Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 2223 - 2229
Received 16 February 2021; Accepted 08 March 2021.

Certain Investigation on Load Balancing using Cloudlet Assignment and min-max Algorithm

*** Mr.S.Ranjithkumar[1], Mr.Kandasamy Sellamuthu[2], Mr.R.S.Rajkumar[3], Mr.V.Krishnakumar [4]**

[1]Department of Computer Science and Engineering, Sri Ramakrishna Engineeirng College, Coimbatore, TamilNadu, India.
[2], Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, Tamilnadu, India.
[3]Department of Computer Science and Engineering, Sri Ramakrishna Engineeirng College, Coimbatore, Tamilnadu, India.
[4]Department of Computer Science and Engineering, Sri Ramakrishna Engineeirng College, Coimbatore, Tamilnadu, India.

Abstract: Cloud Computing is a trending technology where users can access the resources that run on servers.Cloud usage has major issues like scheduling and load balancing. As cloud users increase, there is an increase in computation costs, decrease in performance, and network traffic. This issue is quite common in most of the servers. Load balancing is the process of balancing overloads or underloads of the required server. It is required to design a load balancer that has the least CPU utilization time for the allocation and utilization of resources. This paper solves the issue of load balancing using machine learning algorithms thereby the means to optimize it. The future system reduces the waiting period and the energy depletion of the data centres in the cloud by 43.57%.

Keywords- Cloud Computing, Evolutionary Algorithm, Parallel Genetic Algorithm, Fitness Function.

1. INTRODUCTION

Cloud Computing is used to provide a service over the internet. The new trend of cloud computing is linked to the word 'internet' as the services are provided over the internet. Buying the computing services than doing it ourselves reduces cost, time and increases resource reusability. The users can access data and applications over a network through cloud computing. It is the use of demanded services on a payable basis. It is a virtual platform with elasticated resources. It reduces the burden by eliminating the need to install and run applications on a customer's application. Cloud computing mainly focuses on virtualization. Deployment of the applications developed using the cloud can be done quickly. Load balancing is a method of splitting the workloads and resources among available servers. Load balancing improves the resource utilization of the servers, prevents overloading of servers, reduces the migration frequency and the network traffic. If many users try to access the resources at the same time, it becomes difficult to manage all the requests. Load balancing plays a vital role since it can handle sudden traffic spikes. Load balancing also maintains the firmness of the system, prevents system failures thereby improving system performance. Task scheduling is the allocation of resources and Load balancing in cloud computing involves optimal task scheduling. In this paper, we have proposed enhanced algorithms on how to implement load balancing with minimum CPU utilization time using CloudSim.

In CloudSim, the tasks are referred to as Cloudlets. Cloudlets are tasks that are ready to be executed in CloudSim.

2. LITERATURE REVIEW

Yang Jiao [1] proposed the planning and implementation of Distributed system-based webserver which aimed to extend the performance of the server. Moreover, as against the previous works, the scheme can further improve the security and prediction models. It improves the network performance of the server. The client request is assigned to the server that has the littlest ratio of the present number of connections to the weighted value. It used the ratio method to seek out the load of the server. The weighted value is then optimized and therefore the performance is tested. It mainly focuses on improving the safety of the distributed systems by using the Markov models to research the prediction models.

Gherbi et al [2] proposed a machine learning technique to classify virtual machines based on Central processing Unit (CPU) and Random Access Memory (RAM) utilization, to categorize user tasks based on size. The objective of our research to allow more resources dynamically to reduce the number of job turn downs. The group of virtual machines is generated in a data center bottomed upon resource requirements. The tasks with identical weights are grouped. Firstly, the tasks are classified into heavy, medium, or light tasks. Finally,the mapping of tasks to the convenient virtual machine group is done. The proposed algorithm follows steps include virtual machine utilization classification, retrieval of information from log files, computing the needed resource for all the task and grouping the tasks, assigning light virtual machine in the heavyweight or standard group, computing utilization of each virtual machine.The load balancer workload has been minimized by sharing the same virtual machine.

