Deepika Sharma al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 1, Issue 2, Sept. 2014, pp. 5-9

Evaluating Heuristic based Load Balancing Algorithm through Ant Colony Optimization

Deepika Sharma[1], Dr. Surjeet Dalal[2]

[1]Student, M. Tech, ESEAR, Ambala

[2]Associate Professor, Dept. of CSE, E-Max group of Institutions, Ambala

Abstract— **Grid is an infrastructure that involves the integrated and collaborative use of computers, networks, databases and scientific instruments owned and managed by multiple organizations. Computational grid has been considered as the best paradigm for handling large scale distributed system having geographically allocated resources. Load balancing algorithms are important in the research of network applications. In this paper we present an algorithm which reduces the average execution time and cost of the tasks. This method considers both cost and time constraints. This algorithm provides services like resource discovery. For evaluation purpose a comparison of execution times and cost of proposed algorithm and the other similar algorithm is also provided in this paper. Results support the proposed approach.**

Keywords— **Grid, load balancing, heuristic Load balancing algorithm, Gridsim, computation time.**.

I. INTRODUCTION

Poster and Kesselman introduced grid computing as a way to utilize the geographically distributed available idle workstation's computation power to remote grid users for the execution of their computation requiring jobs or tasks or processes. Grid Computing enables sharing, selection, aggregation of geographically distributed resources dynamically at run time depending on their accessibility, ability and users Quality of Service requirements The user essentially interacts with a resource broker that hides the complexities of Grid computing [4,5]. The broker discovers resources that the user can access using information services, negotiates for access costs using trading services, maps tasks to resources (scheduling), stages the application and data for processing (deployment), starts job execution, and finally gathers the results. It is also responsible for monitoring and tracking application execution progress along with adapting to the changes in Grid
runtime environment conditions and resource failures as the grid resource utilization is gaining significance various scheduling methods or load balancing across the grid is required in order to improve the performance of the grid system. While balancing the load, certain type of information such as the number of jobs waiting in queue, job arrival rate, CPU processing rate, and so forth at each processor, as well as at neighboring processors, may be exchanged among the processors for improving the overall performance. Based on the information that can be used, Scheduling is mainly classified into two types static and dynamic scheduling or adaptive [7,10].Static approach assumes that a prior information about all the characteristics of the jobs, the computing devices and the communication network are known and provided. Load balancing decisions are made deterministically or probabilistically at compile time and remain constant during runtime. Dynamic load balancing algorithm attempts to use the runtime state information to make more informative decision in sharing the system load. However, dynamic scheme is used in modern load balancing method a lot due to their robustness and flexibility. Dynamic algorithm is characterized by parameters such as i) Centralized vs. Decentralized. An algorithm is centralized if the parameters necessary for making the load balancing decision are collected at, and used by, a single device. In decentralized approach all the devices are involved in making the load balancing decision. Scalability and fault tolerance are the advantages of decentralized algorithms.. ii) Cooperative vs. Non-cooperative. An algorithm is said to be cooperative if the distributed components that constitute the system cooperate in the decision-making process. Otherwise, it is non-cooperative.and iii) Adaptive vs. Non-adaptive. If the parameters of the algorithm can change when the algorithm is being run, the algorithm is said to be adaptive (to the changes in the environment in which it is running). Otherwise, it is non-adaptive. The characteristic of grid makes resource scheduling even more challenging in the computational grid environment. In this paper we propose a new load balancing algorithm which is heuristic in nature. Unlike previous algorithms which considered the system load of grid nodes or the completion time for a job but don't take into account job personal resource requirements (such as cost , QOS of a node) . In this paper we present an algorithm which considers the job size and is mainly focused on formulating a decentralized heuristic based load balancing algorithm for resource scheduling. The rest of the paper is organised as follows: Section II gives a detailed Literature review. Section III deals with the system model of the Grid. Section IV will describe the proposed algorithm concept and design. Section V will show the simulation experiment and results, and finally Section VI will conclude the whole paper.

