VOL. 2, NO. 11, October 2011 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

Proffering a Brand New Method for Mapping of Tasks to Resource in Grid Computing with the Aims of Load Balancing Based on P2P Model Uses Ant Colony Algorithm

Ali Sarhadi

[1] Department of Computer Engineering, Toyserkan Branch, Islamic Azad University, Toyserkan Iran
Sarhadi_ali@iau-malayer.ac.ir

ABSTRACT

Computational Grid is a new paradigm in distributed computing which aims to realize a large-scale high performance computing environment over geographically distributed resources. Resource management and load balancing are key concerns when implementing grid middleware and improving resource utilization. The main goal of load balancing is to provide a distributed, low cost, scheme that balances the load across all the resources. To improve the global throughput of Grid resources, effective and efficient load balancing algorithms are fundamentally important. This paper identifies the issues in task mapping based on ant colony algorithm with the aims of load balancing. To show the effectiveness of the proposed method, computer simulations have been used. The results of simulation show that the performance of proposed algorithms has a significant improvement.

Keywords*: Load balancing, ant colony, task mapping, p2p*

1. INTRODUCTION

Grid technologies [2] [10] have emerged to enable large scale flexible resource sharing among dynamic virtual organizations. Computational grids are expected to offer dependable, consistent, pervasive, and inexpensive access to high-end resources irrespective of their physical location and the location of access points. Grid Systems allow applications to assemble and use collections of resources on an as-needed basis, without regard to its physical location. Grid middleware and other software architecture that manage resources have to locate and allocate resources according to application requirements. They also have to manage other activities like authentication and process creation that are required to prepare tasks mapping [8]. In brief, Grid resource management faces to two major problems; one is matching computational needs to appropriate resources, and the other is exploiting resources over highly dynamic environment [8]. In this paper, our main concern is to address the implications imposed by heterogeneity of applications and resources in computational Grid by means of ant colony algorithm based on resource load balancing. The matching of tasks to machines and scheduling the execution order of these tasks is referred to as mapping [9].

There are some of the challenges involved in the mapping problems include resource heterogeneity and resource load balancing.

2. RELATED WORK

Most load balancing approaches are orientated towards application partitioning via graph algorithms [8]. However, they do not address the issue of reducing migration cost, which is the cost entailed by load redistribution, which can consume much more time than the actual computation of a new decomposition. Some works [9] have proposed latency - tolerant algorithm that takes advantage of overlapping the computation of internal data and the communication of incoming data to reduce data migration cost. Unfortunately, it requires applications to provide such a parallelism between data processing and migration, which restricts its applicability. There is large number of load balancing techniques and heuristics, presented in literature, target only homogeneous resources. However, modern computing systems, such as the computational Grid, are most likely to be widely distributed and strongly heterogeneous. Therefore, it is essential to consider the impact of these characteristics on the design of load balancing techniques.

The traditional objective, when balancing sets of computational tasks, is to minimize the overall execution time called *makespan*. However, in the context of heterogeneous distributed platforms, makespan minimization problems are in most cases *NP-complete*. In addition, when dealing with large scale systems, an absolute minimization of the total execution time is not the only objective of a load balancing strategy. The communication cost, induced by load redistribution, is also a critical issue. For this purpose, Yagoubi proposes in [11], a hierarchical load balancing model as a new framework to balance computing load in a Grid. Unfortunately, the root of the proposed model can become a bottleneck.

We propose, in this paper, a new load balancing method to address the new challenges in Grid computing.

Comparatively to the existing works, the main characteristics of our strategy are:

1) New organization of resource information node based on p2p model has been introduced.

2) It uses ant colony technique for transfer of request from a node to another node in information node organization.

3. LOAD BALANCING

A typical distributed system involves a large number of geographically distributed worker nodes which can be interconnected and effectively utilized in order to achieve performances not ordinarily attainable on a single node. Each worker node possesses an initial load, which represents an amount of work to be performed, and may have a different processing capacity.

To minimize the time needed to perform all tasks, the workload has to be evenly distributed over all nodes which are based on their processing capabilities. This is why load balancing is needed.

The load balancing problem is closely related to scheduling and resource allocation. It is concerned with all techniques allowing an evenly distribution of the workload among the available resources in a system [1]. The main objective of a load balancing consists primarily to optimize the average response time of applications; this often means the maintenance the workload proportionally equivalent on the whole system resources.

Load balancing algorithms can be classified into two categories: static or dynamic. In static algorithms, the decisions related to load balance are made at compile time when resource requirements are estimated. And Dynamic load balancing algorithms make changes to the distribution of work among nodes at run-time; they use current or load information when making distribution decisions.