Mohammed Samiullah [3] proposed a load aware matrix approach in load balancing which reduce the waiting time of high-time consuming task, operation delay. The objective is to establish a dynamic load balancing algorithm and study response time, performance time, waiting time, scalability, power consumption, speed of virtual machines. It describes a min-min algorithm which requires a minimum time of complication and fewer resources, max-min algorithm where the task which requires more resources are executed first, Ant colony load balancing algorithm to distribute the load among virtual machines, Honey bee load balancing algorithm used in a heterogeneous environment where the newly arrived task assigned to the least loaded virtual machine, Resource-aware scheduling algorithm which is the combo of min-min and min-mix removed the fault in those algorithms, Active monitoring load balancing algorithm that has a load balancer which maintains increasing and decreasing connection in each node.

Chen et al [4] proposed a load balancing algorithm supported the server running state which calculated comprehensive loading consistent with network traffic, CPU utilization and memory utilization, and service. It followed load balancing strategies Random algorithm where the server is assigned in a random way to the client request, Round robin algorithm where the controller assign the server for the client request in a definite order, Server-based load balancing algorithm the system selects a system based on the probability, Open flow-based load balancing algorithm where all the clients use the virtual IP address to access the servers. It resulted that server-based load-balancing algorithm is better than the random or round-robin in assigning the user and balanced utilization of CPU for each server.

Foram et al [5] proposed a load balancing algorithm to improve the performance by dividing the jobs between various processors. Lined on the present status of the framework they grouped two algorithms, 1. Static load balancing algorithm and 2. Dynamic load balancing algorithm. In a static load balancing algorithm, the commonality of utilizations and assets of the framework is required. At the place of occupation appearance, the presentation of the virtual machines is settled. In a dynamic load balancing algorithm, the choice for load adjusting relies upon the present status of the framework which helps in improving the general execution of a framework by moving the load progressively. It additionally illustrated other static load balancing Round Robin, Equally Spread current augmentation load, Throttled load adjusting, and Adaptive asset allotment. It focused in on two boundaries first and foremost, load on the server and furthermore, the current performance of the server which expands customer fulfillment, asset usage, and lessening the reaction time and the energy utilization.

Jemimaand Varsha [6][7]proposed a max-min algorithm and a min-min algorithm for job scheduling. The minimum completion time for each task is calculated for all the machines. This process goes on until all the tasks are assigned to a machine. The task with overall minimum completion time is picked and assigned to the corresponding machine. The smaller task is executed first later the larger task are executed which resulted in poor machine use. The max-min algorithm is similar to a min-min algorithm. In max-min, the minimum completion time is computed for each job that with overall maximum completion time is picked and assigned to the corresponding machine.

Yu et al., [8] proposed a Stochastic Load Balancing for Virtual Resource Management in Datacenters where their scheme provides a probabilistic guarantee against overloading. It labels the resource distribution on virtual machine demand and resolves the hotspots and virtual migration with the demand of resources. The migration cost has been minimized in consideration of network topology and the performance is improved.

In the existing min-max algorithm, the cloudlets have to be sorted as per the completion time in the ascending order, which consumes lots of time as the tasks that have the largest completion time has to wait until all other small
tasks are completed and this might even leave the largest cloudlets incomplete. In the max-min algorithm, all the cloudlets that have the largest execution time are completed first. The cloudlet with the shortest execution time may have to wait until all the large tasks have completed its execution.

3. RELATED WORK\

Recently, several cloudlet-scheduling techniques are developed to unravel the scheduling problem. Two sorts of algorithms are developed; heuristic algorithms and meta-heuristic algorithms. Heuristic algorithms use predictions to realize a near-optimal solution [8]. On the opposite hand, the meta-heuristic methods directly search the answer space and produce efficient results on the broad domain problems, but these methods have time complexity [9]. Genetic Algorithm (GA) and Ant Colony Optimization (ACO) are famous algorithms. In [10], the authors use GA with ACO to develop a two-phase algorithm for the scheduling process. Within the first phase, the authors apply one GA generation and choose the only solution. Within the second phase, they apply the ACO algorithm, using the GA solution because the initial solution, to get a near-optimal solution.

