and then the scheduler identifies the resource which is suitable for the particular job by calculating Expected Execution Time (EET), Expected Cost (EC) and Load of the resource. According to those results the jobs are allocated to it. The process will continue until all the jobs in the queue are allocated.

6.2. Parameter Evaluation

The number of gridlets are created for analysing the parameter of the system shown in Table 2.

- Resource Utilization:

The resources utilized in the system are known by calculating load to each resource and take average of it.

- Makespan:

Makespan is the overall execution time of the system. It is calculated by taking the maximum value of the ECT.

- Cost:

Cost is the price charged for used resources i.e., EC. It is calculated by EET and cost for each resource, based on that the user's satisfaction percentage is calculated for the system.

Table 2.Performance Analysis based on Parameters

Jobs	Resource Utilization (%)	User satisfaction (%) based on cost	Makespan (ms)
10	91	95	143
20	95	90	267
30	96	88	417

The resource utilization parameter is analyzed in percentage with the number of jobs like 10, 20 and 30. The utilization of the resource increases as the job increases because the resources are used efficiently. Figure 1 shows the resource utilization performance in the system.

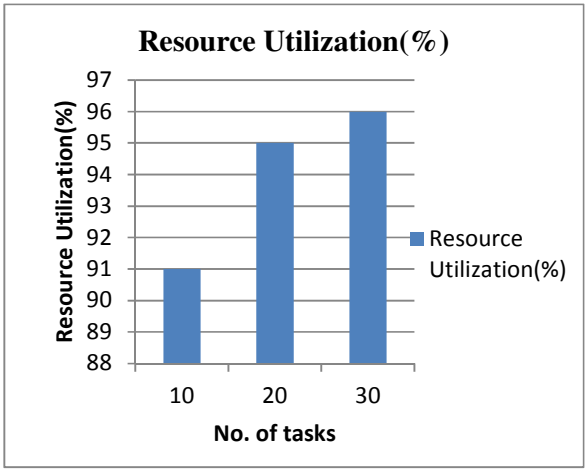


Figure 1.Performance based on Resource Utilization

The user satisfaction is analyzed with percentage based on the cost of the system. In which the user satisfaction decreased as the job increase because the cost of the system increase when the resource usage increase. Figure 2 shows the user satisfaction of the system with vary amount of jobs.

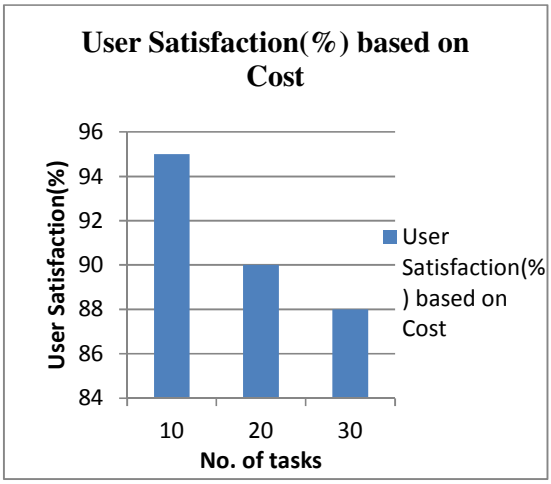


Figure 2.Performance of User Satisfaction based on Cost

The overall execution of the system is calculated in millisecond. The makespan increases as the job increase because the time taken for all the jobs to finish processing take more time. Figure 3 shows the makespan calculation of the system.

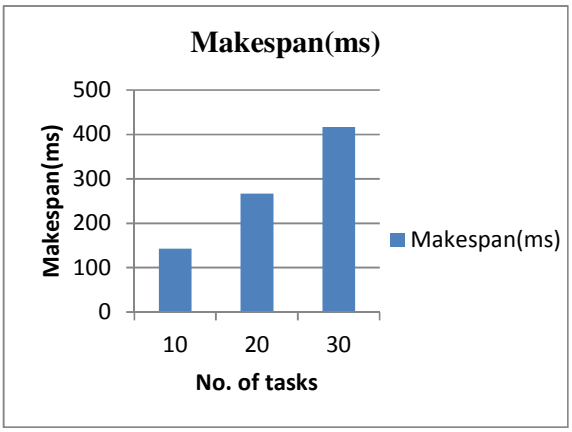


Figure 3 Performance based on Makespan

By analyzing these parameters, the system performance are increased by reduce the response time and overall cost of the system. It also maximizes the resource utilization rate and throughput with the user satisfaction.