3.1. PROPOSED MODEL FOR LOAD BALANCING BASED ON P2P ORGANIZATION

Our model is based on a p2p organization of grid information node. In this paper a flat structure for organization of information nodes is proposed. All grid resource information is registered in information nodes. The Proposed structure is an efficient and extendable environment. It does create a dynamic network of directories. In this structure for mapping of tasks to resources and finding of appropriate resources for incoming tasks peer to peer method is proposed. Directory is an aim of information node or peer in peer to peer search method. Peer to peer resource search method is fully distributed. In this way information's nodes that participate in resource search are equally in terms of importance. That means each information node can be process every query search and perform the search. This structure is illustrated in fig 1.

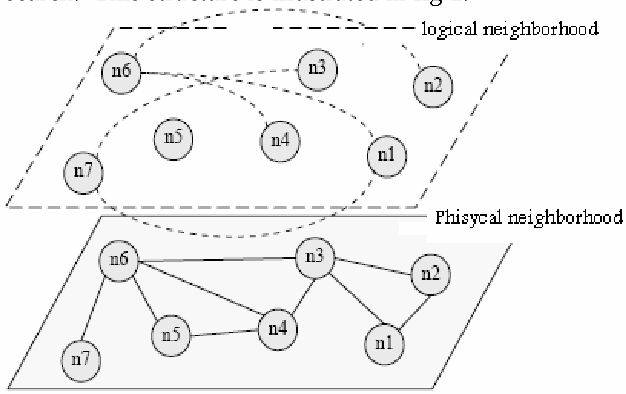

Fig. 1: P2P structure for information node organization.

Grid from resource searching perspective is a set of information nodes that are geography distributed. Each information node can be attached to other information nodes through contacting with them. In peer to peer search method each information node knows limited set of information nodes that associated with them. These set known as neighborhood of this node. These neighborhoods are logical neighbors' not physical ones. Note to fig 1.

Users send their request to known nodes. If any information about the location of required resource in user request exists in local node, this nodes return information to user, otherwise request sends to one of the neighbors. This operation repeated until appropriate resource was found or TTL is valid. A Time-to-Live (TTL) value is used to manage the number-of-hop permitted in any query search. In this structure load balancing can be defined as the workloads of all information nodes are balanced. In this paper for solving load balancing problem in proposed p2p structure of information node ant colony algorithm is used.

VOL. 2, NO. 11, October 2011 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

4. ANT COLONY ALGORITHM

Ant colony optimization (ACO) is a Meta heuristic alternative for solving the complicated optimization problems [4]. The base of this algorithm is based on the mass movement of ants in the nature. To attain to the food, initially, an ant begins randomly moving and along its route, it will leave a material called pheromone [3]. Passing the time, it will increase its content to such extent that resulted in using the same route by other ants to attain their goal.

ACO was expressed for the first time in a PhD thesis for solving the problem of hawker seller [15]. In this problem we are going to find the shortest route in the weighted graph such that all nodes can be met only once. In this problem, ants will be put on different nodes of graph. Any ant may choose its next node randomly and by calculating the special probability function according to eq. (1). In any repeat in the problem, such choices will be continued to such extent that all nodes could be surveyed.

$$P_k(i,j) = \frac{[\tau(i,j)]^{\alpha}.[\eta(i,j)]^{\beta}}{\sum_{l \in N_i^K}[\tau(i,l)]^{\alpha}.[\eta(i,l)]^{\beta}} \quad (1)$$

Where artificial ant K in node i will choose node j, with probable value $P_k(i,j)$. In addition, $\eta(i,j) = \frac{1}{d(i,j)}$ is a heuristic function where $d(i,j)$ is he distance between two towns i and j. N_i^k indicates the neighborhoods of node i where ant K has not yet met it. $\tau(i,j)$ is also indicating the rate of pheromone existing in the route. Meanwhile, the rate of pheromone present in a route may not be remained in the same size and may be changed according to eq. (2) [5].

$$\tau(i,j) = \rho.\tau(i,j) + \sum_{k=1}^{m} \Delta\tau_k(i,j) \quad (2)$$

Where ρ is the evaporation rate of pheromone and m is the number of ants. The $\Delta\tau_k(i,j)$ which is equal with the size of ants remained, will be calculated according to equation (2).

$$\Delta\tau_k(i,j) = \begin{cases} L_k & \text{If arc (i,j) is used by ant k} \\ 0 & \text{Otherwise} \end{cases}$$

According to Elitist algorithm, at the end of choosing whole nodes by any ant, the ant that could find the best rout allowed to be updated. It must be noted that in all problems that are being solved by any way using ant colony optimization, choosing the parameters is very important.

