ISSN 2277-3061
Volume 16 Number 6
International journal of Computers and Technology

AN ENHANCED TASK ALLOCATION STRATEGY IN CLOUD ENVIRONMENT

Kavita Redishettywar [(1)], Prof. Rafik Juber Thekiya [(2)]
[(1)] Research Scholar, Department of Computer Science and Engineering, Matoshri Pratishthan Group of Institutions, Khupsarwadi, Nanded
[(2)] Assistant Professor, Department of Computer Science and Engineering, Matoshri Pratishthan Group of Institutions, Khupsarwadi, Nanded

ABSTRACT

Cloud computing is a vigorous technology by which a user can get software, application, operating system and hardware as a service without actually possessing it and paying only according to the usage. Cloud Computing is a hot topic of research for the researchers these days. With the rapid growth of Interne technology cloud computing have become main source of computing for small as well big IT companies. In the cloud computing milieu the cloud data centers and the users of the cloud-computing are globally situated, therefore it is a big challenge for cloud data centers to efficiently handle the requests which are coming from millions of users and service them in an efficient manner. Load balancing ensures that no single node will be overloaded and used to distribute workload among multiple nodes. It helps to improve system performance and proper utilization of resources. We propose an improved load balancing algorithm for job scheduling in the cloud environment using K-Means clustering of cloudlets and virtual machines in the cloud environment. All the cloudlets given by the user are divided into 3 clusters depending upon client's priority, cost and instruction length of the cloudlet. The virtual machines inside the datacenter hosts are also grouped into multiple clusters depending upon virtual machine capacity in terms of processor, memory, and bandwidth. Sorting is applied at both the ends to reduce the latency. Multiple number of experiments have been conducted by taking different configurations of cloudlets and virtual machine. Various parameters like waiting time, execution time, turnaround time and the usage cost have been computed inside the cloudsim environment to demonstrate the results. Compared with the other job scheduling algorithms, the improved load balancing algorithm can outperform them according to the experimental results.

Keywords

Cloud computing, Load balancing, Virtual machine, Host, Datacenter, Datacenter Broker

INTRODUCTION

Cloud computing [1] is an emerging paradigm in the computer industry where the computing is moved to a cloud of computers. It has become one of the buzz words of the industry. The core concept of cloud computing is, quite simply, that the vast computing resources that we need will reside somewhere out there in the cloud of computers and we'll connect to them and use them as and when needed. Computing can be described as any activity of using and/or developing computer hardware and software. It includes everything that sits in the bottom layer, i.e. everything from raw compute power to storage capabilities. Cloud computing [1] ties together all these entities and delivers them as a single integrated entity under its own sophisticated management.

Cloud computing is a combination of many computing fields and has gained much popularity in the recent years. Cloud computing provides computing, storage, services, and applications over the Internet. Moreover, cloud computing facilitates to reduce capital cost, decouple services from the underlying technology, and provides flexibility in terms of resource provisioning. Cloud computing has become very beneficial for business services, applications and other types of consumer requirements. Very large enterprises are practicing to scale back all their hardware and infrastructure on the cloud and for that reason most of them has already begin consolidating their IT operations and virtualization mechanisms and technologies on the cloud[1]. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing has claimed to jump the enterprise business to a brand-new level and permits them to cut back all the prices through improved production, reduced administration and infrastructure, architecture price and quicker preparation cycles. Cloud computing is a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. In cloud computing, the word cloud(also phrased as "the cloud") is used as a metaphor for "the Internet", so the phrase cloud computing means "a type of Internet-based computing", where different services such as servers, storage and applications are delivered to an organization's computers and devices through the Internet. Cloud computing is comparable to grid computing, a type of computing where unused processing cycles of all computers in a network are harnesses to solve problems too intensive for any stand-alone machine. The Cloud Computing may be a term that describes the infrastructure, platform, services and different kind of applications. As this is a platform it reconfigures servers or applications where the server can be a physical machines or virtual display machines. Cloud computing is different from ancient computing paradigms because it is customizable, scalable, encapsulated, abstract entity that gives totally different level of services, processes to the clients, driven by economies of scale and also the services area unit dynamically and totally configurable [2].

