Volume 3, Issue 4, April 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Review of Peer to Peer Grid Load Balancing Model Based on Ant Colony Optimization with Resource Management

Anamika Jain
M.Tech Student , Department of Computer Science
Govt. Engg. College, Ajmer, India.

Ravinder Singh
Asst. Professor, Department of CSE
Govt. Engg. College, Ajmer, India.

Abstract*— 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. 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. There are some of the challenges involved in the mapping problems include resource heterogeneity and resource load balancing. 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: New organization of resource information node based on p2p model has been introduced. It uses new hybrid ant colony technique for transfer of request from a node to another node in information node organization. In load balancing based p2p grid computing ant algorithm can improve with heuristic or hybrid ant colony algorithm. This hybridization of the ant colony algorithm with other algorithms to produce high throughput computing. Further research may be done to create different heuristic based algorithms for grid computing problems.

Keywords*— Peer to Peer Grid, Ant Colony Optimization, Load Balancing, Resource Management .

I. INTRODUCTION

Grid Technology is defined as the technology that enables virtualization of resources, and resource sharing between various organizations. Grid Computing enables sharing, selection, aggregation of geographically distributed resources dynamically at run time depending on their availability, capability and users QoS requirements. The main objective of the grid technology is to maximize the utilization of the organization's computing resources by making them as shareable entities, and provide computing on demand to the users. Grid supports more efficient data and resource management in order to balance the demand of many new applications. Computational grid denotes that the system has enormous computational capacity than single independent system. Resource allocation and task scheduling are fundamental issues in achieving high performance in grid computing systems. Equalizing the load of all available resources is another important issue in the grid. Grid technologies 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.

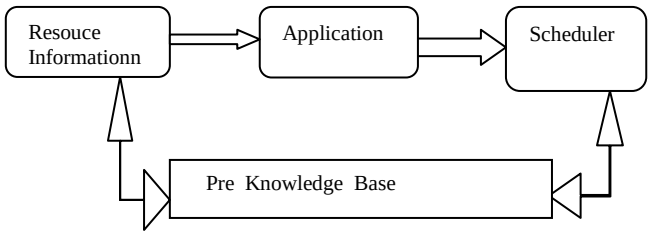
II. LOAD BALANCING

Users are privileged to use Grid for solving large computational problems. It has been strongly demanded to identify the issues affecting the performance. The main advantage of Grid is to share resources among numerous applications. Therefore, the amount of resources available to any given application highly fluctuates over time. In this scenario load balancing plays key role. Load Balancing is one of key concern that can affect the overall performance of application. Load balancing is a technique to distribute workload equally across multiple computers to enhance resources, utilizing parallelism, improve throughput and reduce response time, in Grid environment. In Grid Environment with efficient Load Balancing enhanced system performance and a lower turn-around time for individual jobs can be achieved. To minimize the decision time is one of the objectives for load balancing which has yet not been achieved

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

A. Static Load Balancing Algorithm:

The decisions related to load balance are made at compile time when resource requirements are estimated The advantage of algorithm is the simplicity according to both implementation and overhead, since there is no need to constantly monitor the nodes for performance statistics Static algorithms work properly only when nodes are having low variation in the load. Therefore these algorithms are not well suited for the grid environment, where load is



varying at various times.

Figure 1.1: Static Load Balancing

A few static load balancing techniques are:

(i) Round robin algorithm - the tasks are passed to processes in a sequential order; when the last process has received a task the schedule continues with the first process (a new round).

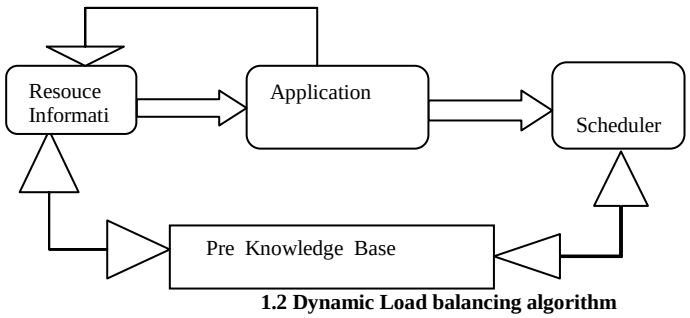
(ii)Randomized algorithm: the allocation of tasks to processes is random.

