MIPS issues in Grid Computing

Shilpa Kukreja
Assistant Professor, CSE Department, PIET, Haryana

***Abstract*—Grid computing is very beneficial application which is sharing their resources on different node. There is a large number of user access the data. The basic principal of Grid computing are that data entering by the user is not store locally, but it store in data center on internet. The reliability of the Grid depends on how the loads are being handle. Load balancing means distributing workload across multiple computing resources. Load balancing must take into two major tasks one is the resource providing or resource allocation and second is task scheduling in distributed environment. Many researchers have been proposed various techniques to improve the load balancing. This paper implements the execution time by using the proposed algorithms. Our objective is to develop load balancing algorithm by using Ant Colony Based Algorithms (ACO).**

***Keywords*—ACO, Grid Computing, MIPS rating.**

I. Introduction

The Grid computing means storing and accessing data and programs over the Internet instead of your computer's hard drive. The Grid is just a metaphor for the Internet. . Grid Computing use a technology for the internet and central remote servers to maintain applications and data. Grid computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more inefficient computing by centralizing storage, memory, processing and bandwidth. Grid computing is a model of network computing where a program or application runs on a connected server or servers rather than on a local computing device such as a PC, tablet or smart phone. Example of Grid computing is like Google, Gmail, Yahoo etc.

Figure 1: Grid computing

Grid computing is a combination of –PaaS, IaaS, SaaS . As (IaaS) Infrastructure as a service, it provides hardware and network facility to the end user; thus end user will itself installs or develops its own OS, Software and application. As (SaaS) and Software as a service , Grid offers a pre-made application along

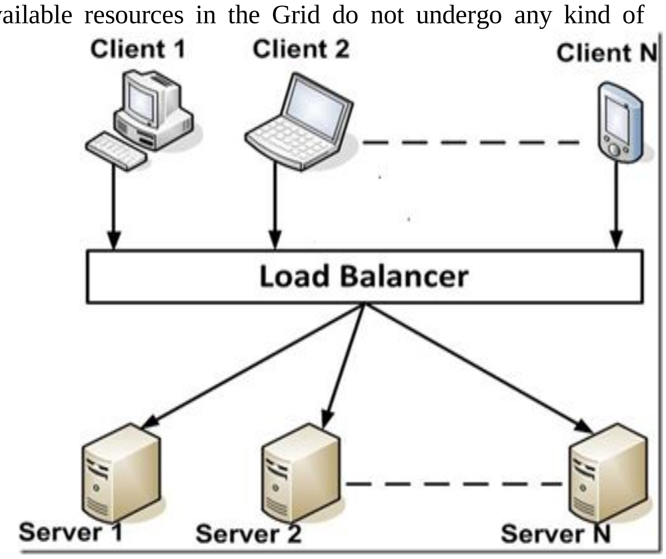
with required software, OS, Hardware and network facility. As (PaaS) Platform as a service , end users are given OS, Hardware, and Network facilities and it is the end user which installs or develops its own software and application. Grid computing is an on demand service in which shared resources, information, software and other devices are provided according to the clients' requirement at specific time. It's a term which is generally used in case of Internet. The whole Internet can be viewed as a Grid.

II. Load Balancing

Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Load balancing is dividing the amount of work that a computer has to do between two or more computer so that more work gets done in the same amount of time, and, in general, all users get served faster.

Figure2. Load balance.

A load balancer is a device that acts as a reverse proxy and distributes network or application traffic across a number of servers. Load balancers are used to increase capacity (concurrent users) and reliability of applications. Grid load balancers manage online traffic by distributing workload between multiple servers and resources automatically. They maximize throughput, minimize response time, and avoid overload. Load balancing perform mainly two tasks one is the resource providing and second is task scheduling in distributed environment. Resource allocation is the task of mapping of the resource to different entities of Grid on demand basis. Resource must be allocated in such manner that no node in the Grid is overloaded and all available resources in the Grid do not undergo any kind of



wastage .Task scheduling define the manner in which different entities are provisioned. Resource provisioning define which resource will be available to meet user requirement whereas task scheduling define the manner in which the allocation resource is available to the end user. Task scheduling can be done into two mode one is Space shared and other is Time shared. In space sharing mode resources are allocated until task does not undergo complete execution. In time sharing mode resource are continuously preempted till task undergoes completion.

