ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (162-166), Month: October - December 2015, Available at: www.researchpublish.com

Load Balancing with Task Subtraction of Same Nodes

Ranjan Kumar Mondal[1], Debabrata Sarddar[2]

[1]Research Scholar, [2]Assistant Professor
[1,2]Department of Computer Science and Engineering, University of Kalyani, Kalyani, India

Abstract: **Cloud computing helps to share data and provide many resources to users. Users pay only for those resources as much they used. Cloud computing stores the data and distributed resources in the open environment. The amount of data storage increases quickly in open environment. So, load balancing is a main challenge in cloud environment. 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 new scheduling algorithm that choose a suitable node with its average task. It is very easy way to select an appropriate node. This approach can provide efficient utilization of computing resources and maintain the load balancing in cloud computing environment.**

Keywords: **Cloud Computing, Load Balancing, Distributed System, Threshold.**

1. INTRODUCTION

Cloud computing is a on demand service in which shared resources work together to perform a task to get the results in minimum possible time by distribution of any dataset among all the connected processing units. Cloud computing is also referred to refer the network based services which give an illusion of providing a real server hardware but in real it is simulated by the software's running on one or more real machines. Such virtual servers do not exist physically so they can be scaled up and down at any point of time. Cloud computing that we are dealing with is related to the processes to decomposing 'Big Data' tasks into small dataset tasks and distributing them to several computing units, so the task can be performed in the minimum possible time. Three main characteristics are required by any cloud service : resource publication through a single provider, ability to acquire transactional resource on demand, and mechanisms to bill users on the basis of resource utilization. For detailed discussion about deployment of cloud and services provided by the cloud please use the publications.

Cloud Computing Characteristics:

• **On demand service-** Cloud computing provide services to users on their demand .Users can access the services as they want.

• **Broad Network Access-** In cloud computing capabilities are available over the network .All the capabilities are accessed through different mechanisms.

• **Resource Pooling-** Different models are used to pooled the resources which provide by the providers to their consumers. All the resources dynamically assigned and reassigned according to consumer demand.

• **Rapid Elasticity-** Quantity of resources is increase at any time according to the customer's requirements.

• **Measured Service-** In cloud computing resource usage can be monitored, controlled for both the provider and consumer of the all service.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (162-166), Month: October - December 2015, Available at: www.researchpublish.com

Challenges in Cloud Computing:

There are many challenges in cloud computing:-

1. Security
2. Efficient load balancing
3. Performance Monitoring
4. Consistent and Robust Service abstractions
5. Resource Scheduling
6. Scale and QoS management
7. Requires a fast speed Internet connection.

2. LOAD BALANCING

Load balancing in cloud computing is a new challenge now. Always a distributed solution is required because it is not always practically feasible or cost efficient to maintain one or more idle services just as to fulfill the required demands. Jobs can't be assigned to appropriate servers and clients individually for efficient load balancing as cloud is a very complex structure and components are present throughout a wide spread area.

Load balancing algorithms are classified as static and dynamic algorithms. Static algorithms are suitable for homogeneous and stable environments and can produce very good results in these environments. However, they are usually not flexible and cannot match the dynamic changes to the attributes during the execution time. Dynamic algorithms are more flexible and take into consideration different types of attributes in the system both prior to and during run-time[1]. Load balancing is the process of improving the performance of system through a redistribution of load among processor.

Persistence Issue in Load Balancing:

One of the main issues faced by a load balancer is when operating a load balancer service is 'how to handle information that must be kept across the multiple requests in a user's session' e.g. session data. If the information is stored only on computation requesting system, it will not be accessible to other computing devices and since subsequent requests takes place for this information this can lead to a performance issue. One of the solutions to this issue is sending all the requests to the same computing device which contains this information.

This is known as persistence or stickiness. One major downside of this technique is lack of automatic failures. If the device containing the information goes down the whole system gets down, also the any session of the processes present on the system is also lost. So the problem is because of non backed up centralized system, one of the solution is using a backup system together but this will lead to major performance issue. Next solution is related to backup system.