(iii)Simulated annealing or genetic algorithms: mixture allocation procedure including optimization techniques.

B. Dynamic Load Balancing Algorithm:

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.Dynamic load balancing algorithms are advantageous over static algorithms. But to gain this advantage, we need to consider the cost to collect and maintain the load information.

Dynamic Information



1.2 Dynamic Load balancing algorithm

III. Load Balancing Policies

Load balancing algorithms can be defined by their implementation of the following policies.

(a)**Information policy:** specifies what workload information to be collected, when it is to be collected and from where.

(b)**Triggering policy:** determines the appropriate period to start load balancing operation.

(c)**Resource type policy:** classifies a resource as server or receiver of tasks according to its availability status.

(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 that should be migrated from overloaded resources (source) to most idle resources (receiver).

IV. Peer to Peer Grid:

Peer means a host computer. When many hosts are connected through internet to share files, computing capabilities, networks, bandwidth and storage, it calls peer to peer networking system. Peer to Peer system offers an alternative to such traditional client-server systems for a number of application domains. In P2P system, every node of the system acts as both client and server and provides part of the overall recourses/information available from the system. In pure p2p system no central coordination or central database exists and no peer has a global view of the system. Grids are developed by the infrastructure used to allow flawless access to supercomputer and their datasets. P2P technology could allow using services to set-up and joining peer groups browse and access files on a peer.

4.1 Advantages of Peer to Peer Grid:-

There are two fundamental approaches to achieve this exist:

(i) Unstructured: the data is distributed randomly over the peers and constrained broadcasting mechanism is being used. Example is Gnutella.

(ii) Structured: A distribute, scalable indexing structure is built up to route search requests. Examples are Free Net and P-Grid.

In system following the first approach peer can manage their data completely independently, the approach is fully decentralized, the types of search predict are not limited, and no update dependencies exist. However, these advantages are paid with high search costs in terms of messages or additional delay. The second approach is clearly superior in terms of search efficiency, but the need to establish a distributed index usually requires some form of central coordination or global knowledge. They exhibit the advantages of the structured approach but try to maintain the beneficial characteristics of unstructured networks, such as complete decentralization and support for sub-networks. P-Grid shares some common properties with Chord and CAN but requires much less tight coupling since it does not exploit global knowledge as these systems do and thus is more related to Free Net. Since we believe that only fully decentralize system will be able to fully exploit advantages in terms of scalability and self-organization we focus our interest on second class.

V. Ant Algorithm

Ant colony optimization (ACO) is a Meta heuristic alternative for solving the complicated optimization problems. 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 .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. 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 in any repeat in the problem, such choices will be continued to such extent that all nodes could be surveyed.

VI. CURRENT STATUS OF REASEARCH

Most load balancing approaches are orientated towards application partitioning via graph algorithms. 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 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 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 no 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 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 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.

VII. PROPOSED MODEL FOR LOAD BALANCING BASED ON P2P ORGANIZATION

The model is based on a p2p organization of grid information node.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.

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. This is used for solving load balancing problem in proposed p2p structure of information node ant colony algorithm is used.

Heuristic Algorithms

Scheduler employs many batch mode heuristics algorithms such as OLB, Min-Min, Max-Min and Sufferage heuristics. A new intelligent heuristic search algorithm is developed which guarantees an optimal solution for flow-shop problems with an arbitrary number of jobs and machines provided the job sequence is constrained to be the same on each machine. Opportunistic Load balancing (OLB) algorithm selects the next available machine to allocate for the next task. The procedure's selection will be in an arbitrary manner, so it will produce a non optimal solution. The Min-Min algorithm calculates the minimum completion time of jobs in each machine. Then the overall minimum completion time task is assigned to that particular machine. Max-Min algorithm computes the minimum completion time of each task and then the over all maximum completion time task is assigned to the corresponding machine. The above stated algorithms are static in nature. Grid environment is dynamic in nature. So it is impossible to apply static algorithms in dynamic environment. Due to this reason we are moving to dynamic based algorithm.

