Load Balancing Algorithm with Total Task of Different Nodes

Ranjan Kumar Mondal[1]
Department of CSE,
University of Kalyani, Kalyani, India
ranjan@klyuniv.ac.in

Payel Ray[2]
Department of CSE,
University of Kalyani, Kalyani, India
payelray009@gmail.com

Enakshmi Nandi[3]
Department of CSE,
University of Kalyani, Kalyani, India
pamelaroychowdhurikalyani@gmail.com

Debabrata Sarddar[4]
Assistant Professor, Department of CSE,
University of Kalyani, Kalyani, India
dsarddar1@gmail.com

Abstract.
Cloud computing as an Internet computing has become popular day by day to provide various type of services and resources to web user. Cloud computing employs Internet resources to execute large-scale tasks. There are several heterogeneous nodes in a cloud computing system. Namely, each node has different capability to execute task; hence, only consider the CPU remaining of the node is not enough when a node is chosen to execute a task. Therefore, how to select an efficient node to execute a task is very important in a cloud computing.
In this paper, we propose a scheduling algorithm, Load Balancing Algorithm with Total Task of Different Nodes (LBATTDN), which combines minimum completion time and load balancing strategies. For the case study, LBATTDN can provide efficient utilization of computing resources and maintain the load balancing in cloud computing environment.

Keywords: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 e-technology, where virtual resources are provided as services over the Internet. Users need not have gain the technology knowledge or control over the cloud that support the resources. Cloud Computing is high utility software having the ability to change the software industry and making the software even more attractive. Cloud computing has strong impact on software industry and their business. The elasticity of resources is unprecedented in the history of information technology. The increase in web traffic and different services are increasing day by day making load balancing a big research topic. Cloud computing is anew technology which uses implicit machine instead of physical machine to host, store and network the different components.

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

2. Load Balancing in Cloud Computing

Load balancing is the technique of distributing the load between various resources in any system. Thus load require to be distributed over the resources in cloud-based architecture, so that each resources does almost the equal amount of work at any point of time. Basic requirement is to provide some techniques to balance requests to provide the solution of fast response for request. Cloud Load Balancers manage online traffic by distributing workloads between multiple servers and resources automatically. They maximize throughput, minimize response time, and avoid overload.

In this paper, an overall review of the latest load balancing technique in the Cloud Computing environment is submitted. The ideas of every algorithm are discussed and finally sum up as an overview. There are several issues while dealing with load balancing in a cloud computing environment. Everyloadbalancingalgorithm must be such as to instate the required target. Some algorithms target at achieve higher throughput, some target at achieve minimum response time, some other target to achieve maximum resource utilization while some target at achieve a trade-off between all these metrics. Figure1 represent a framework underneath which various load balancing algorithms work in a cloud computing environment.

3. ISSUE EFFECTING LOAD BALANCING

There are many types of load balancing techniques which are available for cloud computing. These load balancing techniques are: geographical distribution, static and dynamic.

3.1 The geographical distribution

The geographical distribution of the nodes matters a lot in the collective performance of any real time cloud computing systems, specifically in case of the big scaled applications like Twitter, Facebook etc. A well-distributed system of nodes in cloud environment is useful in handling fault tolerance and maintaining the efficiency of the system. Geographical load balancing (GLB) can be defined as a series of decisions about online assignment and/or migration of virtual machines (VMs) or computational tasks to geographically distributed datacenters in order to meet the service level agreements (SLAs) or service deadlines for VMs/tasks and to decrease the operational cost of the cloud system.

3.2Static Load Balancing Algorithm

In static load balancing algorithms, the execution of the processors is determined at the beginning of the execution, it does not depend on current state of the system. The aim of static load balancing is to decrease the overall execution time of a synchronous program while minimizing the communication delays. These algorithms are mostly suitable for homogeneous and stable environments and can produce better results .Some of the examples of static load balancing algorithms are: Randomized algorithm, Round Robin algorithm and Threshold algorithm.

