Load Balancing Through Node Allocation

Palash Samanta[1], Ranjan Kumar Mondal[2]

[1]Department of Engineering and Technology, University of Kalyani, Kalyani, India
Email: samanta.palsh30@gmail.com

[2]Department of Computer Science & Engineering, University of Kalyani, Kalyani, India
Email: ranjan@klyuniv.ac.in

---***---

Abstract - *Now a days, various types of web providers are growing rapidly. Cloud computing has become popular day by day to provide various type of web services and web resources to cloud users. Cloud computing employs web resources to execute multiple large-scale tasks. So, to select a proper node to execute a task is able to enhance the performance of large-scale cloud computing environment. There are different cloud nodes in a cloud computing system. Namely, different node has different capability to execute task; hence, consider the CPU remaining of the node or network bandwidth are not enough when a node is selected to execute a job. So, it is very important issue in a cloud computing how to select a proper node to execute a task. In this paper, we propose a new scheduling algorithm named **Load Balancing through Node Allocation, LBNA** which combines minimum completion time and load balancing strategies. For the case study, LBNA can provide proper utilization of cloud computing resources and maintain the load balancing in cloud computing environment.*

Index Terms: Cloud Computing, Load Balancing, Distributed System, Scheduling.

1. INTRODUCTION

Recently, cloud computing as a new internet service concept has become popular to provide various services to user such as multi-media sharing, on-line office software, game and on-line storage.

Cloud computing is the use of the pooled web resources accessible over Internet. Web resources can be hardware or software. Cloud derives its name from the cloud shaped symbol that represent whole web structure, as it is utilized as an abstraction for its complex infrastructure. It provides web services as per requirement. It allows user to cloud services. It offers services as per payment. Cloud provides resources over Internet.

Virtualization provides abstraction of independent hardware access to each virtual machine that is called VM also. Multi-tenancy allows the same software platform to be shared by multiple applications. Multi-tenancy is important for developing software as a service user application. Applications communicate over the web using web services [1].

In this paper, we propose an efficient load balance algorithm, named LBNA. From the case study, LBNA achieves better load balancing and minimum completion time for completing all tasks than other algorithms such as LB3M, MM and LBMM.

2. LITERATURE REVIEW

Cloud computing is a type of distributed computing where massively scalable resources are provided to multiple customers "as a service" using web technologies. The web-providers have to achieve a great large, general-purpose computing architecture, infrastructure for various customers and web services to provide the multiple web application services.

According to the information of system in a cloud computing environment, the performance of the system will be managed and enhanced. There are several methods can collect the relevant information of node that includes broadcasting, the centralized polling and agent. Agent is one of the technologies used extensively in recent years. It has inherent navigational autonomy and can ask to be sent to some other nodes. In other words, agent should not have to be installed on every node the agent visits, it could collect related information of each node participating in cloud computing environment, such as CPU utilization, remaining CPU capability, remaining memory, transmission rate, etc. Therefore, when agent is dispatched, it does not need any control or connection, and travel flow can be reducing in maintaining the system. However, in this study, the agent is used to gather the related information, and reduce the resources wasting and cost.

There are different characteristics of each scheduling algorithm.

Opportunistic Load Balancing (OLB) is to attempt each node keep busy, therefore it does not consider the current workload of each node. OLB assigns every job in free order to present node of useful .The advantage is very simple and reach load balance but its shortcoming is not consider each expectation execution time of task, therefore the Make span is very low. In other words, OLB dispatches unexecuted tasks to presently available resources at random order, regardless of the node's present workload.

Minimum Execution Time (MET) assigns each job in arbitrary order to the nodes on which it is expected to be executed fastest, regardless of the current load on that node. MET tries to find well job-node pairings, but because it does not consider the current load on a node it will often cause load imbalance between the nodes and not adapt application in the heterogeneity computer system.

Minimum Completion Time (MCT) assigns each task in random order to the nodes with the minimum expected completion time for the task. The completion time is simply the ETC, but this is a more successful heuristic as both execution times and node loads are considered.

