International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Load Balancing of Unbalanced Matrix Problem of Maximum Machines with Min Min Algorithm

Ranjan Kumar Mondal, Payel Ray, Enakshmi Nandi, Debabrata Sarddar

Abstract: Most eminent computing technology related to the cloud is an exclusively web-based approach where resources are hosted on a cloud; it prospers the resources. Cloud computing is the most promising technologies that offer a standard for bulky sized computing. That is a structure for enabling applications to execution on virtualized resources and accessed by a network protocol. It provides resource and services in a very elastic behavior that can be scaled according to the required of the clients. Limited numbers of devices execute less number of tasks at that time. So it is more complex to perform each task at once. Several devices execute each task, so it has required balancing total loads that reduce the completion time and executes each task in a definite way.

We have said earlier that there are not feasible to stay behind an equal server to execute similar tasks. The tasks that are to be executed by machine in the cloud system must be less than the united VM for a time. Overloaded servers have to perform a less number of jobs. Here in our approach, we want to show a scheduling algorithm for balancing of loads and presentation with minimum execution time and makespan.

Index Terms: Load Balancing, Task Scheduling, Hungarian method.

I. INTRODUCTION:

A cloud system is an only just progressing practice of providing online computing resources, storage and authorizes customers to arrange applications with scalability, accessibility and fault tolerance [1]. Cloud system is about storing the objects on remote workstations instead of on individual machines or other devices [2]. This information is able to be retrieved using the web on any device, awfully where in the world as long as that device is able to hold up cloud system systems [3]. The cloud system is comprised of a front-end, that is the customer side and a back-end that is a set of the workstations and machines owned by an agent to store the data [4]. An essential workstation that is a fragment of the back-end follows protocols and employs middleware to communicate between networked machines [5]. Cloud system accumulates all the computing resources and manages them mechanically [6]. Its characteristics describe a cloud system: on-need self-service, pooling of resources, right of entry to the web, the elasticity of service accessibility and capacity of services made use of by entity clients. Cloud system is hugely were with tools like Google Drives replacing Microsoft Office, Amazon Web Services replacing conventional enterprise data storage, websites replacing branch offices and Dropbox storing all the data and information.

A. Cloud Computing Characteristics

Cloud computing characteristics are as following:

1. Service gives on-demand - When any client wants services and resources, then the cloud give services on demand.

2. Rapid Elasticity - Amount of supplies in the web| is able to be increased and decrease easily.

3. Resource pooling - Resources are allocated at the various location according to client’s need.

4. Pay per use - According to the buyer ’s consumption of resources is charged.

B. Services of Cloud Computing

Cloud computing gives the following services:

1. SaaS: SaaS give those services in this case the client is able to access software and applications over the web. Example- salesforce.com, Google-docs, Google-mail, Facebook.
2. PaaS: PaaS accommodate all the services and resources to the purchasers that are needed for developing resources. It gives all the services through the web. There is no need to install or download the software at the client end. Example - Google App Engine.
3. IaaS: IaaS gives hardware as a service and services provided as cargo space, OS, arrangement, and virtualization. In IaaS there do not require to buy any software and hardware. These services are accommodating the client's demand. Example - Amazon EC2.

C. Virtualization

Virtual workspaces: An concept of an absolute setting that is able to be completed with passion existing to authorized end-users by utilizing distinct protocols, supply quota (e.g., CPU, memory share), software design (O/S, give services). Realize on Virtual Machines (virtual machines):

Manuscript published on 30 September 2019
* Correspondence Author

Ranjan Kumar Mondal*, CSE, University of Kalyani, WB, India, ranjan@klyuniv.ac.in
Payel Ray, CSE, University of Kalyani, WB, India payelray009@gmail.com,
Enakshmi Nandi, CSE, University of Kalyani, WB, India nandienakshmi@gmail.com,
Dr. Debabrata Sarddar, CSE, University of Kalyani, WB, India, dsarddar12gmail.com

concept of a physical multitude device, Hypervisor emulates information from virtual machines, and permits organization of virtual machines; the virtual machine is, Xen, etc. Give infrastructure API: Plugins to hardware/ support arrangements [7].

Benefits of virtual machines: Perform operating systems in which the hardware is unexisting, more easy to generate the latest equipment, endorsement systems, etc., Software testing by means of 'clean' installs of operating systems and software, follow more machines that are physically existing,Timeshare evenly loaded systems on one host, correct problems, effortless migration of virtual machines.

II. LOAD BALANCING

The balancing process of load is a procedure of engaging t he complete load to the different nodes of the distributed system to make resource use well-organized and to get a better response time of respective tasks, alongside discarding a circumstances in which a little number of nodes are overloaded whereas a number of other nodes are under loaded [8]. Typical data center implementations depend on extensive, powerful computing hardware and network infrastructure; this is an important issue about standard risks which are related to any device, surplus hardware failure, network instability, and resource restrictions in the era of elevated requirement. Load balancing is composed of accessible resources, those are unused or not being utilized properly. To maintain load distribution properly, users are able to transfer the load from respective machines (which have extra workloads) to the reasonably light loaded machines. Our purpose for this article is to build up a useful load balancing approach by Ant Colony Optimization method to make light of various factors similar as a load of CPU, Delay or bandwidth for clouds of multiple figures.