II. RELATED WORK

Many researchers have proposed scheduling algorithms for parallel and distributed systems as well as Grid computing

environment. For a dynamic load balancing algorithm, it is unacceptable to exchange state information frequently because of the high communication overhead. In order to reduce the communication overhead Martin et al.[18] studied the effect of communication latency, overhead and bandwidth in cluster architecture for application performance. Distributed network computing environments have become a cost effective and popular choice to achieve high performance and to solve large scale computation problems. There are many types of algorithms that have been used in resource balancing in Grid computing system. Martino and Mililotti addresses the use of Genetic Algorithm(GA) and Tabu Search(TS) to solve the grid load balancing problem in the dynamic environment [5]. These algorithms have their limitations like these algorithms require extra storage and processing requirement at the scheduling nodes. Grid computing involves coupled and coordinated use of geographically distributed resources for purposes such as large scale computation and distributed data analysis [17].

A new hybrid load balancing policy can be integrated in static and dynamic load balancing techniques with the objective to allocate effective node, identify the system imbalance immediately when a node becomes unable to participate in task processing. Ibarra and Kim in [15] used a combination of intelligent agents and multi-agent that work in grid load balancing area. In static grid load balancing, the iterative heuristic algorithm is better than the FCFS algorithm. Grid infrastructures are dynamic in nature in the sense of resource availability and hence a changing network topology. Resource heterogeneity and network heterogeneity also exits in Grid environment. Scheduling jobs onto resources is NP hard problem. So, we need an algorithm that consider optimization in terms of time and cost for efficient scheduling and execution. Current grid resource scheduling algorithms are mainly based on User-Directed Assignment (UDA), Minimum Completion Time (MCT), Minimum Execution Time (MET), Min-Min [5], Max-Min, heuristic algorithm, genetic algorithm (GA)[5], Ant Algorithm (AA) [6], multi-Agent, computational economy model [7].In [19] authors proposed a dynamic load balancing algorithm which considers CPU length, CPU and memory utilization and network traffic as load metric.

Although [6][14] presented scheduling algorithms based on ant algorithm, but the authors just analyzed ant algorithm based classified task scheduling method and ACA based scheduling without considering about the price attribute. In this paper, price factor and time factor are taken into consideration, and the objective is to save total execution time and cost.

III. SYSTEM MODEL OF THE GRID

Grid system consists of sharing of heterogeneous resources (like different capacity processors, memory, networks etc) [2].In order to maximize the performance of computational Grid environment, it is essential to distribute the load on different grid nodes. In grid computing environment, there exists more than one resource to process jobs. One of the main challenges is to find the best or optimal resources to process a particular job in term of minimizing the job computational time. Optimal resources refer to resources having high CPU speeds and large memory spaces. Computational time is a measure of how long that resource takes to complete the job. In order to maximize the performance of computational Grid environment, it is essential to distribute the load on different grid nodes. Following are the members of Grid system:-

GridResource Broker: It is an important entity of a grid. A grid resource broker is connected to an instance of the user.Each grid job (composed of gridlets or tasks) is first submitted to the broker, which then schedule the grid job according to the user scheduling policy.

GIS: Grid resource is a member of grid. All the available resources register themselves to the GIS (Grid information Service).GIS contains all information about all available resources with their computing capacity and cost at which they offer their services to grid users. All Grid resources that join and leave the grid are monitored by GIS. Whenever a Grid broker has jobs to execute, it consults GIS to give information about available grid resources.