3.3 Dynamic Load Balancing Algorithm

In dynamic load balancing algorithm, the decisions in load balancing are based on the current state of the system, no prior knowledge is needed. The major advantage of dynamic load balancing is that if someone node fails, it will not stop the system, it will only affect the performance of the system. These algorithms are more resilient than static algorithms, can easily adapt to alteration and provide better results in heterogeneous and dynamic environments. Dynamic load balancer uses morality for keeping the track of updated information. There are four policies for dynamic load balancers: selection policy, transfer policy, location policy and information policy. The task of load balancing is shared among distributed nodes. In a distributed system, dynamic load balancing can be done in two different ways: distributed and non-distributed.

3.3.1 Distributed Dynamic

Load Balancing Algorithm In the distributed one, the dynamic load balancing algorithm is executed by all nodes present in the system and the task of scheduling is shared among them. The interaction among the nodes to achieve load balancing can take two forms: cooperative and non -cooperative.

3.3.2 Non-Distributed Load Balancing Algorithm

In the non-distributed or undistributed, the nodes work personal in order to instate a common goal. Non distributed dynamic load balancing algorithms are ahead classified into two: centralized and semi-centralized.

3.3.2.1 Semi-distributed Dynamic Load Balancing

In semi-distributed dynamic load balancing, the nodes of the system are divisions into clusters, where the load balancing in each cluster is of centralized form. A central node is elected in every cluster by appropriate election technique which takes care of load balancing within that cluster. Therefore, the load balancing of all system is done via the central nodes of each cluster.

3.3.2.2 Centralized Dynamic Load Balancing

In centralized dynamic load balancing, the algorithm is only executed by a single node in the whole system i.e. central node. This node is perfectly responsible for load balancing of the whole system and rest of the nodes interacts only with the central node.

The remaining part of this paper are organized as section 6 describes related work about various load balancing technique used in recently in soft computing , artificial antigens(AI) and other related subjects we conclude our paper and also provide direction for future enhancement.

4. The Proposed Method

There are several heterogeneous nodes in a cloud computing system. Namely, each node has different capability to execute task; hence, only consider the CPU remaining of the node is not enough when a node is chosen to execute a task. Therefore, how to select an efficient node to execute a task is very important in a cloud computing.

Due to the task that has different characteristic for user to pay execution. Hence it is need some of the resources of specific, for instance, when implement organism sequence assembly, it is probable have to big requirement toward memory remaining. And in order to reach the best efficient in the execution each tasks, so we will aimed by tasks property to adopt a different condition decision variable in which it is according to resource of task requirement to set decision variable.

4.1.Service manager

The cloud computing environment is composed of heterogeneous nodes, where the property of each node may greatly differ. In other words, the computing capability provided by the CPU, the available size of memory, and transmission rate are different. In addition, cloud computing utilizes the resources of each node, so the available resource of each node may vary in a busy condition. From the perspective of task completion time, the available CPU capacity, the available size of memory, and transmission rate are the three decisive factors for the duration of execution.

To make the manager select appropriate nodes effectively, all of the nodes (includes service manager and service node) in the system will be evaluated by the threshold that is derived from the demand for resource needed to execute the task. The service manager that passes the "threshold of service manager" considered effective, and will be the candidate of effective nodes by manager. The service nodes that pass the "threshold of service node" considered effective, and will be the candidate of effective nodes by service manager.

4.2. Method

Step 1: It is to calculate the Total Task of each node for all tasks, respectively.

Step 2: It is to find the Task that has the maximum Total Task value.

Step 3: It is to find the unassigned node that has the minimum completion time less than the Total Task value for the task selected in Step 2. Then, this task is dispatched to the selected node for computation.