http://cirworld.com/ DOI:10.24297/ijct.v16i6.6304

LOAD BALANCING

One of the foremost usually used applications of load balancing is to produce quality of service from multiple servers, typically called a server data center. Usually load-balanced systems are properly working inside popular internet sites, big chat networks, high-bandwidth file transfer protocol sites, and domain name System (DNS) servers. It additionally prevents the clients from contacting back-end servers directly, which can have security advantages by hiding the structure of the inner network. Some load balancers give a mechanism for improving the one parameter specially within back end server Load balancing offers the IT team an opportunity to attain a considerably higher fault tolerance. It will mechanically give the capability required to handle any increase or decrease of application traffic. It is additionally necessary that the load balancer itself doesn't become the cause of failure. Sometimes load balancers enforced in high-availability servers can additionally replicate the user's session needed by the application. Load balancing is dividing work load between a set of computers in order to receive the good response time and all the nodes are equally loaded and, in general, all users get served quicker. Load balancing may be enforced with hardware, software, or a mix of each. Typically, load balancing is that the main reason for server's unbalanced response time.

RELATED WORK

The most of the researches have been working in the area of load balancing in cloud computing process for enhancing the overall performances of the clouds. Some of these tasks should contain the improved traditional mechanisms to achieve the objective of load balancing. So as to appreciate their contribution, determination and better understandability the work ahead.

Al-Rayis et al. [1] explains that basically, load balancers can be deployed based on three different architectures. The centralized load balancing architecture which includes a central load balancer to make the decision for the entire system regarding which cloud resource should take what workload and based on which algorithm(s).

Bhoi et al. [2] discussed that in enhanced Max-Min Task Scheduling Algorithm in cloud computing helps in supplying a high performance computing based on protocols which allowed shared computation and storage over long distances. It depends upon expected execution time instead of completion time. Max-Min algorithm assign task with maximum execution time to resource produces minimum completion time while Enhanced Max-min assign task with average execution time to resource produces minimum execution time.

Bhadani et al. [3] proposed a Central Load Balancing Policy for Virtual Machines (CLBVM) that balances the load evenly in a distributed virtual machine/cloud computing environment.

Bendiab et al. [4] introduced the Map Reduced based Entity Resolution load balancing technique in networking which is based on large datasets. In this technique, two main tasks are done: Map task and Reduce task which the author has described.

Birattari et al. [5] proposed troubleshoot of load balance in Cloud computing using Stochastic Hill Climbing.

Buzato et al. [6] proposed Bee Life algorithm which was used for scheduling in Cloud computing. Bee Life algorithm is inspired by the behavior and reproduction of bee to find food source. The algorithm evaluated the performance of the resources and it has the aim to reduce time and complexity of work.

Babu et al. [7] proposed a Honey Bee Behavior inspired Load Balancing [HBB-LB] technique which helps to achieve even load balancing across virtual machine to maximize throughput. It considers the priority of task waiting in queue for execution in virtual machines.

Dorigo et al. [8] has proposed a load balancing technique called colony of cooperating agents in ants based on soft computing for solving the optimization problem. This technique solves the problem with high probability. It is a simple loop moving in direction of increasing value which is uphill. And this make minor change in to original assignment according to some criteria.

Deldari et al. [9] proposed a novel load balancing algorithm called VectorDot in intelligent ants. It handles the hierarchical complexity of the datacenter and multidimensionality of resource loads across servers, network switches, and storage in an agile data center that has integrated server and storage virtualization technologies.

Desai et al. [10] discusses about the emerging technology i.e. a new standard of large scale distributed computing and parallel computing. It provides shared resources, information or other resources as per clients' requirements at specific times. For better management of available good load balancing techniques are required. And through better load balancing in cloud, performance is increased and user gets better services. So in this author has discussed many different load balancing techniques used to solve the issue in cloud computing environment.

Elzeki et al. [11] discussed in Improved Max-Min Algorithm in Cloud Computing that focuses on the cloud computing which further deals with the allocation of the tasks to the resources while observing different parameters like waiting time, Average waiting time, Turn Around time, processing cost. So, an algorithm named as Max-Min in improved manner from load balancing has been shown to overcome such kinds of problems.