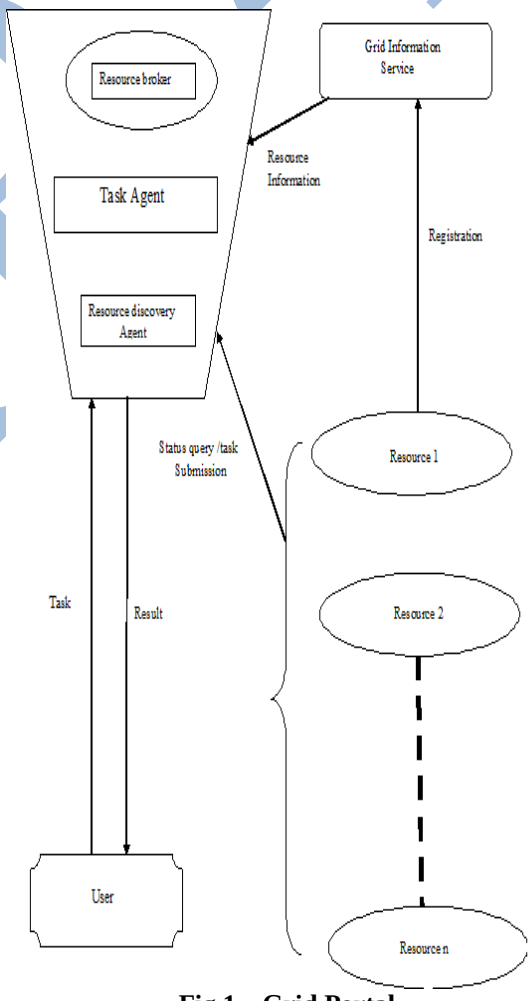

Fig.1. Grid Portal

Grid Use: They register themselves to the GIS by specifying the QOS requirement such as cost of computation, deadline to complete the execution, the number of processors, speed of processing of internal scheduling policy and time zone.

Task Agent: TA, by deploying Ant algorithm, select a resource for next task assignment and dispatch the task to

selected resource through RMA. After task execution, results are received from resources and are returned to user by TA.

Resource discovery Agent: RDA queries GIS regarding resources. It send query to registered resources for their availability status and gets the status information and makes it available to TA. Every task is dispatched through RDA.

Since in a Grid environment, the network topology is varying, our model captures this constraint as well by considering an arbitrary topology. The data transfer rate is not fixed and varies from link to link. The aim of this paper is to present load-balancing algorithms adapted to heterogeneous Grid computing environment. In this paper, we attempt to frame an adaptive decentralized sender-initiated load-balancing algorithms for computational Grid environments. Interaction among the entities is shown in the Fig.1.

IV. HEURISTIC LOAD BALANCING ALGORITHM

Deadline plays an important role in load balancing.An algorithm for load balancing is successful, ifthe task executes within the deadline.All the tasks in this algorithm are distributed according to Ant Colony Algorithm. HLBA is inspired on an analogy with real life behaviour of a colony of ants when looking for food, and is effective algorithm for the solution of many combinatorial optimization problems. Investigations show that: Ant has the ability of finding an optimal path from nest to food. On the way of ants moving, they lay some pheromone on the ground. While an isolated ant moves essentially at random, an ant encountering a previously laid trail can detect it and decide with high probability to follow it, thus reinforcing the trail with its own pheromone. The probability of ant chooses a way is proportion to the concentration of a way's pheromone. In HLBA, the pheromone is associated with resources rather than path. The increase or decrease of pheromone depends on task status at resources. The main objective of algorithm is reduction in total cost and execution time. Theprocess of Heuristic based Load Balancing Algorithm is shown below:

Let the number of tasks (ants) in task set *T* maintained by task agent is *P* and the number of registered resources is *Q*. When a new resource *i R* is registered to GIS, then it will initialize its pheromone based on:

$$\tau_i(0) = N + M$$

where *N*represent the number of processing elements and *M* corresponds to MIPS rating of processing element. Whenever a new task is assigned to or some task is returned from R_i, then the pheromone of $R_i$is changed as:

$$i\tau_i^{new} = \rho * \tau_i^{old} + \Delta\tau_i$$

where$\Delta_i$is pheromone variance and ρ, $0 < \rho < 1$ is a pheromone decay parameter. When a task is assigned to R_i, its pheromone is reduced i.e. $\Delta_i = -C$, where*C*represents the computational complexity of assigned task.When a task is successfully returnedfrom R_i, $\Delta_i = \Phi * C$, where Φ is the encouragement argument. Pheromone increases when task execution at a resource is successful.The possibility of next task assignment to resource $R_j$is computed as:

$$p_j(t) = \frac{[\tau_j(t)]^{\alpha} + [\eta_j]^{\beta}}{\sum_r [\tau_j(t)]^{\alpha} + [\eta_j]^{\beta}}$$

where*j, r* ∈*available resources*. $\tau_j(t)$ denotes the current pheromone of resource $R_j$and $\eta_j$represents the initial pheromone of R_ji.e.$\eta_j = \tau_j(0)$. αis the parameter on relative performance of current pheromone trail intensity , β is the parameter on relative importance of initial performance attributes.