5. UPDATING PHEROMONE TABLE WITH AIMS OF LOAD BALANCING

Load balancing is a key point for updating pheromone table of each node. That means pheromones updating are based on load balancing. Here how to pheromone table updating is explained. The node that is nearer to resource from a node that is farther to it is more suitable node. When searching process is reached to ending node, in case of being a successful searching process, answers are produced and sent to initial and middle nodes. These answers are as follow:

$$U_{ij} = \frac{1}{d_i^{rel}} \; \forall i \in \{1,2,\ldots,e-1\} \quad (3)$$

That d_i^{rel} is relative distance expressed in eq. (3). Then the values of P_{ij} (P_{ij} is associated score with R_j type in n_i node) will be updated. As explained above, pheromone table updating is based on load balancing. We define a general mapping from the task domain *k=1,..,τ* to the resources domain *j=1,…,μ* at iteration *n*. The load of each resource, which is denoted by L^n(j), is defined as the time taken to execute all the assigned tasks:

$$L^n(j) = \sum CT(k,j) \quad 1 \le j \le \mu, 1 \le k \le \tau$$

CT (k,j) define complete time of task k with resource k.

So L^n(j) should be involved in the updating of pheromone table. Therefore the reverse relation of this equation should be entered in updating of pheromone table, so that it causes the updating pheromone table in accordance with load of each resource. Reverse relation of load parameter is showed in bellow.

$$K^n(j) = \frac{1}{\sum CT(k,j)}$$

Therefore pheromone tables in each n_i node is updated according to the eq.(4).

$$P_{ij}(t) = P_{ij}(t-\alpha) * f(\alpha) + U_{ij} + K^n(j)$$
$$\forall i \in \{1,2,\ldots,e-1\} \quad (4)$$

$$0 <= f(t) <= 1, \qquad f(t) = \frac{1}{t}$$

α is time distance to the last time of updating score value. f(t) is evaporation coefficient of pheromone and is related to environment dynamics.

VOL. 2, NO. 11, October 2011 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

6. SIMULATION TOOLS

In order to evaluate the practicability and performance of proposed strategy, we have implemented it on the **GridSim V4.0** simulator [7], which we have extended to support simulation of varying Grid load balancing problems. GridSim is a Java-based discrete-event Grid simulation toolkit. It provides a comprehensive facility for simulation of different classes of heterogeneous resources, users, applications, resource brokers, and schedulers. In GridSim, application tasks/jobs are modeled as Gridlet objects that contain all the information related to the job.

7. SIMULATION ASSUMPTION

(1) Experiments are performed in environment with 1000 to 5000 information node.
(2) There is 10000 different type of resource in grid environment.
(3) It is assumed that the multiplicity of resource is same and there is 100 sample of each resource in grid environment.
(4) The bandwidth of all links can be considered constant.
(5) TTL is equal to 100.
(6) In each experiments 10 information nodes randomly selected and to each of them independently set of 200 requests are sent.

8. PERFORMANCE EVALUATION

All the experiments and simulation software have been performed on 4 GHz P4 Intel Pentium with 2 GB main memory, running on Linux fedora operating system. Reported results, is for average 20 times of simulation. Grid resource is high heterogeneous. We neglect the network topology and the communication costs associated with it. Instead, we assume that each of the users can submit applications to any of the resources.
The experiments were performed, on the basis of variation of several performance parameters in a Grid, namely the number of clusters, their worker nodes and the number of tasks. We have focused on the following objectives for performance evaluation of proffered method:

1- Average wait time: this parameter is obtained by a report from the average waiting time after load balancing with proposed method and the average waiting time before load balancing for 100, 200 and 500 incoming task**.**

2- Average execution time*:* this parameter is expressed as average execution time after load balancing with proposed method and the average execution time before load balancing for 100, 200 and 500 incoming task.