Fahringer et al. [12] introduced a static load balancing technique called Ant Colony Optimization. In this technique, an ant starts the movement as the request is initiated. This technique uses the Ants behavior to collect information of cloud node to assign task to the particular node. In this technique, once the request is initiated, the ant and the pheromone starts the forward movement in the pathway from the "head" node.

Fang et al. [13] discussed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing by first mapping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud computing environment.

OBJECTIVES

The primary objectives of this research work are summarized as follows:

• To study the performance of existing load balancing algorithm.

• To propose a new efficient load balancing with clustering at both sides i.e client side and server side.

• To apply the sorting mechanism on the clusters formed at the client as well as cloud side.

• To reduce the overhead time of scanning the entire VM's in a cluster by arranging them in descending order.

• To implement the concept of priority based execution depending upon client's cost.

• To implement the proposed algorithm in cloudsim simulator.

• To evaluate the performance of proposed algorithm with current algorithm.

The Cloud network consist multiple users input with their different requirements which needed to fulfill by efficiently utilizing the available resources. There are different ways to fulfill user's requirement (Like priority). The main objective of this research work is to answer the question: in identical cloud environments, which load balancing architecture: centralized, decentralized or hierarchical architecture will give the best results in terms of response time and server load To answer this question a robust evaluation framework is implemented which includes the following steps:

- To balance the load equally among different VMs.
- Fetch all the available virtual machines in the datacenter/host.
- Retrieve the processing capacity of the available virtual machines.
- Clustering at the cloudlet side is done on the basis of the user requirements in terms of cloudlet length and cost.
- High, medium, low priority is assigned to tasks and priority is directly proportional to cost i.e. high is the priority, more cost will be charged.
- Cluster to cluster assignment of cloudlets is done which reduces the time as compare when cloudlets were assigned one by one.
- Descending order is applied at both side within clusters for maximizing the benefit.
- The task is allocated with the help of load balancing algorithm.

In this strategy, current system state plays major role while making decisions. Despite the fact that dynamic load balancing has higher run rime complexity then static one, dynamic has better performance report as it considers current load of system for choosing next datacenter to serve the request. This will surely provide an optimal choice from available ones for that state of system. Workload in the cloud is regularly a multi-objective problem. In this thesis we will highlight and pay attention to some of these problem and possible solution, so as to obtain an optimal solution. We expect that every application comprises of a number of slightly parallel tasks. Every application has a strict fulfillment due time. Prior to this due time, all computational assignments in the application must be completely executed with the results conveyed to the client. Our current application model concentrates on random sort of workloads. With two different clients group one with higher resources accessing rights while other group has relatively lower resources accessing rights.

RESEARCH METHODOLOGY

Cloud services provide computing on demand in real time. Number of users accessing cloud environment are always more than that were using it on previous day. Cloud has application areas for developing applications, providing and managing infrastructure, patching applications. Users and their requests for accessing cloud infrastructure are highly dynamic and loading servers running in data center. We need efficient strategy to balance load on these servers so that the servers don't get crash and they can persist long. Precisely Objective is to achieve accuracy, performance of servers and the cloud environment can be maintained.

Steps:

1. Initialize the Cloud Sim in Java
2. Create the Datacenter with different number of hosts.
3. Each Host will have the different numbers of Virtual machines of different capacities.
4. Then we will create the Cloudlets of varying length and size.
5. The list containing the Virtual machines [18] and Cloudlets will be given to the Data Center Broker (DCB)

6. DCB will compute the processing capacity of all the virtual machines and will divide them into multiple clusters using K-Means clustering by using various parameters like bandwidth, memory and processing capability.

7. Data center broker will maintain the list of the cloudlets and will also divide them into multiple clusters using K-Means Clustering. Various parameters used over here are cloudlet length, priority of the cloudlet and the cost associated with it.

8. All the cloudlets and the virtual machines inside the clusters are sorted in descending order.

9. Dispatch our cloudlet to appropriate virtual machine in the cluster. Since all the cloudlets and virtual machines are sorted in descending order, so the cloudlet with higher instruction size and higher priority will be assigned to the virtual machine with higher resources.