So the algorithm is designed in order to influence the performance of the system which depends upon several factors and these factors could be simplified for reducing the complexities of the method to be used.Heuristic Load Balancing Algorithm is presented below:

Phase 1 (Initialization Phase)

Begin

Initialise all parameters including resources(processing elements, MIPS rating), pheromone intensity,Task set etc.

Phase 2 (Operational Phase)

While (Task setT !=Φ)

Begin

Select the next task't' from Task set T.

Determine next resource Rifor task assignment having high transition probability (or pheromone intensity).

$p_i(t) = max\ p_l(t)$

$l \in q$

Phase 3 (Result Phase)

Schedule task 't' to Ri and update Task set by T = T - {t}.

If any node fails or complete its executionpart update its pheromone intensity of that corresponding resource.

END While

END

All the tasks are allocated to the resources depending on the value of pheromone or transition probability, which varies according to the status of the task at a particular resource. Resource having highest pheromone intensity would get the task before any other resource having lower than that intensity value. The algorithm is further elaborated by flowchart as given in Fig.2:

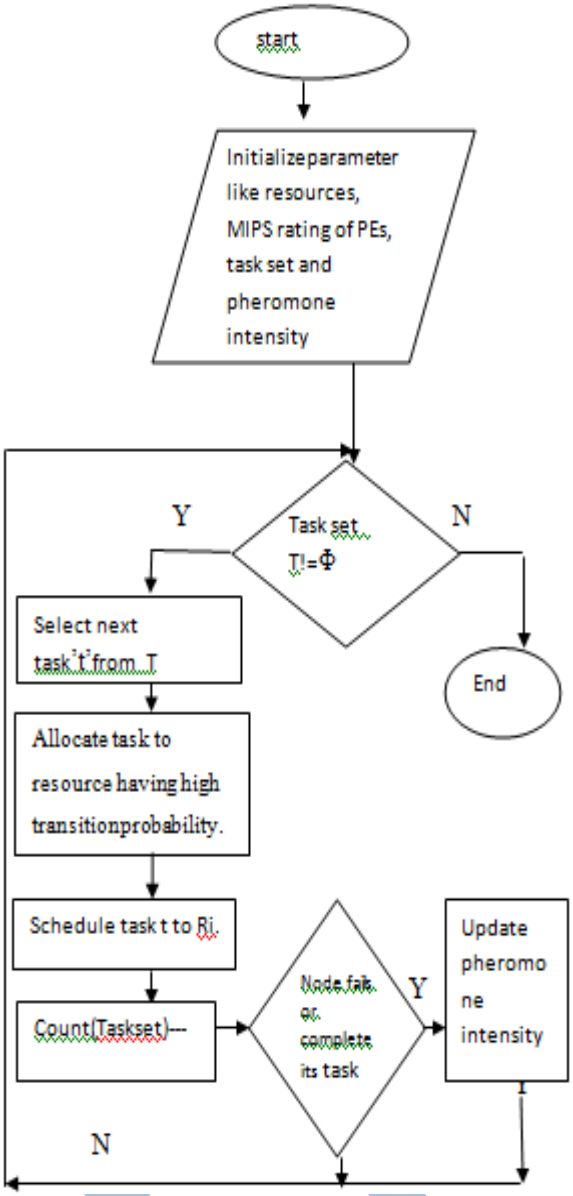

Fig.2. Flowchart for Heuristic Load Balancing Algorithm

V.SIMULATION RESULTS

Simulation results are usedto compare the performance of the proposed approach and earlier developed approach. For simulation GridSim simulatoris used which is a toolkit for the modelling and simulation of distributed resource management and scheduling for Grid computing [11]. In order to test the efficiency of the improved load balancing approach a comparison is done for the execution time (time utilized by PE for processing a task) of heuristic load balancing algorithm and non-heuristic load balancing algorithm.