III. RELATED WORKS

Cloud computing presents a selection of services to the client such as different types of media contribution, online office software, game, and web-based storage. In cloud surroundings, all systems perform schedules or a sub-schedules. The Opportunistic Load Balancing method plans to maintain all systems hectic despite the present workloads of all systems [9]. Opportunistic Load Balancing algorithm assigns jobs to presented systems in a random sequence — the least amount Completion Time algorithm in allocating tasks to the system shaving the accepted smallest time of completion for a respective job over other systems [10]. The Min-Min scheduling presumes the similar arrangement move towards as the lowest amount Completion Time [11] algorithm to allocate a task, the organization to terminate this job with lowest completion time over another system [12]. The Load Balance Min-Min method [13] accustoms Min-Min method and load balancing scheme. It can evade the without cause replicated job. Load Balancing with Job Switching [14] decrease loads of huge amount of loaded device to underloaded device by switching a detailed schedule. Load Balancing of Unbalanced Matrix with Hungarian Method [15] utilizes the Hungarian method where schedules are more than each machine. Load balancing process of unbalanced cost matrix [16] is similar to the earlier algorithm.

VI. PROPOSED WORK:

Every time a dilemma is not a square matrix, so the quantity of sources machines of a system is asymmetrical to the quantity of destinations technology of that scheme, the predicament is known as unbalanced assignment matrix.

To give an algorithmic illustration of the method, let us a problem to consist of a set of' systems i={N_1, N_2, ..., N_i}. And 'j' jobs j={J_1, J_2,..., J_j} is considered to be distributed for execution on 'm' accessible systems and the execution value C_{mn}, where m=1,2,3..., i and n=1,2,3...,i, where j is greater than i, i.e., the no. of jobs is always larger than a no. of machines.

Algorithm:

Step-1: *Input: an m*n* problem where m<n.

Step-2: *Find Minimum execution cost from each Job and allocate that job to the corresponding Machine and remove that job from List till all Jobs are allocated to corresponding machines.*

Step-3: *While all minimum Jobs are allocated to their corresponding Machine then Stop This.*

Step-4: *Some machine would not be assigned by job simply ignore it.*

Example: *Suppose an industry has six machines employed for four jobs. Each job could be allocated to one and only one machine at the same time.*

The input of every job on every machine is prearranged in the following Table.

Jobs/Machines	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
J_{11}	10	19	8	15	21	18
J_{12}	10	18	17	17	42	23
J_{13}	13	26	19	14	19	22
J_{14}	22	19	18	18	38	22
J_{15}	14	17	10	19	24	11

This corresponds to the following most constructive result in the new cost matrix:

$$\begin{array}{cccccc} 10 & 19 & 8 & 15 & 21 & 18 \\ 10 & 18 & 17 & 17 & 42 & 23 \\ 13 & 26 & 19 & 14 & 19 & 22 \\ 22 & 19 & 18 & 18 & 38 & 22 \\ 14 & 17 & 10 & 19 & 24 & 11 \end{array}$$

V. EXPERIMENT

Table 1 displays the execution time for the entire jobs in different computing systems. The execution time of task shows in systems. For calculating the performance of the proposed algorithm, our approach is judged against other existing MM algorithm and LBMM algorithm by the case exposed in Table 1. Figure 1 presents the evaluation execution time of all executing systems amongst approach, MM, and LBMM. The execution times for completing each task by using the proposed, LBMM algorithm and MM algorithm are 17, 27 and 42 ms, respectively. Our work achieved the least execution time and improved load balancing than further algorithms, for example, LBMM and MM in this case.

VI. COMPARISON

Our proposed work of load balancing of the unbalanced cost matrix can get better to balance for load and presentation better than other existing algorithms, such as LBMM algorithm and MM algorithm from the given figure.

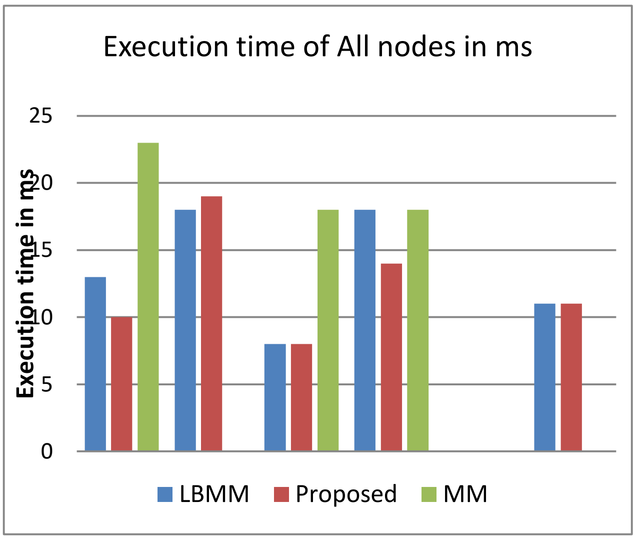

Figure 1: The estimation of execution time (ms) of all jobs at a system.