10. Repeat the same procedure for all the remaining cloudlets in the list.

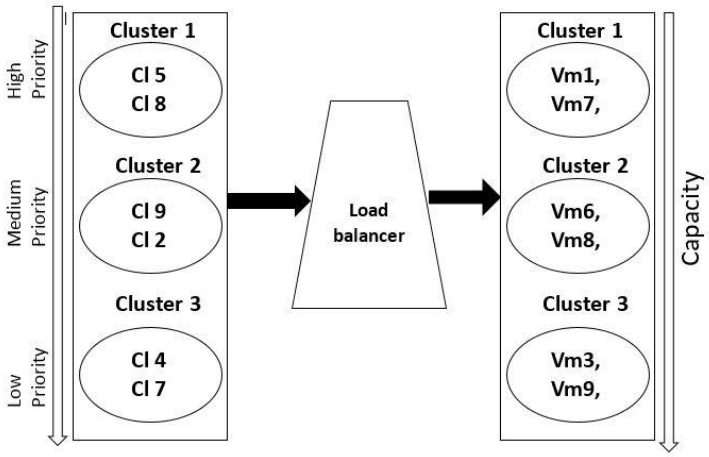

Figure 1. Proposed Load Balancing Model

ALGORITHM

Algorithm of the proposed work is written as follows:

Input:-Unallocated Tasks/Cloudlets/Virtual Machines.

Output:-Response Time, Waiting Time, Processing Cost.

Algorithm:

1. Input the Cloudlets(CL) to the CloudSim.
2. foreach Cloudlet k in CL.
   find the Instruction length, priority/deadline and cost of k.
   end
3. Start the K-Means Clustering and divide them into 3 clusters.
4 Assume Centroid A, B, C
5. foreach Cloudlet k in CL
   Calculate Eucledian Distance of k with A, B and C.
   if previous distance = new distance
   {
   Stop Iterations
   }
   else
   {
   Add Ck into minimum clustered distance
   Again Compute the Centroids
   }
   end for

6. Create Virtual machines (VMs) in the CloudSim.
7. foreach VM m in VMs.
   find the capacity, memory and bandwidth of each m.
   end
8. Start the K-Means Clustering and divide virtual machines into 3 clusters.
9. Assume Centroid I, J, K
10. foreach VM k in VMs
    Calculate Eucledian Distance of k with I, J and K
    if previous distance = new distance
    {
    Stop Iterations
    }
    else
    {
    Add Vk into minimum clustered distance
    Again Compute the Centroids
    }
    end for
11. Apply descending sort on VMs and Cloudlets in Clusters.
12. foreach Cloudlet j
    assign Cloudlet j to VMj of appropriate cluster
    increment the VM.
    if VMmax >=ListSize
    {
    VMindex =0;
    }
    end for

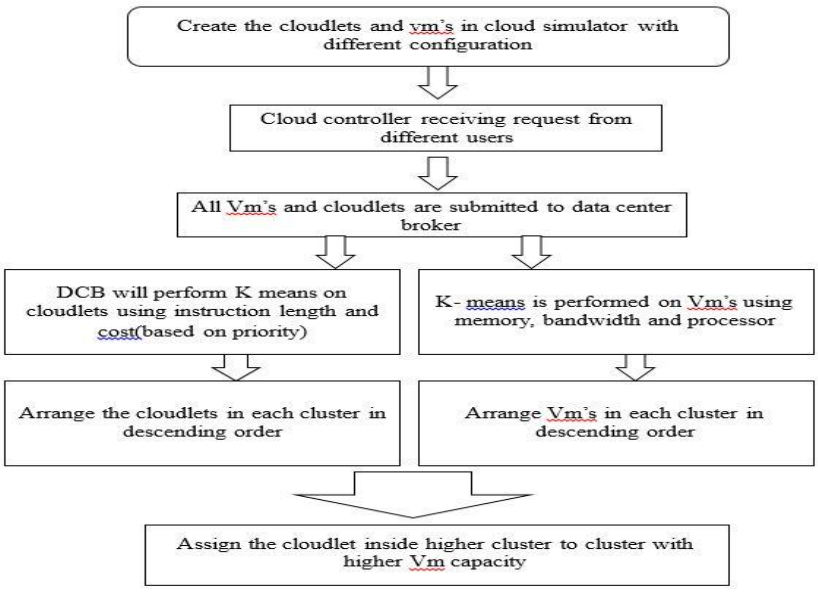

Figure 2. Flow Chart of the Proposed Work

Cloud Sim [4] could be an illustrious machine for cloud parameters developed within the CLOUDS Laboratory, at the pc Science and computer code Engineering Department of the University of Melbourne. The Cloud Sim library is employed for the subsequent operations: massive scale cloud computing at information centers. Cloud Computing includes the virtualized server machines/hosts with all types of modifiable policies. It has the support for simulation and modeling of very large scale knowledge/information centers.