III. RELATRD WORK

Static and Dynamic Scheduling

Load balancing can have either Static or Dynamic environment. In Static environment the Grid requires prior knowledge of nodes capacity, processing power, memory, performance and static of user requirements In static environment there is no change at run time execution. Dynamic load balancing techniques is distributed or hierarchical environment provide better performance.

Load balancing algorithm works on the principle on which situation workload is assigned, during compile time or run time. Depending on the compile time or run time it may be static or dynamic. Static algorithms are more stable than dynamic algorithm and it is easy to predict the behavior of static algorithm also. Dynamic algorithms are really works better in case of distributed environments. Novel dynamic load balancing algorithm will be proposed and implemented as a future course of work. It is also necessary to propose a simulation model to evaluate the parameters or components in order to handle the random selection based load distributed problem [1].

IV. HOW CHOOSE THE ALGORITHM FOR DATACENTER

The scheduling algorithms for the datacenter should be chosen on the requirement of datacenter and the kind of data they store in it. We have analyzed the relation between the data that hits the datacenter as well as the scheduling algorithm which is required to promote resource allocation in the Grid datacenters.

A task scheduling algorithm based on load balancing

A Task Scheduling Algorithm Based on Load Balancing: This is discussed in [3] a two-level task scheduling method based on load balancing to convene dynamic requirements of users and obtain high resource utilization. It accomplishes load balancing by first mapping tasks to virtual machines and then virtual machines to host resources by this means improving the task response time, resource consumption and overall performance of the Grid computing environment.

Randomized

Randomized algorithm is of type static in nature [4]. In this algorithm a process can be handled by a particular node n with a probability p. The process allocation order is preserved for each processor independent of allotment from remote processor. This algorithm facilitates well in case of processes that are equal loaded. On the other hand, trouble arises when loads are of different computational complexities. Randomized algorithm does not keep up deterministic approach. It facilitates well while Round Robin algorithm generates overhead for process queue.

Local search algorithms apply on the output of ant algorithms

In paper [5] author proposed that ant algorithm can be improved using some form of local search algorithm. Local search algorithm can be applied to the output of the ant algorithm to find the optimal resource to schedule a job. Author used Move-Top, Move-Minimum Completion Time Job First and Move-Maximum Completion Time Job First local search methods. The job completion time is the only main input for the proposed algorithm. The factors such as CPU workload, communication delay, QoS are not considered. After experimentation it is found that the ant colony algorithm with local search algorithms performs 30% better than the algorithm without local search.

V. PERPOSED ACO PERFORM BETTER THEN OTHER ALGORITHMS

In paper [6] author proposed an ant colony algorithm for dynamic job scheduling in Grid environment. The next resource selection depends on the pheromone value and the transition probability. Author improved the existing ant colony algorithm and tried to minimize the total tardiness time of the job. Author considered that the initial pheromone value depends on current and the expected tardiness time of the job. Two pheromone updating rules are used local update rule and global update rule. Transition probability is used to select the next resource for the job in which the heuristic desirability of the assignment of the job on a machine is inversely proportional to the completion time of the job. In last author compared the performance of various job schedulers and dispatching rules for Grid environment like FCFS, METDD, MTERDW in ACO and it is found that proposed ACO performs 17% better than others. The different cost measures for the algorithm such as makespan time, Grid efficiency and job error ratio, job workflow may be considered for the future work.

VI. COMBINATION OF LOCAL AND TABU SEARCH

In paper [7] author described an ant colony optimization algorithm in combination with local an

Tabu search. The author suggested that in ant colony algorithm the ants build their solutions using both information encoded in the pheromone trail also specific information in the form of the heuristic. The pheromone value updation rule is taken from the Max-Min algorithm in which the pheromone is only updated by the bets ant. And for heuristic information the Min-Min heuristic is used, which suggests that the heuristic value of a particular job should be directly proportional to the minimum completion time of the job. The local search algorithm is applied to each of the solutions built by the ants before the pheromone updation stage to take ant solution to its local optimum. The tabu search algorithm performs number of trails or iterations on the solution build by the ant colony algorithm and after each iteration the solution gets improved. In last the author compared the performance of Min-Min algorithm with the Min-Min+local Search and Min-Min+Tabu Search and ACO and found that the proposed algorithms perform better than the Min-Min.

REFERENCE

[1]. H. El-Rewini, T. G. Lewis, H. H. Ali. Task scheduling in parallel and distributed systems, Prentice-Hall, Inc., Upper Saddle River, NJ, 1994

[2]. D. Gupta, P. Bepari. Load sharing in distributed systems, In Proceedings of the National Workshop on Distributed Computing, January 1999.