Two experiments have been conducted to evaluate the performance of the proposed algorithm in terms of their processing time. First experiment processed 50-200 jobs with fixed number of resources taken as 10 while the second experiment processed 10 jobs with varying number of resources from 5-20. Details of the scheduling parameters are shown in the Table 1.

Table 1
Scheduling Parameters for the Experiments

Experiment	Ist	2nd
No. of machines per resource	1	1
No. of Processing Elements(PEs) per machine	1-5	1-5
MIPS rating for PEs	10 or 50 MIPS	10 or 50 MIPS
Bandwidth	1000 or 5000	1000 0r 5000

Average computation time (in seconds) has been used to compare the performance of algorithms. Fig. 3 depicts the comparison between the average computation times of the algorithms for the first experiment. In this number of resources remain fixed as 10 and the number of tasks varied from 50 to 200. It can be seen that the proposed algorithm labelled as Heuristic based load balancing algorithm has the lower execution time as compared to Non-Heuristic based load balancing algorithm at different instants.

A comparison between the computation times for the second experiment is shown in Fig. 4. Heuristic load balancing algorithm still performs better than non-heuristic load balancing algorithm. The performance of Heuristic approach is better because the communication that occur when each ant taking a step for searching the best resource is less when processing a large amount of jobs. Each ant will carry the history of visited node and will refer to it to find the best resource to process the job.

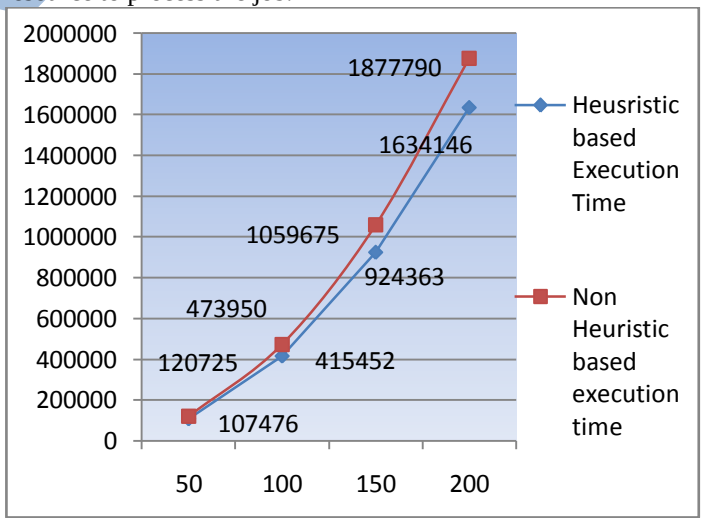

Fig. 3. Comparison Among the Execution Times of Heuristic and Non-Heuristic Algorithms.

In the second experiment the number of task remain fixed as 10 and number of resources varied from 5 to 20 and again compute the total execution time(in seconds) for different number of resources. Result supports the use of HLBA for time efficient execution of tasks thenNHLBA.

Fig. 3 and Fig. 4 depict the result of the total execution time for two different scenarios. Results support the proposed approach by reducing the total execution time of jobs.

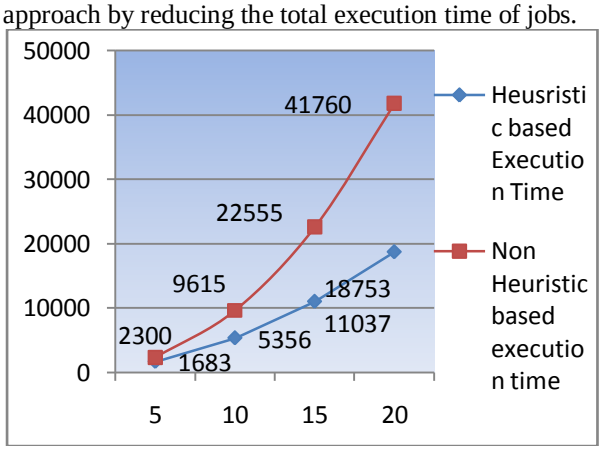

Fig. 4. Comparison Among the Execution times of Heuristic and Non-heuristic Load Balancing Algorithms for Experiment 2.