We need efficient strategy to balance load on these servers so that the servers don't get crash and they can persist long. Precisely Objective is to achieve accuracy, performance of servers and the cloud environment can be maintained. Various experiments have been conducted and the results of existing work and the proposed work have been mentioned in the Table 1 and Table 2

EXPERIMENTAL RESULTS

Table 1. Results of Existing Work

Sno.	No of Cloudlets	Total Turn Around Time	Total Execution Time	Total Waiting Time	Average Waiting Time	Total Processing Cost
1	5	10	9	1	0	28
2	10	19	18	1	0	56
3	40	187	73	114	3	223
4	60	390	110	280	5	335
5	100	1015	183	832	8	558
6	150	2207	274	1933	13	837
7	200	3857	366	3491	17	1115
8	300	8525	548	7977	27	1673
9	400	15020	731	14289	36	2230
10	500	23347	914	22433	45	2788
11	700	45473	1279	44194	63	3903
12	1000	92361	1827	90534	91	5575
13	2000	367388	3654	363734	182	11148
14	3000	825076	5481	819596	273	16721
15	5000	2288438	9134	2279304	456	27867
16	7000	4482446	12787	4469659	639	39013
17	8000	5853443	14614	5838829	730	44586
18	9000	7407101	16440	7390660	821	50159
19	10000	9143420	18267	9125153	913	55732
20	20000	36553005	36533	36516471	1826	111463
21	30000	82228747	54799	82173947	2739	167193
22	40000	146170648	73066	146097582	3652	222923
23	50000	228378712	91332	228287380	4566	278654
24	60000	328852930	109598	328743332	5479	334384
25	70000	447593308	127865	447465443	6392	390115

In the table 2, we have mentioned the results of the various experiments conducted on the proposed algorithm by taking the same configuration of virtual machines and the same properties of the cloudlets.

Table 2. Results of Proposed Work

Sno.	No of Cloudlets	Total Turn Around Time	Total Execution Time	Total Waiting Time	Average Waiting Time	Total Processing Cost
1	5	5	5	1	0	15
2	10	15	10	5	0	30
3	40	161	41	120	3	126
4	60	351	63	288	5	191
5	100	923	104	820	8	317
6	150	2052	157	1895	13	478
7	200	3590	208	3381	17	636
8	300	8020	313	7706	26	956
9	400	14136	417	13719	34	1273
10	500	22029	522	21507	43	1592
11	700	43015	731	42285	60	2229
12	1000	87560	1044	86516	87	3185
13	2000	349217	2088	347129	174	6371
14	3000	785170	3133	782037	261	9559
15	5000	2178548	5222	2173326	435	15931
16	7000	4268335	7310	4261025	609	22304
17	8000	5574486	8355	5566131	696	25490
18	9000	7055333	9400	7045933	783	28678
19	10000	8708634	10443	8698190	870	31863
20	20000	34824312	20888	34803424	1740	63728
21	30000	78349034	31332	78317702	2611	95594
22	40000	139275024	41776	139233249	3481	127458
23	50000	217611392	52220	217559173	4351	159323
24	60000	313358803	62664	313296139	5222	191189
25	70000	426502150	73108	426429043	6092	223052

Figure 5.1 illustrates the turn around time for the existing work and proposed work. It is clear from the line chart, that the turn around times of the cloudlets have been reduced, thereby increasing the overall efficiency of the system.