Min-Min scheduling algorithm establishes the MCT for every unscheduled task and then assigns the task with the minimum completion time to that node that offers it this time. Min-min uses the same mechanism as MCT. However, because it considers the minimum completion time for all jobs at each round, it can schedule the job that will increase the overall make span the least. Therefore, it helps to balance the nodes better than MCT.
In addition, spirit of Min-min is that every composed of the best is all minimum completion time for allocation resource. Because of OLB scheduling algorithm is very simply and easy to implement and each computer often keep busy.

3. THE PROPOSED METHOD

There are different types of nodes in a cloud computing system. Namely, each node or processor has different capability to execute task; hence, only consider the CPU remaining of the node is not sufficient when a node is chosen to execute a task. Therefore, how to select a particular node to execute a task is very important issue in a cloud computing area.
Due to the tasks that have different types of characteristic for user to pay execution. Hence it is need resources of specific, for instance, when implement organism sequence assembly, it is probable that have to big requirement toward memory remaining and CPU remaining. And in order to reach the best efficient tasks, so we will targeted by tasks property to adopt a different condition decision variable where it is according to resources of tasks requirement to set decision variables.

Problem Definition:

Due to the problem of the resource allocation with minimum completion time of a task with reasonably low cost. In this paper we are going to discuss a proposed algorithm that has been developed under some assumptions.

4. METHOD

The progress of Load Balancing through Node Allocation is presented as following:
Step 1: First we have to calculate the average completion time of each node for all tasks, respectively.
Step 2: Then find the node that has the maximum average completion time.
Step 3: It is to find the unassigned node that has the minimum completion time less than the maximum average completion time for the node selected in step 2. Then this task is dispatched to the selected node for computation.
Step 4: If there are no unassigned node can be selected in step 2, all nodes including unassigned and assigned node should be reevaluated. The minimum completion time of an assigned node is the sum of minimum completion time of assigned task on this node and the minimum completion time of current task. The minimum completion time of an unassigned node is the current minimum completion time for the task. It is to find the node that has the minimum completion time less for the task selected in step 2.
Step 5: Repeat step 2 to 4 until all tasks have completed totally.

5. CASE STUDY

Table 1 shows the completion time for each task at different computing nodes. The threshold is the average completion time of node r_i in all tasks. To evaluate the performance of Load Balancing through Node Allocation is compared with MM and LBMM by the case shown in Table 1. The completion times for completing all tasks by using LB3M, LBMM, MM are 24, 33, 35 seconds respectively.
Load Balancing through Node Allocation also gives the maximum completion time is 24 seconds but it may give some where better result on some cases as the minimum completion time for all tasks are in one node then it will give the better result. As shown on table 2. In this table the completion times for completing all tasks by using LB3M, LBMM, MM are 52, 33, 52 seconds respectively. But in Load Balancing through Node Allocation give the better result 24 seconds.

Table 1. Completion time (in second) of each node at different tasks.

Task \ Node	C11	C12	C13	C14
t1	12	13	10	***14***
t2	***16***	24	13	25
t3	26	31	***12***	33
t4	17	***24***	18	31
Threshold	17.75	23	13.25	25.75

Table 2. Completion time (in second) of each node at different tasks.

Task \ Node	C11	C12	C13	C14
t1	12	13	10	14
t2	16	24	13	25
t3	26	31	12	33
t4	18	24	17	31
Threshold	18	23	13	25.75

6. COMPARISON:

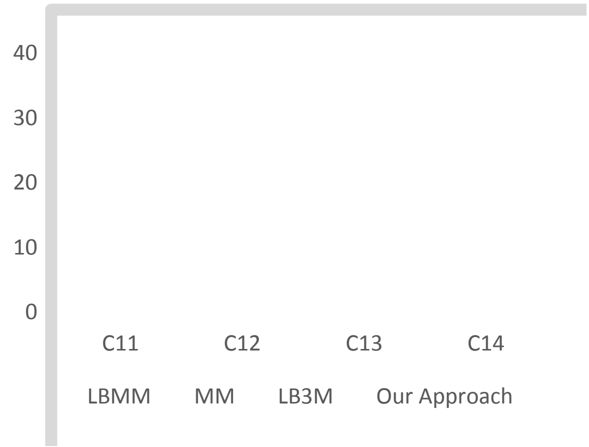

Figure 1.The comparison of completion time of each task at different node.