Ant Colony Optimization for Grid Load Balancing:

Balanced job assignment based on ant algorithm for computing grids called balanced ant colony optimization (BACO) was proposed by Chang et al. (2007) with the aim to minimize the computation time of job executing in Taiwan UniGrid environment which also focused on load balancing factors of each resource. By considering the resource status and the size of the given job, BACO algorithm chose optimal resources to process the submitted jobs. The local and global

pheromone update techniques were used to balance the system load. Local pheromone update function updated the status of the selected resource after a job had been assigned and the job scheduler depends on the latest information of the selected resource for the next job submission. Global pheromone update function updated the status of each resource for all jobs after the completion of the jobs. By using these two update techniques, the job scheduler obtained the latest information of all resources for the next job submission. From the experimental results, BACO was capable of balancing the entire system load regardless of the size of the jobs in the static scheduling benchmark. The study did not consider the capacity of each resource during the scheduling process. In Moallem & Ludwig (2009), two distributed artificial life-inspired algorithms were introduced, which are ACO and particle swarm optimization (PSO) in solving the static grid load balancing problem. Distributed load balancing are categorized as a robust algorithm that can adapt to any topology changes in a network. In the study, an ant acted as a broker to find the best node in terms of the pheromone value stored in the pheromone table. The node with the lightest load was selected as the best node. The position of each node in the flock could be determined by its load in PSO. The particle compared the load of nodes with its neighbors and moved towards the best neighbor by sending assigned jobs to it. The proposed algorithm performed better than ACO for job scheduling where jobs were submitted from different sources and different time intervals. PSO showed better results than ACO in terms of the makespan. However, PSO used more bandwidth and communication compared to ACO. The main drawback of Ant Colony was that jobs are not scheduled efficiently and therefore load among the resources were not balanced. This problem was fixed by increasing the number of ants that can explore the entire grid system to find resources with the lightest load.

ACO algorithm for dynamic load balancing in distributed systems through the use of multiple ant colonies was proposed by Ali et al. (2010). In this algorithm, information on resources was dynamically updated at each ant movement. Load balancing system was based on multiple ant colonies information. Multiple ant colonies were adopted such that each node sent a colored colony throughout the network. Colored ant colonies were used to prevent ants of the same nest from following the same route and also force them to be distributed all over the nodes in the system. Each ant acted like a mobile agent which carried newly updated load balancing information to the next node. The algorithm was compared to the work-stealing approach for load balancing in grid computing. Experimental results showed that multiple ant colonies worked better than work-stealing algorithm in terms of their efficiency. However, the multiple ant colonies did not consider resources capacity and jobs characteristics. This can make matching the jobs with the best resources a difficult task for the scheduling algorithm. From the above research, ACS is the most popular variant of ACO that has been successfully used in grid load balancing.

VIII. Conclusion and Future Scope

This paper presents the resource management maintains in peer to peer grid but there are some disadvantages according to time ,cost and resource failure which are defines as below:

1. Due to ant algorithm in p2p based load balancing grid there is produced in throughput.
2. In p2p grid ant algorithm for load balancing is another problem of resource failure. In this case the dynamic algorithm will stop and move jobs to the other available resources in a dynamic evaporation rate environment.
3. The p2p grid ant algorithm on multiple ant colonies in solving the grid resource management problem. Involving the multiple ant colonies might improve the performance of the scheduling algorithm as ant populations will be divided into appropriate number of colonies to find the appropriate ways for these colonies to organize their activities.

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 theTraveling 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 DistributedComputing, 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.

[11] Baca, D.F., 1989. "Allocating Modules to Processors in a Distributed System", IEEETransactions on Software Engineering, pp.1427–1436.

[12] Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. "Swarm Intelligence: from natural to artificial systems", Oxford University Press, pp. 75-98.

[13] Bonabeau, E., Theraulaz, G., 2000. Swarm Smarts, Scientific American, pp.72-79.

[14] Braun, T. D., Siegel, H. J., Beck, N., Bölöni, L. L., Maheswaran, M., Reuther, A. I., Robertson,J.P., Theys, M.D., Yao, B., Hensgen, D., and Freund, R.F., 2001. "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(6), pp.810–837.

[15] Chapman, C., Musolesi, M., Emmerich, W., Mascolo, C., 2007. "Predictive ResourceScheduling in Computational Grids", IEEE Parallel and Distributed Processing Symposium,2007 (IPDPS 2007), pp. 1-10

