

# Load Balancing Algorithm with Threshold value for Cloud Computing

Ranjan Kumar Mondal<sup>1</sup>, Enakshmi Nandi<sup>2</sup>, Payel Ray<sup>3</sup>, Debabrata Sardder<sup>4</sup>

<sup>1,2,3</sup>Research Scholars: Dept. of Computer Science & Engineering  
University Of Kalyani, Kalyani, Nadia

<sup>4</sup>Assistant Professor: Dept. of Computer Science & Engineering  
University of Kalyani, Kalyani, Nadia

**Abstract**— Cloud Computing is an environment made in client's machine from an online application stored on the cloud and run through a web browser for storing and accessing data and programs over the web in spite of our computer's hard drive. In the cloud storage, load balancing is a key issue. It would consume a lot of cost to maintain load information, since the system is too huge to timely disperse load. Load balancing is one of the main challenges in cloud computing which is required to distribute the dynamic workload across multiple nodes to ensure that no single node is overwhelmed. It helps in optimal utilization of resources and hence in enhancing the performance of the system. A few existing scheduling algorithms can maintain load balancing and provide better strategies through efficient job scheduling and resource allocation techniques as well. In order to gain maximum profits with optimized load balancing algorithms, it is necessary to utilize resources efficiently. Cloud computing employs web resources to execute large-scale tasks. Hence, for selecting proper node for executing a task is able to increase the performance of large-scale cloud computing environment.

**Keywords**— Cloud Computing; Load Balancing; Distributed System; Scheduling.

## I. INTRODUCTION

Recently, cloud computing as a new internet service concept has become popular to provide various services to client such as multi-media sharing, on-line office software, game and on-line storage. In a cloud environment, each host as a computational node performs a task or a subtask. The Opportunistic Load Balancing algorithm (OLB) intends to keep each node busy regardless of the current workload of each node [1, 2, 4]. OLB assigns tasks to available nodes in random order. The Minimum Completion Time algorithm (MCT) assigns a task to the node that has the expected minimum completion time of this task over other nodes [4]. The Min-Min scheduling algorithm (MM) adopts the same scheduling approach as the Minimum Completion Time algorithm (MCT) [4] to assign a task the node that can finish this task with minimum completion time over other nodes [5]. The Load Balance Min-Min (LBMM) scheduling algorithm [6] adopts MM scheduling approach and load balancing strategy. It can avoid the unnecessary duplicated assignment.

Cloud Computing provides the scalable IT resources such as applications and services, as well as the infrastructure on which they operate, over the Internet, on pay-per-use basis to adjust the capacity quickly and easily. It helps to accommodate changes in demand and helps any organization in avoiding the capital costs of software and hardware [2] [3]. Thus, Cloud Computing is a

framework for enabling a suitable, on-demand network access to a shared pool of computing resources (e.g. networks, servers, storage, applications, and services).

These resources can be provisioned and de-provisioned quickly with minimal management effort or service provider interaction. Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

As cloud computing is in its evolving stage, so there are many problems prevalent in cloud computing. Such as:

- Ensuring proper access control (authentication, authorization, and auditing)
- Network level migration, so that it requires minimum cost and time to move a job
- To provide proper security to the data in transit and to the data at rest.
- Data availability issues in cloud
- Legal quagmire and transitive trust issues
- Data lineage, data provenance and inadvertent disclosure of sensitive information is possible

## II. LOAD BALANCING

Load Balancing is a computer networking method to distribute workload across various computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. The load balancing service is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server.

Load balancing is one of the central issues in cloud computing [5]. It is a mechanism that distributes the dynamic local workload evenly across all the nodes in the whole cloud to avoid a situation where some nodes are heavily loaded while others are idle or doing little work. It helps to achieve a high user satisfaction and resource utilization ratio, hence improving the overall performance and resource utility of the system. When one or more components of any service fail, load balancing helps in continuation of the service by implementing fair-over, i.e. in provisioning and de-provisioning of instances of applications without fail.

The main objective of load balancing is improving the performance by balancing the load among these various resources (network links, central processing units, disk drives.) to achieve optimal resource utilization,

maximum throughput, maximum response time, and avoiding overload. To distribute load on different systems, different load balancing algorithms are used.

### III. PROPOSED WORK

There are different 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.

#### 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.

#### A. Methodology

**Step 1:** It is to calculate the Average task (Threshold value) of each node for all tasks, respectively.

**Step 2:** It is to find the node that has the maximum Threshold value.

**Step 3:** It is to find the unassigned node that has the minimum completion time less than the Threshold 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 average 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.