Step 4: If there is no unassigned node can be selected in Step 2, all nodes including unassigned and assigned nodes should be reevaluated. It is to find the unassigned node or assigned node that has the minimum completion time less than the maximum Total Task completion time for the task selected in Step 2. Then, this task is dispatched to the selected node for computation.

Step 5: Repeat Step 2 to Step 4, until all tasks have been completed totally.

In the following section, an example to be executed by using the proposed algorithm is given.

5. Case study

Table 1 shows the completion time for each task at different computing nodes.

Task \ Node	C_{11}	C_{12}	C_{13}	C_{14}
t_1	12	13	10	14
t_2	16	24	13	25
t_3	26	31	**12**	33
t_4	17	24	18	31

Table 1:

Step 1: It is to calculate the Total Task of each node for all tasks, respectively.

Task \ Node	C_{11}	C_{12}	C_{13}	C_{14}	Total Task
t_1	12	13	10	14	49
t_2	16	24	13	25	78
t_3	26	31	12	33	102
t_4	17	24	18	31	90

Table 2:

Step 2: It is to find the Task that has the maximum Total Task value.

Task \ Node	C_{11}	C_{12}	C_{13}	C_{14}	Total Task
t_1	12	13	10	14	49
t_2	16	24	13	25	78
t_3	26	31	**12**	33	**102**
t_4	17	24	18	31	90

Table 3:

It is to find the Task that has the next maximum Total Task value.

Task \ Node	C_{11}	C_{12}	C_{14}	Total Task
t_1	12	13	14	39
t_2	16	24	25	65
t_4	**17**	24	31	**72**

Table 4:

It is to find the Task that has the next maximum Total Task value.

Task \ Node	C_{12}	C_{14}	Total Task
t_1	13	14	27
t_2	**24**	25	**49**

Table 5:

It is to find the last task.

Task \ Node	C_{14}	Total Task
t_1	14	14

Table 6:

Final result is following:

Task \ Node	C_{11}	C_{12}	C_{13}	C_{14}
t_1	12	13	10	**14**
t_2	16	**24**	13	25
t_3	26	31	**12**	33
t_4	**17**	24	18	31

Table 7:

6. Comparison:

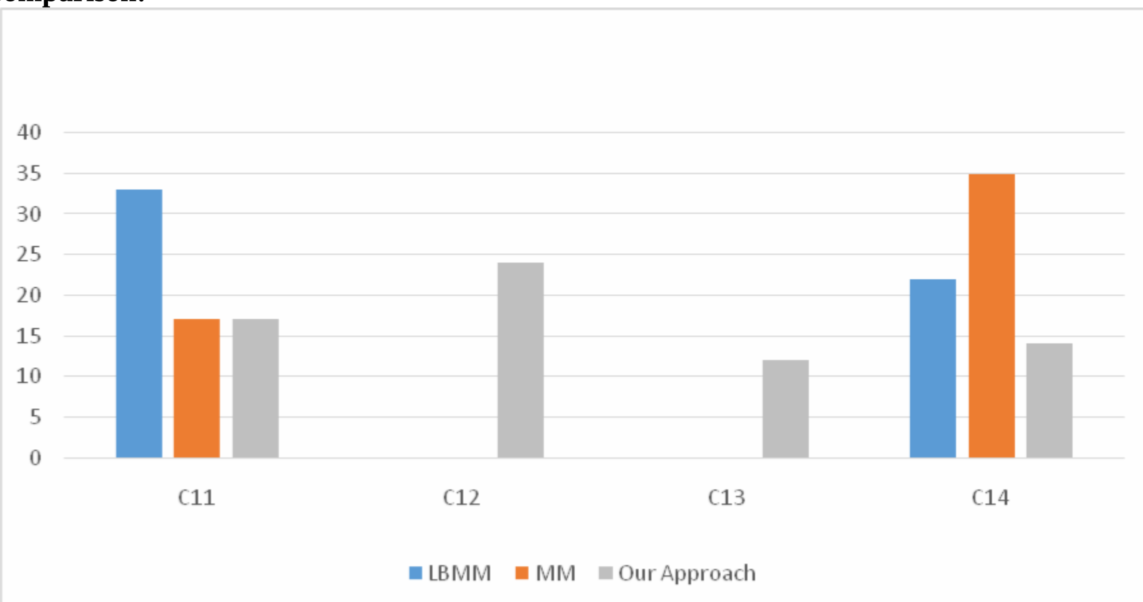

Fig 1.The comparison of completion time of each task at different node for case study.