One other solution that can be used is by using database for storing the information, but this increases the load on the database. But databases provide solutions to some of the problems that exists in centralized systems. Databases can be backed up easily solving the problem of single point of failure, databases are also highly scalable. Since in a backed up system of databases there are several systems holding the same information the query load can also be distributed over them to get a better performance. Microsoft's ASP.net State Server technology is an example of this kind of database. All servers in the cloud store their information data on State Server and any server in the cloud can retrieve the data.

3. THE PROPOSED METHOD

There are several heterogeneous nodes in a cloud computing system. All nodes have no capability to execute same 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.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (162-166), Month: October - December 2015, Available at: www.researchpublish.com

Due to the task maybe 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 tasks property to adopt a different condition decision variable in which it is according to resource of task requirement to set decision variable.

Method:

Step 1: It is to subtract the highest task value and lowest task value of each node, respectively.

Step 2: It is to find highest subtraction.

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

Step 4: It is to find next highest subtraction and its minimum task.

Step 5: Repeat 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.

4. CASE STUDY

There is an example of completion time (in sec) of each task at different computing nodes in table 1.

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 subtract the highest task value and lowest task value of each node, respectively.

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
Subtraction	14	18	8	17

Table 2

Step 2: It is to find highest subtraction.

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
Subtraction	14	**18**	8	17

Table 3

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

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
Subtraction	14	**18**	8	17

Table 4

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (162-166), Month: October - December 2015, Available at: www.researchpublish.com

Step 4: It is to find next highest subtraction.

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
Subtraction	14	**18**	8	**17**

Table 5

Repetition

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
Subtraction	**14**	**18**	8	**17**

Table 6

And

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
Subtraction	**14**	**18**	**8**	**17**

Table 7

Final Result:

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 8

5. COMPARISON

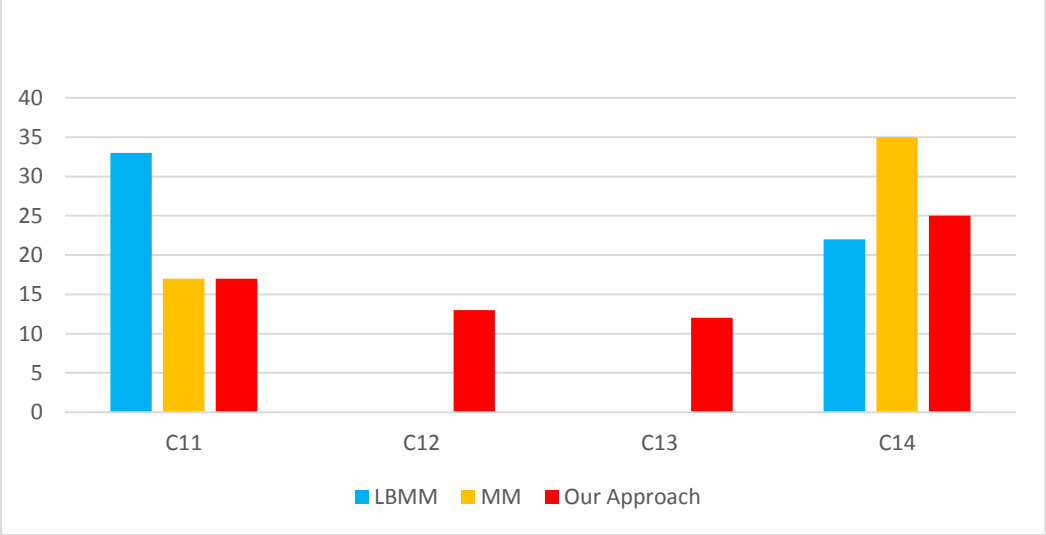

Fig 1.The comparison of completion time of each task at different node for case study

6. CONCLUSION

In this paper, we proposed an efficient scheduling algorithm, LBTSSN, 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 LB3M, MM and LBMM from the case study.

In this paper, we have presented a new scheduling algorithm for scheduling. The goal of the scheduler in this paper is minimizing makespan and maximizes resources utilization.

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.

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.

Debabrata Sarddar is an Assistant Professor at the Department of Computer Science and Engineering, 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 Wireless and Mobile Systems and WSN, and Cloud computing.