VII. CONCLUSION

The important problem addressed in the system is scheduling and pricing to the computing systems. To overcome the problem, the system integrates load balancing and optimal pricing. It is determined by aggregating the information of processing speeds, time, load and prices of the computing resources. According to this information the scheduler submit their jobs to the resource that satisfies the user's expected cost. The strategy of the system improves the resource utilization rate, minimize the response time and overall cost of the system is reduced. In the future, the ability of a system to perform its function correctly even in the presence of faults i.e., fault tolerance can be considered because the

failure at job site has cascading effect on the grid performance. The dynamic nature of grid environment introduces challenging security. The new services like resources can be created by users dynamically without administrator permission. These services must synchronize and interact with other services. So, the security should provide by name the service with acceptable identity to the user i.e., user credentials.

REFERENCES

[1] BibhudaSahoo, SudiptaMohapatra and Sanjay Kumar Jena (2008), 'A Genetic Algorithm Based Dynamic Load Balancing Scheme for Heterogeneous Distributed Systems', Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications, PDTA, pp. 487-499.

[2] Bhatia Rashmi (2013), 'Grid Computing and Security Issues' ,International Journal of Scientific and Research Publications ISSN: 2250-3153, Vol-3, Issue 8, pp.1-5

[3] Dinesh Gawande, Rajesh Dharmik and ChandaPanse (2012), 'A Load Balancing in Grid Environment', International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol-2, Issue 2, pp.445-450.

[4] Ghosh Roy, Basu and Das S.K. (2004), 'A Game Theory Based Pricing Strategy for Job Allocation in Mobile Grids', Proc.18th IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS), pp. 26-30.

[5] Kokilavani and George Amalarethinam (2011), 'Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing', International Journal of Computer Applications (0975-8887), Vol-20, No.2, pp. 43-49.

[6] Leyli Mohammed Khanli and BehnazDidevar (2011),'A New Hybrid Load Balancing Algorithm in Grid Computing Systems', International Journal of Computer Science Emerging Technologies, Vol-2, No.5, pp. 304-309.

[7] Mohsen AminiSalehi, HosseinDeldari and BahareMokarramDoori (2008), 'Balancing Load in Computational Grid Applying Adaptive Intelligent Colonies of Ants', Informatica32, pp. 327-335.

[8] NeerajPandey, Shashi Kant Verma and Vivek Kumar Tamta (2013),' Load Balancing Approaches in Grid Computing Environment', International Journal of Computer Applications (0975 - 8887), Vol-72, No.12, pp. 42-49.

[9] Penmatsa S. and Chronopoulos A.T. (2005), 'Job Allocation Schemes in Computational Grids Based on Cost Optimization', Proc. 19th IEEE Int'l Parallel and Distributed Processing Symp.(IPDPS), pp. 515-524.

[10] Qin Zheng and BharadwajVeeravalli (2014), 'On the Design of Mutually Aware Optimal Pricing and Load Balancing Strategies for Grid Computing Systems', IEEE transactions on computers, Vol-63, No.7, pp. 407-419.

[11] Said Fathy and El-Zoghdy (2012), 'A Capacity -Based Load balancing and Job migration algorithm for heterogeneous computational Grids', International Journal of Computer Networks & Communications (IJCNC), Vol-4, No.1, pp. 113-125.

[12] Tang X. and Chanson S.T. (2000), 'Optimizing Static Job Scheduling in a Network of Heterogeneous Computers', Proc. Int'l Conf. Parallel Processing (ICPP), pp. 373-384.

[13] Visalakshi P. and Sivanandam S.N. (2009), 'Dynamic Task Scheduling with Load Balancing using Hybrid particle Swarm Optimization', International JournalOpenProblems Comput.Math., Vol-2, No.3, pp. 475-488.