7. CONCLUSION

In this paper, we proposed an efficient scheduling algorithm, LBATTDN, for the cloud computing network to assign tasks to computing nodes according to their resource capability. Similarly, our approach can achieve better load balancing and performance than other algorithms, such asMM and LBMM from the case study.

REFERENCES

[1]. Hung, Che-Lun, Hsiao-hsi Wang, and Yu-Chen Hu. "Efficient Load Balancing Algorithm for Cloud Computing Network." In International Conference on Information Science and Technology (IST 2012), April, pp. 28-30. 2012.

[2]. Armstrong, R., Hensgen, D., Kidd, T.: The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions. In: 7th IEEE Heterogeneous Computing Workshop, pp. 79—87, (1998)

[3]. Freund, R. F., Siegel, H. J. : Heterogeneous processing. IEEE Computer, vol. 26, pp.13—17, (1993)

[4]. Ritchie, G., Levine, J.: A Fast, Effective Local Search for Scheduling Independent Jobs in Heterogeneous Computing Environments. Journal of Computer Applications, vol. 25, pp. 1190—1192, (2005)

[5]. 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., Freund, R. F.: 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, pp. 810—837, (2001)

[6]. Wang, S. C., Yan, K. Q., Liao, W. P., Wang, S. S.: Towards a Load Balancing in a threelevel cloud computing network. In: Computer Science and Information Technology, pp. 108—113, (2010).

[7]. Ranjan Kumar Mondal, Debabrata Sarddar "Load Balancing with Task Subtraction of Same Nodes". International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online) Vol. 3, Issue 4, pp: (162-166), Month: October - December 2015.

[8]. Ranjan Kumar Mondal, Enakshmi Nandi, and Debabrata Sarddar. "Load Balancing Scheduling with Shortest Load First." *International Journal of Grid and Distributed Computing* 8.4 (2015): 171-178.

[9]. Ranjan Kumar Mondal, Payel Ray, Debabrata 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.

Authors Profile

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.

Payel Ray received her M.Tech in Computer Science and Engineering from Jadavpur University, Jadavpur, India; and B.Tech in Computer Science & Engineering from Murshidabad College of Engineering and Technology, Berhampore, Murshidabad, West Bengal under West Bengal University of Technology, West Bengal, India. At present, she is a Ph.D research scholar in Computer Science and Engineering from University of Kalyani. Her research interests include Cloud Computing, Wireless Adhoc and Sensor Network and Mobile Communication Systems.

Enakshmi Nandi received her M.Tech in VLSI and Micro-electronics from Techno India, Salt Lake, West Bengal and B.Tech in Electronics and Communication Engineering from JIS College of Engineering, West Bengal under West Bengal University of Technology, West Bengal, India. At present, she is Research scholar in Computer Science and Engineering from University of Kalyani. Her research interests include Cloud Computing, Mobile Communication system, Device and Nanotechnology.

Debabrata Sarddar is an Assistant Professor at the Department of Computer Science and Engineering from University of Kalyani,

Kalyani, Nadia, West Bengal, India. He completed his PhD from Jadavpur University. He did his M. Tech in Computer Science & Engineering from DAVV, Indore in 2006, and his B.E in Computer Science & Engineering from NIT, Durgapur in 2001. He has published more than 75 research papers in different journals and conferences. His research interests include Cloud Computing, Wireless and Mobile Communication Systems.