### IV. CASE STUDY

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

Task Node	C <sub>11</sub>	C <sub>12</sub>	C <sub>13</sub>	C <sub>14</sub>
t <sub>1</sub>	12	13	10	14
t <sub>2</sub>	16	24	13	25
t <sub>3</sub>	26	31	12	33
t <sub>4</sub>	17	24	18	31

Table 1

Calculate the Threshold value of each node for all tasks, respectively.

Task Node	C <sub>11</sub>	C <sub>12</sub>	C <sub>13</sub>	C <sub>14</sub>	Threshold
t <sub>1</sub>	12	13	10	14	12.25
t <sub>2</sub>	16	24	13	25	19.5
t <sub>3</sub>	26	31	12	33	<b>25.5</b>
t <sub>4</sub>	17	24	18	31	22.5

Table 2

Select highest threshold value and its minimum task from that row and assign that task its corresponding node.

Task Node	C <sub>11</sub>	C <sub>12</sub>	C <sub>13</sub>	C <sub>14</sub>	Threshold
t <sub>1</sub>	12	13	10	14	12.25
t <sub>2</sub>	16	24	13	25	19.5
t <sub>3</sub>	26	31	<b>12</b>	33	<b>25.5</b>
t <sub>4</sub>	17	24	18	31	22.5

Table 3

Dispatch that selected node and its corresponding row.

Task Node	C <sub>11</sub>	C <sub>12</sub>	C <sub>14</sub>	Threshold
t <sub>1</sub>	12	13	14	13
t <sub>2</sub>	16	24	25	21.67
t <sub>4</sub>	<b>17</b>	24	31	<b>24</b>

Task Node	C <sub>12</sub>	C <sub>14</sub>	Threshold
t <sub>1</sub>	13	14	13.5
t <sub>2</sub>	<b>24</b>	25	<b>24.5</b>

Task Node	C <sub>14</sub>	Threshold
t <sub>1</sub>	<b>14</b>	<b>14</b>

Table 6

Final result is following:

Task Node	C <sub>11</sub>	C <sub>12</sub>	C <sub>13</sub>	C <sub>14</sub>
t <sub>1</sub>	12	13	10	<b>14</b>
t <sub>2</sub>	16	<b>24</b>	13	25
t <sub>3</sub>	26	31	<b>12</b>	33
t <sub>4</sub>	<b>17</b>	24	18	31

## V. RESULT

### Comparison:

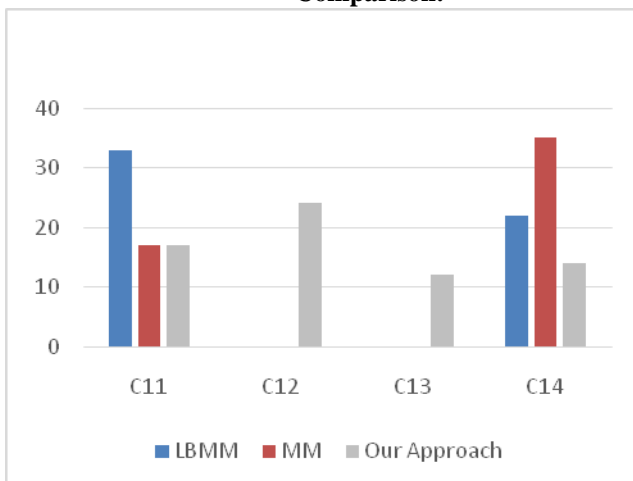


Fig 1. The comparison of completion time of each task at different node for case study.

## VI. CONCLUSION

In this paper, we proposed an efficient scheduling algorithm, LBTV, 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 as, MM and LBMM from the case study.

## ACKNOWLEDGMENT

We would like to express our gratitude to Dr. Kalyani Mali, Head of Department, Computer Science and Engineering of University of Kalyani. Without her 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]. 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., Gherrity, M., Ambrosius, S., Campbell, M., Halderman, M., Hensgen, D., Keith, E., Kidd, T., Kussow, M., Lima, J., Mirabile, F., Moore, L., Rust, B., Siegel, H.: Scheduling resources in multi-user, heterogeneous, computing environments with SmartNet. In: 7th IEEE Heterogeneous Computing Workshop, pp. 184—199, (1998)
- [4]. Freund, R. F., Siegel, H. J. : Heterogeneous processing. IEEE Computer, vol. 26, pp.13—17, (1993)
- [5]. 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)
- [6]. 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)
- [7]. 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).
- [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, 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.

## AUTHORS AND AFFILIATIONS



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.



Enakshmi Nandi received her M.Tech in VLSI and Microelectronics 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.



Payel Ray received her M.Tech in Computer Science and Engineering from Jadavpur University, Jadavpur, India; and B.Tech in Computer Science and 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 and Mobile Communication Systems.



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 System