VII. CONCLUSION

In this paper, our method demonstrates a resourceful algorithm in the ground of cloud computing networks to allocate all schedules to computer systems with their respective resource facilities. Correspondingly, this algorithm can reach improved load balancing and act than existing some algorithms, such as MM and LBMM from the theoretical study.

REFERENCES

1. Chan, M. C., Jiang, J. R., & Huang, S. T. “Fault-tolerant and secure networked storage”. In Digital Information Management, 2012 Seventh International Conference on (pp. 186-191). IEEE.
2. Jeevitha, M., Chandrasekar, A., & Karthik, S. “Survey On Verification Of Storage Correctness In Cloud Computing”. International Journal Of Engineering And Computer Science, 4(09), 2015.
3. Chen, C. H., Lin, H. F., Chang, H. C., Ho, P. H., & Lo, C. C. “An analytical framework of a deployment strategy for cloud computing services”: a case study of academic websites. Mathematical Problems in Engineering, 2013.
4. Dinh, H. T., Lee, C., Niyato, D., & Wang, P. “A survey of mobile cloud computing: architecture, applications, and approaches”. Wireless communications and mobile computing, 13(18), 1587-1611, 2013.
5. Pareek, P. “Cloud Computing Security From Single to MultiClouds using Secret Sharing Algorithm”. International Journal of Advanced Research in Computer Engineering & Technology, 2(12), 3261-3264, 2013.
6. Cavilla, H. A. L. “Flexible Computing with Virtual Machine”s (Doctoral dissertation, University of Toronto), (2009).
7. Moura, J., & Hutchison, D. “Review and analysis of networking challenges in cloud computing”. Journal of Network and Computer Applications, (2016), 60, 113-129.
8. Braun T. D., et.al. “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)
9. Wang S. C., et.al. “Towards a Load Balancing in a three-level cloud computing network”. In: CSIT, pp. 108—113, (2010).
10. Hung, Che-Lun, et. al. "Efficient load balancing algorithm for a cloud computing network." In International Conference on Information Science and Technology (IST 2012), April, pp. 28-30. 2012.
11. Kokilavani, T., et.al.. "Load balanced min-min algorithm for static meta-task scheduling in grid computing." International Journal of Computer Applications 20, no. 2 (2011): 43-49.
12. Wu, Min-You, Wei Shu, and Hong Zhang. "Segmented min-min: A static mapping algorithm for meta-tasks on heterogeneous computing systems." In Heterogeneous Computing Workshop, 2000. Proceedings. 9th, pp. 375-385.
13. R K Mondal, S Roy, P Samanta, P Ray, E Nandi, B Biswas, M K Sanyal, and D Sarddar. "Load Balancing with Job Switching in Cloud Computing Network." In Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications, pp. 305-312. Springer, Singapore, 2017.
14. Kuhn, Harold W. "The Hungarian method for the assignment problem." Naval Research Logistics (NRL) 2, no. 1-2 (1955): 83-97.
15. R K Mondal, P Ray, E Nandi, B Biswas, M K Sanyal, and D Sarddar. "Load Balancing of Unbalanced Matrix with Hungarian Method." In International Conference on Computational Intelligence, Communications, and Business Analytics, pp. 256-270. 2017.
16. R K Mondal, P Ray, E Nandi, P Sen, and D Sarddar. "Load balancing of the unbalanced cost matrix in a cloud computing network." In Computer, Communication and Electrical Technology: Proceedings of the International Conference on Advancement of Computer Communication and Electrical Technology (ACCET 2016), p. 81. CRC Press, 2017.

AUTHORS PROFILE

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

Payel Ray received her M.Tech in CSE from Jadavpur University, Jadavpur, and B.Tech in CSE from MCET, Berhampore, Murshidabad, WB under WBUT, WB, India. At present, she is a Ph.D. research scholar in CSE from the 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, WB and B.Tech in ECE from JIS College of Engineering, WB under WBUT, West Bengal, India. At present, she is a Ph.D. Research scholar in CSE from University of Kalyani. Her research interests include Cloud Computing, Mobile Communication system, Device, and nanotechnology.

Dr. Debabrata Sarddar is an Assistant Professor at CSE from University of Kalyani, Kalyani, Nadia, WB, India. He completed his Ph.D. from JU. He did his M. Tech in CSE from DAVV, Indore in 2006, and his B.E in CSE from NIT, Durgapur in 2001. He has published more than 100 research papers in different journals and conferences. His research interests include Cloud Computing, Wireless, and Mobile Communication System.