4. CLOUDLET ASSIGNMENT ALGORITHM

To complete a task, one might need more computation time than the other tasks or even less which might lead to overload or underload. It is quite impossible to store the data too. This paper follows an algorithm where it uses the strategy of virtual machine scheduling. If the size of the cloudlet that needs to be executed is small, it is allocated to a virtual machine that has storage that could make the task execute optimally. Evolutionary algorithms play a vital role here. A parallel Genetic Algorithm, which is similar to the divide and conquers method is used. It splits the huge task into subtasks which get assigned to virtual machines. In a Parallel Genetic Algorithm, each process is not dependent, they run on their own. The selection of individuals from different groups and crossing them over is the crossover phase. The fitness function is applied where the cloudlet having the optimal fitness value is assigned to a virtual machine based on the PSO algorithm. For example, if the size of the task to be executed is 1GB, it can allocate to VM_1, VM_2, or VM_3 which has 1.5GB,2GB,3GB memory respectively. To avoid the wastage of memory it is allocated to VM_1 since it has the most optimal fitness value.

Cloudlet Assignment algorithm based on Parallel Genetic algorithm is as follows:

Cloudlet Assignment Algorithm

```
1.  Start
2.  Number_of_cloudlets = X;
3.  Number_of_Vm = X;
4.  Number_of_datacenter = X;
5.  startTime = System.nanoTime();
6.    //create datacenters, virtual machine and brokers.
7.    //create and submit cloudlets to the broker.
8.    //train the data
9.  population_set = Init
10. if (cloudlet < Vmsize)
11.  //apply fitness function
12.  //Assign to Virtual machines
13. Endtime = System.nanoTime();
14.  //Calculate execution time
15. return cloudlet_list;
16. Stop
```

This algorithm is illustrated by a graph in **Fig.1**on how the cloudlets are assigned to virtual machines.

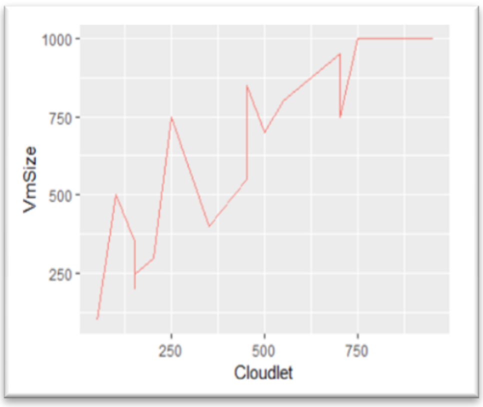

Fig.1 Cloudlet Assignment

5. ENHANCED MIN-MAX ALGORITHM

We have proposed an enhanced min-max algorithm where the cloudlets and the virtual machines are clustered based on each cloudlet’s execution time. K value was selected based on the server load and the distance between the sample points. It was found that the waiting time of the proposed algorithm took lesser time than that of the traditional min-max algorithm and max-min algorithm.

The Enhanced min-max algorithm is as follows:

Step 1: Start.
Step 2: Create datacenters, virtual machines and brokers.
Step 3: Compute the completion time of the cloudlets and
add to the cloudlet_list.
Step 4: Consider tasks as sample points.
Calculate the largest distance between the sample points which are far.
Step 5: Calculate the clustering centres.
Step 6: Cluster the cloudlets based on the clustering centres.
Step 7: Similarly cluster the virtual machines and assign the clustered cloudlets to them.
Step 8: Stop.

6. RESULT ANALYSIS

The proposed cloudlet assignment algorithm is found to be efficient since it saves the memory space and is similar to Support Vector Regression. It is illustrated in **Table 1**.

Table 1. Cloudlet Assignment Algorithm

Cloudlets (MIPS)	VMSize	Fitness Value
700	750	22.84
450	550	19.04
150	250	17.01
750	850	16.01
150	200	16.12
150	250	17.28
150	200	17.21
900	1000	21.87
100	200	18.36
950	1000	89.90
500	700	18.90
700	800	20.75
50	100	15.54
250	300	24.32
200	300	15.52

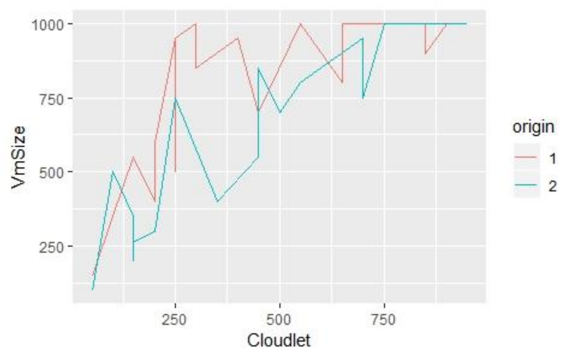

Fig.2 Comparison with the existing and the proposed algorithms

In **Fig.2**, 1 represents the existing algorithm and 2 represents the proposed cloudlet assignment.