[3]. Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp. 1107–1117, 2013.

[4]. L. D. Dhinesh Babu and P. Venkata Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," Applied Soft Computing Journal, vol. 13, no. 5, pp. 2292–2303, 2013.

[5]. J. Cao, K. Li, and I. Stojmenovic, "Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers," IEEE Transactions on Computers, vol. 63, no. 1, pp. 45–58, 2014.

[6]. R. N. Calheiros and R. Buyya, "Meeting deadlines of scientific workflows in public clouds with tasks replication," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 7, pp. 1787–1796, 2014.

[7]. R. Basker, V. Rhymend Uthariaraj, and D. Chitra Devi, "An enhanced scheduling in weighted round robin for the cloud infrastructure services," International Journal of Recent Advance in Engineering & Technology, vol. 2, no. 3, pp. 81–86, 2014.

[8]. Z. Yu, . Menng, and H. Chen, "An efficient list scheduling algorithm of dependent task in grid," in Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT '10), IEEE, Chengdu, China, July 2010.

[9]. H. M. Fard and H. Deldari, "An economic approach for scheduling dependent tasks in grid computing," in Proceedings of the 11th IEEE International Conference on Computational Science and Engineering (CSEWorkshops '08), pp. 71–76, IEEE, San Paulo, Brazil, July 2008.

[10]. W. Kadri, B. Yagoubi, and M. Meddeber, "Efficient dependent tasks assignment algorithm for grid computing environment," in Proceedings of the 2nd International Symposium on Modelling and Implementation of Complex Systems (MISC '12), Constantine, Algeria, May 2012.

[11]. S. Ijaz, E. U. Munir, W. Anwar, and W. Nasir, "Efficient scheduling strategy for task graphs in heterogeneous computing environment," The International Arab Journal of Information Technology, vol. 10, no. 5, 2013.

[12]. Y. Xu, K. Li, L. He, and T. K. Truong, "A DAG scheduling scheme on heterogeneous computing systems using double molecular structure-based chemical reaction optimization," Journal of Parallel and Distributed Computing, vol. 73, no. 9, pp. 1306–1322, 2013.

[13]. L.-T. Lee, C.-W. Chen, H.-Y. Chang, C.-C. Tang, and K.-C. Pan, "A non-critical path earliest-finish algorithm for interdependent tasks in heterogeneous computing environments," in Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications (HPCC '09), pp. 603–608, Seoul, Republic of Korea, June 2009.

[14]. B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," Advances in Engineering Software, vol. 42, no. 7, pp. 419–425, 2011.

[15]. B. Mondal, K. Dasgupta, and P. Dutta, "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach," Procedia Technology, vol. 4, pp. 783–789, 2012.

[16]. M. Rahman, R. Hassan, R. Ranjan, and R. Buyya, "Adaptive workflow scheduling for dynamic grid and cloud computing environment," Concurrency and Computation: Practice and Experience, vol. 25, no. 13, pp. 1816–1842, 2013.

[17]. G. Gharooni-fard, F. Moein-darbari, H. Deldari, and A. Morvaridi, "Scheduling of scientific workflows using a chaosgenetic algorithm," Procedia Computer Science, vol. 1, no. 1, pp. 1445–1454, 2010, International Conference on Computational Science, ICCS 2010.

[18]. C. Lin and S. Lu, "Scheduling scientific workflows elastically for cloud computing," in Proceedings of the IEEE 4th International Conference on Cloud Computing, Washington, DC, USA, July 2015.

[19]. Vijindra and S. Shenai, "Survey on scheduling issues in cloud computing," Procedia Engineering, vol. 38, pp. 2881–2888, 2016, Proceedings of the International Conference on Modelling Optimization and Computing.

[20]. M. Xu, L. Cui, H. Wang, and Y. Bi, "A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing," in Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA '09), pp. 629–634, IEEE, Chengdu, China, August 2015.

[21]. C. Lin, S. Lu, X. Fei et al., "A reference architecture for scientific workflow management systems and the VIEW SOA solution," IEEE Transactions on Services Computing, vol. 2, no. 1, pp. 79– 92, 2009.

[22]. S. Ghanbari and M. Othman, "A priority based job scheduling algorithm in cloud computing," in Proceedings of the International Conference on Advances Science and Contemporary Engineering, pp. 778–785, October 2012.