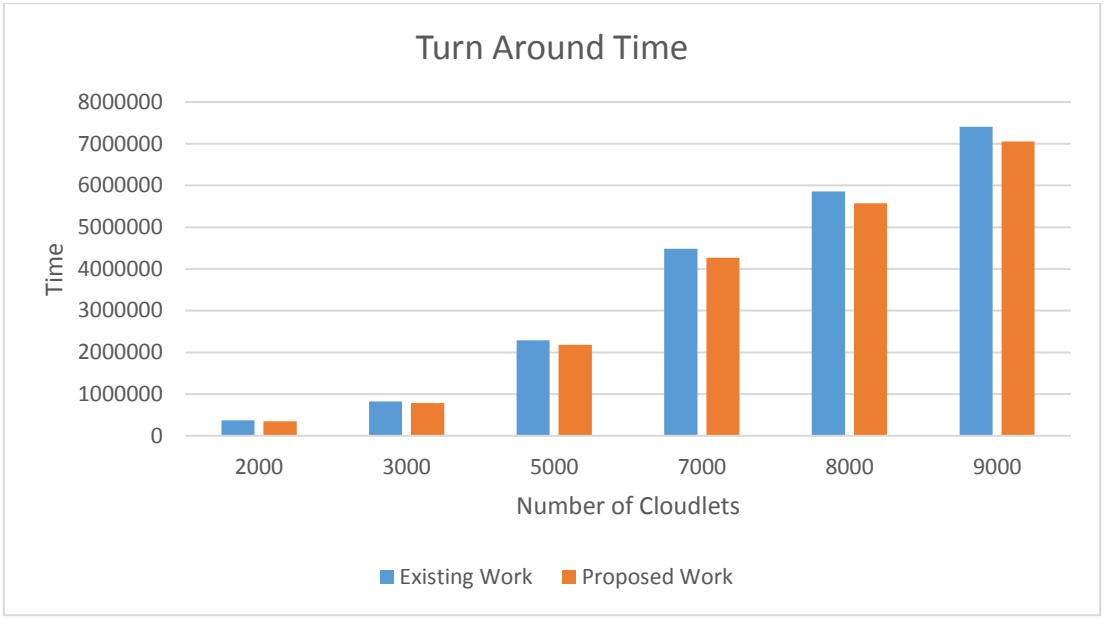

Figure 3. Turn Around Time

From the above bar chart in figure 5.2, it is clear that the processing time has been reduced.

Response Time

- It is defined as total time taken by a load balancing algorithm to finish the execution of a cloudlet.
- Response Time (RT) = FT – ST

where,

FT = finish time of execution

ST = start time of execution

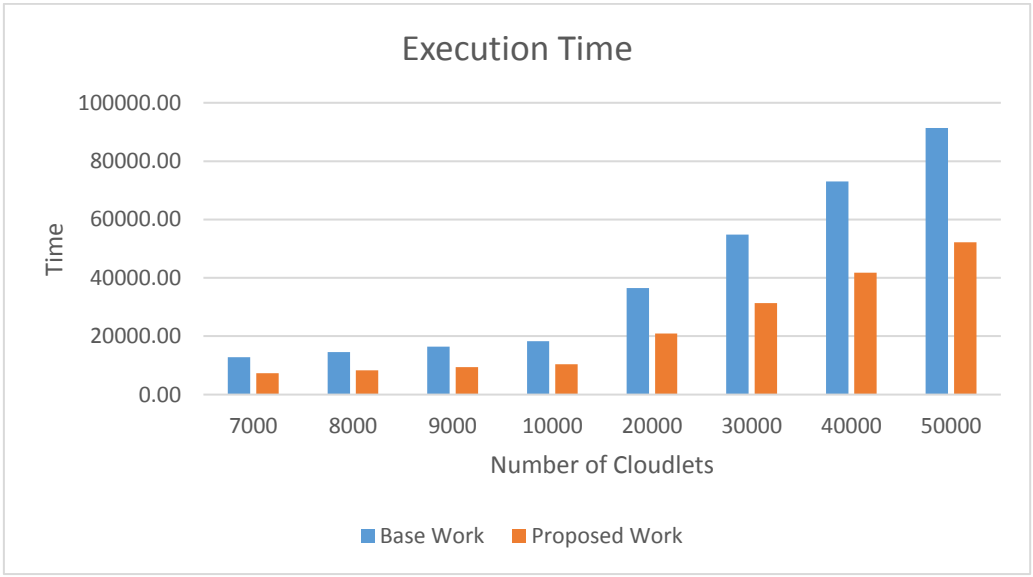

Figure 4. Execution Time of Proposed Algorithm

From the bar chart in figure 5.3, it is clear that the processing cost has been reduced.