Cloudlets are expressed in MIPS which is Million Instructions Per Second.

In comparison with the existing min-max algorithm and the proposed min-max algorithm, the total execution time is found to be efficient. Consider 10 cloudlets and the length of the cloudlets as in **Table 2**.

Table 2. Cloudlets and their length

Cloudlets	Length
T1	400
T2	600
T3	350
T4	750
T5	800
T6	950
T7	1000
T8	550
T9	650
T10	600

Table 3. Waiting time of the min-max algorithm

Cloudlets	ET (msec)	Waiting Time(msec)
T5	100	0
T1	120	100
T8	125	220
T9	140	345
T2	145	485
T6	150	630
T7	180	780
T10	200	960
T3	210	1160
T4	220	1370

Table 3. shows the waiting time of cloudlets in the min-max algorithm. All the cloudlets are sorted in ascending order based on the Execution Time (ET) such that the cloudlets(tasks) are executed in the above order. The cloudlet T4 has to wait for 1370 msec until all other tasks are executed.

Table 4. Waiting time of Enhanced min-max algorithm

Cloudlets	ET (msec)	VM	Waiting time (msec)
T5	100	VM1	0
T1	120	VM1	100
T8	125	VM1	220
T9	140	VM2	0
T2	145	VM2	140
T6	150	VM2	285
T7	180	VM3	0
T10	200	VM3	180
T3	210	VM3	380
T4	220	VM3	590

Table 4. shows the proposed min-max algorithm. The waiting time of all the cloudlets is reduced when they are clustered to virtual machines thereby, optimizing the traditional min-max algorithm.

7. SETUP ENVIRONMENT

1. Server Configuration:
1) Processor: Intel® Core™ i5-7200 CPU @ 2.50GHz 2.70GHz
2) RAM: 4.00 GB or more
3) Operating System: Windows 10 (64 bits)
2. CloudSim Version: CloudSim 4.0 – Library for simulation of cloud scenarios.
3. NetBeans: 7.1, JDK 1.8 – to integrate with the CloudSim.

8. CONCLUSION

This paper proposed a machine learning algorithms which reduce the computation costs, the ability to handle sudden traffic increase, and the overloading ofservers in the cloud. The enhanced algorithms developed and simulated in this project work are very efficient to improve the load balancing concept. It decreases the overall waiting time of the cloudlets. Thereby, the goal of load balancing is achieved. The future scope is to optimize the algorithm for further reduction of computation costs and network traffic spikes.

ACKNOWLEDGMENT
We express our deepest gratitude to our Phd Guide and Hod of our Computer Science and Engineering department who gaveconstant support and encouragement throughout this work.

REFERENCES

1. Yang Jiao, Wei Wang, “Design and Implementation of Load Balancing of Distributed-system-based Web Server”, IEEE, Third International Symposium on Electronic Commerce and Security,2010.
2. M.Elrotub, A. Gherbi, “Virtual machine classification-based approach to enhanced workload balancing for cloud computing applications”, Procedia Computer Science. 130, 683-688, 2018.
3. Muhammad Samiullah, “Load balancing in cloud computing”, URL: https://www.researchgate.net/publication/327744098_load_balancing_in_cloud_computingWenbo Chen, Zhihao Shang, Xinning Tian, et al., "Dynamic server cluster load balancing in the virtual environment with OpenFlow", International Journal of Distributed Sensor Networks 11(7), Article ID 531538, Volume 2015.
4. Foram F Kherani, Prof. Jignesh Vania, “Load Balancing in Cloud computing”, International Journal of Engineering Development and Research, Volume 2, Issue 1, pp.907-912, 2014.
5. R. Jemina Priyadarsini, L. Arockiam, “Performance Evaluation of Min-Min and Max-Min Algorithms for Job Scheduling in Federated Cloud”, International Journal of Computer Applications, Volume 99-No.18,pp. 47-54, 2014.
6. Varsha Soni, Dr. N.C. Barwar., “Performance Analysis of Enhanced Max-Min and Min-Min Task

Scheduling Algorithms in Cloud Computing Environment”, International Conference on Emerging Trends in Science, Engineering and Management, pp. 250-255, 2018.