7. CONCLUSION

In this paper, we proposed an efficient scheduling algorithm, LBNA, for the cloud computing network to assign some tasks to computing nodes according to their resource capability. However, the load balancing of cloud computing network is utilized, all calculating result could be integrated first by the second level node before sending back to the management. Thus, the goal of loading balance and better resources manipulation could be achieved.

ACKNOWLEDGEMENT

We would like to express our gratitude to Debabrata Sarddar, Assistant Professor, Department of Computer Science and Engineering of University of Kalyani. Without his assistance and guidance, we would not have been able to make use of the university's infrastructure and laboratory facilities for conducting our research.

References

[1]. Gautam Shroff, "Enterprise Cloud Computing," Cambridge University Press, June, 2011.

[2]. Wang, S. C., Yan, K. Q., Liao, W. P., & Wang, S. S. Towards a load balancing in a three-level cloud computing network. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on* (Vol. 1, pp. 108-113). IEEE.

[3]. Garces-Erice, L., Biersack, E.W., Ross, K.W., Felber, P.A. and Urvoy-Keller, G., 2003. Hierarchical peer-to-peer systems. *Parallel Processing Letters*, *13*(04), pp.643-657.

[4]. A. Vouk, "Cloud Computing- Issues, Research and Implementations," Information Technology Interfaces, pp. 31-40, June 2008.

[5]. Yan, K.Q., Wang, S.C., Chang, C.P. and Lin, J.S., 2007. A hybrid load balancing policy underlying grid computing environment. *Computer Standards & Interfaces*, *29*(2), pp.161-173.

[6]. K.Q., Yan, et al., "A Hybrid Load Balancing Policy Underlying Grid Computing Environment. Computer Standards & Interfaces, Vol. 29, Issue 2, pp 161-173, 2007.

[7]. R. K. Mondal, P. Ray, D. Sarddar "Load Balancing", *International Journal of Research in Computer Applications & Information Technology*, Volume 4, Issue 1, January-February, 2016, pp. 01- 21, ISSN Online: 2347- 5099, Print: 2348- 0009, DOA : 03012016.

[8]. R. K. Mondal, P. Ray, D. Sarddar "Load Balancing with Task Division and Addition". *International Journal of Scientific Research Engineering & Technology* (IJSRET), ISSN 2278 – 0882, Volume 5, Issue 1, January 2016: 15-19.

[9]. R. K. Mondal, P. Ray, E. Nandi, and D. Sarddar. "Load Balancing with Minimum Task." *IJournals: International Journal of Software & Hardware Research in Engineering*, ISSN-2347-4890, Volume 4 Issue 1 January, 2016: 37-41.

[10]. R. K. Mondal, E. Nandi, P. Ray, D. Sarddar, "Load balancing with Modify Approach", *International Journal of Computer Techniques*. ISSN: 2394-2231, Volume 3 Issue 1, Jan- Feb 2015, pp. 1-5.

[11]. R. K. Mondal, P. Ray, E. Nandi, D. Sarddar, "Load Balancing with Tasks Subtraction" International Journal of Science, *Engineering and Technology Research (IJSETR)*, ISSN:2278 –7798, Volume 5, Issue 1, January 2016, pp-336-344.

BIOGRAPHIES

Palash Samanta is currently pursuing B.Tech in Information Technology from Department of Engineering and Technology in University of Kalyani, Kalyani, Nadia, West Bengal, India. His research interests include Cloud Computing, Wireless and Mobile Communication Systems.

Ranjan Kumar Mondal received his M.Tech in Computer Science and Engineering from University of Kalyani, Kalyani, Nadia; and B.Tech in Computer Science and Engineering from Government College of Engineering and Textile Technology, Berhampore, Murshidabad, West Bengal under West Bengal University of Technology, West Bengal, India. At present, he is a Ph.D research scholar in Computer Science and Engineering from University of Kalyani. His research interests include Cloud Computing, Wireless and Mobile Communication Systems.