VI. CONCLUSION

In this paper an algorithm that is adaptive, decentralizedand distributed for load balancing among the aggregated resources in the grid has proposed. Through a series of simulations by varying the number of tasks and resources we obtain the results. It is found out that computation decreases until it reaches the optimal level i.e resources with maximum computation capability.When the numberof users competingfor the same set of resources increases,there will be proportional impact on others dependingon each user's strategies and constraints. Apart from complementing and strengthening the results of the proposed method, the simulations demonstrate the capability of GridSim and the ease with which it can be used for evaluating performance of new scheduling algorithms.Resultof the simulation shows thatHLBA enhances the performance of the system by reducing theexecution time of the jobs.

REFRENCES

[1]. I. Foster, and C. Kesselman, "Globus: a metacomputing infrastructure toolkit", International Journal of High Performance Computing Applications, vol. 11, no. 2, pp. 115-128, 1997.

[2]. I. Foster, and C. Kesselman(editors), *The Grid 2: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, USA, 2003.

[3]. I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: enabling scalable virtual organizations", *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 200-222, 2001.

[4]. X.S. He, X.H. Sun, and G.V. Laszewski, "QoS guided min-min heuristicfor grid task scheduling", *Journal of Computer Science & Technology*, vol.18, no. 4, pp. 442-451, 2003.

[5]. V.D. Martino, and M. Mililotti, "Scheduling in a grid computing environmentusing genetic algorithms", Proceedings of the 16th InternationalSymposium on Parallel and Distributed Processing, pp. 235-239, 2002.

[6]. Z.H. Xu, X.D. Hou, and J.Z. Sun, "Ant algorithm-based task schedulingin grid computing", Proceedings of 2003 IEEE Canadian Conference onElectrical and Computer Engineering, vol. 2, no. 3, pp. 1107-1110, 2003.

[7]. R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economicmodels for resource management and scheduling in Grid computing",Concurrency and Computation: Practice and Experience, vol. 14, no. 2, pp.1507-1542, 2002.

[8]. R. Buyya, D. Abramson, and S. Venugopal, "The Grid Economy",Proceedings of the IEEE, 3, vol. 93, no. 3, pp. 698-714, 2005.

[9]. O.O. Sonmez, and A. Gursoy, "A novel economic-based schedulingheuristic for computational grids", *International Journal of High PerformanceComputing Applications*, vol.21, no. 1, pp. 21-29, 2007.

[10]. R.Buyya, and ManzurMurshed, "GridSim: A Toolkit for theModeling, and Simulation of Distributed Resource Management, andScheduling for Grid Computing", Concurrency and Computation: Practiceand Experience, vol. 14, no. 1, pp. 1175-1220, 2002.

[11]. Li Hao,"Implement of Computational Grid Application SchedulingSimulation with GridSim Toolkit", *Journal of Jilin Normal University*(Nature Science Edition), vol. 3, no. 1, pp. 63-64, 2003.

[12]. Wang yue, and Tao Hongjiu, "Application in TSP Based on Ant ColonyOptimization", *Journal of Wuhan University of Technology* (Informationand Managing EngineeringEdition),vol. 28, no. 11,pp. 24-26, 2006.

[13]. Marco Dorigo, and Luca Maria Gambardella, "Ant colonies for thetraveling salesman problem", BioSystems, vol. 43, no. 2, pp. 73-81, 1997.

[14]. Chen Ye, "Ant Colony System for Continuous Function Optimization",Journal of Sichuan University (Engineering Science Edition), vol. 36, no. 4, pp. 117-120, 2004.

[15]. O. Ibarra and C. Kim, "Heuristic algorithms for scheduling independent tasks on nonidentical processors",*Journal of the ACM (JACM)*, vol. 24, no. 2, pp. 280-289, 1977.