7. S. H. H. Madni, M. S. A. Latiff, M. Abdullahi, S. M.Abdulhamid, M. J. Usman, Performance Comparison ofHeuristic Algorithms for Task Scheduling in IaaS CloudComputing Environment, PloS one, Vol. 12, No. 5, e0176321, May, 2017.
8. F. Villa, E. Vallada, L. Fanjul-Peyro, Heuristic Algorithms for the Unrelated Parallel Machine Scheduling Problem with One Scarce Additional Resource, Expert Systems with Applications, Vol. 93, pp. 28-38, March, 2018.
9. C. Y. Liu, C. M. Zou, P. Wu, A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization in Cloud Computing, Proceedings of the 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), Xian Ning, China, 2014, pp. 68-72.
10. L. Yu, L. Chen, Z. Cai, et al., “Stochastic load balancing for virtual resource management in datacenters”, IEEE Trans. Cloud Computing, pp. 459-472, 2016.
11. R. Buyya, R. Ranjan, and R. N. Calheiros, “Modeling andsimulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities,” in Proceedings of the 7th High Performance Computing and SimulationConference (HPCS’09). IEEE Press, NY, USA, 2009.
12. A. I. Awad, N. A. El-Hefnawy, H. M. Abdel_kader,
13. Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments, Procedia Computer Science, Vol. 65, pp. 920-929, 2015.
14. A. Assad, K. Deep, A Hybrid Harmony Search and Simulated Annealing Algorithm for Continuous Optimization, Information Sciences, Vol. 450, pp. 246-266, June, 2018.
15. Y. J. Moon, H. C. Yu, J. M. Gil, J. B. Lim, A Slave Ants Based Ant Colony Optimization Algorithm for TaskScheduling in Cloud Computing Environments, Humancentric Computing and Information Sciences, Vol. 7, No. 1, pp. 1-10, December, 2017.
16. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, R. Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Software: Practice and experience, Vol. 41, No. 1, pp. 23-50, January, 2011.
17. J. Moore, J. Chase, P. Ranganathan, and R. Sharma, “Making scheduling” cool”: temperature-aware workload placement in data centers,” 2005.
18. Oró E, Depoorter V, Garcia A, Salom J. Energy efficiency and renewable energy integration in data centres. Strategies and modelling review. Renewable and SustainableEnergy Review 2015; 42: 429-445.
19. Tatchell-Evans M, Kapur N, Summers J, Thompson H, Oldham D. An experimental and theoretical investigation of the extent of bypass air within data centres employingaisle containment, and its impact on power consumption. Applied Energy 2017; 186: 457-469.
20. UNFCCC. ICT Helping Tackle Climate Change Could Help Cut Global Emissions 20% by 2030. 2016. Available at: http://newsroom.unfccc.int/unfccc-newsroom/ict-sectorhelping-to-tackle-climate-change/ [Accessed: February 2017]
21. Whitehead B, Andrews D, Shah A, Maidment G. Assessing the environmental impact of data centres Part 1: Background, energy use and metrics. Building and Environment 2014; 82: 151-159.
22. Rong H, Zhang H, Xiao S, Li C, Hu C. Optimizing energy consumption for data centres. Renewable and Sustainable Energy Reviews 2016; 58: 674-691.
23. Ni J, Bai X. A review of air conditioning energy performance in data centres.Renewable and Sustainable Energy Reviews 2017; 67:625-640.
24. Bertoldi P, Hirl B, Labanca N. Energy Efficiency Status Report 2012. Electricity Consumption and Efficiency Trends in the EU-27. European Commission, Joint Research Centre 2012. Available at:http://publications.jrc.ec.europa.eu/repository/handle/JRC69638 [Accessed: November2016]
25. Bertoldi P. A Market Transformation Programme for Improving Energy Efficiency in Data Centres. ACEEE Summer Study on Energy Efficiency in Buildings; 2014: 9.14-9.26
26. N Thillaiarasu, S ChenthurPandian, “A novel scheme for safeguarding confidentiality in public clouds for service users of cloud computing” Cluster Computing 22 (1), 1179-1188