Processing Cost

It is obtained by addition of cost per storage, cost per memory and cost per memory.

Processing Cost = RT * unit_cost.

where, RT = response time

unit_cost = cost per unit time

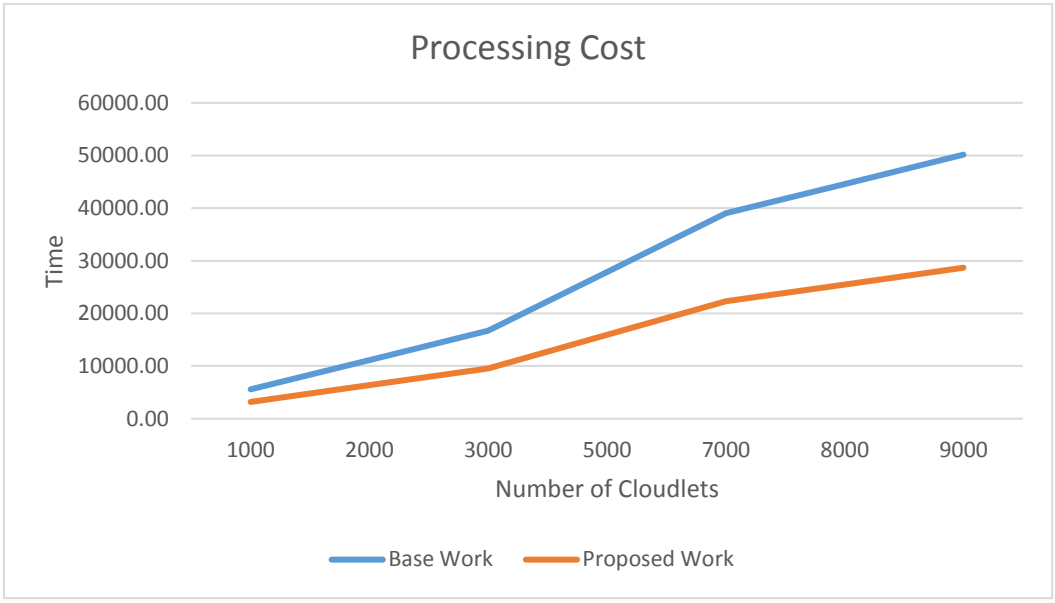

Figure 5. Processing Cost

Figure 5.4 illustrates the waiting time for the base work and proposed work. It is clear from the line chart, that the waiting times of the cloudlets have been reduced, thereby decreasing the cost and increasing the overall efficiency of the system.