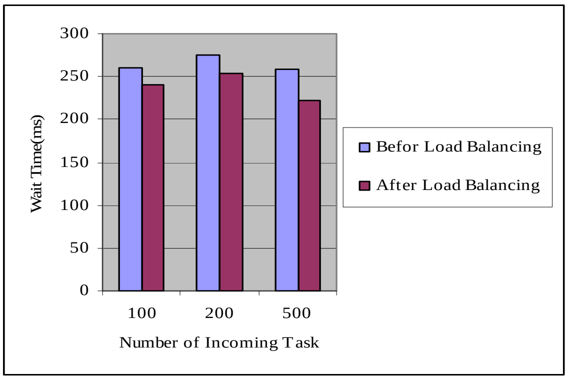

Fig. 2: Average wait time comparison

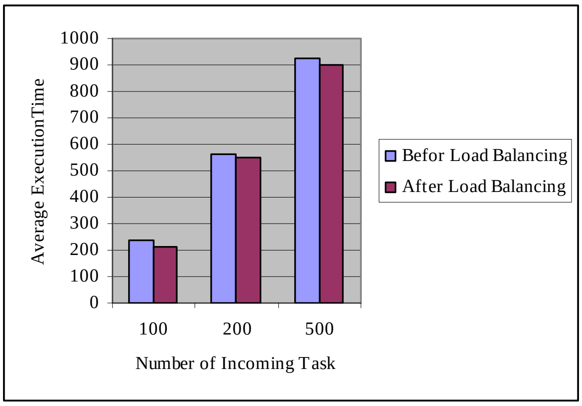

Fig. 3: Average execution time comparison

9. EXPERIMENTAL RESULT

In the first set of experiments, we focused to the average wait time according to the number of tasks before and after load balancing. As the figure 2 show in practically all cases we obtained an improvement in wait time after implementation of proposed method for load balancing about 20 percent. Number of tasks is varying from 100 to 500. The best results are obtained for a number of tasks equal to 500, which leads us to think that proposed load balancing method is interesting if we have a large number of tasks. During the second experiment, we interested to average execution time. We varied the number of tasks from 100 to 500. Results of the experiments are gathered in Fig. 3 This result is explained by the fact that we obtain an improvement in execution time after implementation of proposed load balancing method.

10. CONCLUSION

This paper addressed the problem of load balancing in Grid computing. We proposed a new approach based on ant colony algorithm for load balancing problem which takes into account the heterogeneity of the resources that is completely independent from any physical architecture Grid.

On the basis of this model, defined method has two main objectives:
(i) The reduction of the average wait time of tasks submitted to a Grid computing;
(ii) The reduction of the average execution time during tasks execution.

Proposed strategy uses a p2p model for organization of information nodes. Each information node contains all information about registered resource in grid environment. To implement the proposed strategy, we used the GridSim simulator. The fundamental advantage of simulators is their independence from the execution platform. We can simulate a distributed system mechanism of 10000 nodes on a single PC. This advantage is made possible because the simulator does not run in a real distributed system, but in a model of this matter.

REFERENCES

[1] A. E. El-Abd, "Load Balancing in distributed computing systems using fuzzy expert systems," Proceedings of international conference of Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2002, pp.141-144.

[2] I. Foster, C. Kesselman and S. Tuecke, "The anatomy of the Grid: Enabling scalable virtual organizations", International Journal of Supercomputer Applications, 2008.

[3] M. Dorigo and L. M. Gambardella, "Ant Colony System: A cooperative Learning Approach to the Traveling Salesman Problem", In IEEE Transactions on Evolutionary Computation, Vol.1, No.1, 2006.

[4] M. Dorigo, V. Maniezzo and A. Colorni, "The Ant System: Optimization by a colony of cooperating agents", In IEEE Transactions on Systems, Man, and Cybernetics Part B: ybernetics, 26(1):29--41, 2005.

[5] M. Macheswaran, S. Ali, H. J. Siegel, D. Hensgen, R.F. Freund, "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems," *Journal of Parallel Distributed Computing*, Vol. 59, No. 2, 1999, pp. 107–131.

[6] M. Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network,". proc. 1st IEEE Int. Conf. on Peer-to-Peer Computing (P2P2001), Linkoping Sweden, August 2007.

[7] R. Buyya and M. Murshed, GridSim: "A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing", Technical Report, Monash University, Nov. 2001. To appear in the Journal of Concurrency and Computation: Practice and Experience (CCPE), 1-32pp, Wiley Press, May 2007.

[8] Ritchie G and Levine J, "A hybrid ant algorithm for scheduling independent jobs in heterogeneous computing environments", Proc of the 23rd workshop of the UK planning and scheduling special interest group, 2009.

[9] T. D. Braun, H. J. Siegel ,and N. Beck, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," *Journal of Parallel and Distributed Computing*, Vol. 61, 2001, pp. 810-837.

[10] Viktors Berstis, "Fundamentals of Grid Computing", IBM Redbook series, November 2009, http://ibm.com/redbooks.