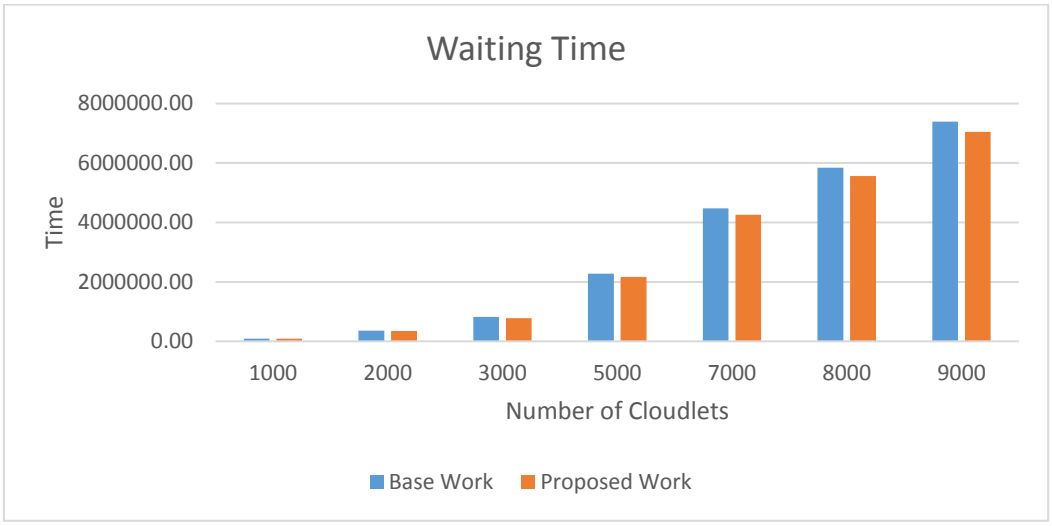

Figure 6. Waiting Time

CONCLUSION

The performance of improved load equalization algorithmic program has been studied in this research work. The request time for the policies applied are same which suggests there's no impact on data centers request time after modifying the algorithm. The processing cost, turnaround time, waiting time, execution time are calculated using various number of experiments. The experiments conducted are compared with previous algorithms. The results indicate that the approaches surpass to previous algorithmic program in terms of execution time, waiting time and turnaround time. The experimental results are obtained by applying the new planned algorithmic program within the Cloud Sim simulator developed in java programming language, shows that the new work has outperformed the present programming algorithms in giant scale distributed systems. To get a much better answer and more precise results, the model ought to be created a lot of realistic by considering problems regarding load equalization like information section, communication price and flow time and results may be tested in real cloud setting. Moreover, fault tolerance, virtual machine migration and the power consumed by the virtual machine are also the considerable factors that can be explored in the future work.

REFERENCES

[1] S. Yakhchi, S. Ghafari, M. Yakhchi, M. Fazeli and A. Patooghy, "ICA-MMT: A Load Balancing Method in Cloud Computing Environment," IEEE, 2015.

[2] S. Kapoor and D. C. Dabas, "Cluster Based Load Balancing in Cloud Computing," IEEE, 2015.

[3] S. Garg, R. Kumar and H. Chauhan, "Efficient Utilization of Virtual Machines in Cloud Computing using Synchronized Throttled Load Balancing," 1st International Conference on Next Generation Computing Technologies (NGCT-2015), pp. 77-80, 2015.

[4] R. Panwar and D. B. Mallick, "Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm," IEEE, pp. 773-778, 2015.

[5] M. Belkhouraf, A. Kartit, H. Ouahmane, H. K. Idrissi,, Z. Kartit and M. . E. Marraki, "A secured load balancing architecture for cloud computing based on multiple clusters," IEEE, 2015.

[6] L. Kang and X. Ting, "Application of Adaptive Load Balancing Algorithm Based on Minimum Traffic in Cloud Computing Architecture," IEEE, 2015.

[7] N. K. Chien, N. H. Son and H. D. Loc, "Load Balancing Algorithm Based on Estimating Finish Time of Services in Cloud Computing," ICACT, pp. 228-233, 2016.

[8] H. H. Bhatt and H. A. Bheda, "Enhance Load Balancing using Flexible Load Sharing in Cloud Computing," IEEE, pp. 72-76, 2015.

[9] S. S. MOHARANA, R. D. RAMESH and D. POWAR, "ANALYSIS OF LOAD BALANCERS IN CLOUD COMPUTING," International Journal of Computer Sciencand Engineering (IJCSE) , pp. 102-107, 2013.

[10] M. P. V. Patel, H. D. Patel and . P. J. Patel, "A Survey On Load Balancing In Cloud Computing," International Journal of Engineering Research & Technology (IJERT), pp. 1-5, 2012.

[11] R. Kaur and P. Luthra, "LOAD BALANCING IN CLOUD COMPUTING," Int. J. of Network Security, , pp. 1-11, 2013.

[12] Kumar Nishant, , P. Sharma, V. Krishna, Nitin and R. Rastogi, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization," IEEE, pp. 3-9, 2012.

[13] Y. Xu, L. Wu, L. Guo,, Z. Chen, L. Yang and Z. Shi, "An Intelligent Load Balancing Algorithm Towards Efficient Cloud Computing," AI for Data Center Management and Cloud Computing: Papers from the 2011 AAAI Workshop (WS-11-08), pp. 27-32, 2011.

[14] A. K. Sidhu and S. Kinger, "Analysis of Load Balancing Techniques in Cloud Computing," International Journal of Computers & Technology Volume 4 No. 2, March-April, 2013, ISSN 2277-3061 , pp. 737-741, 2013.

[15] O. M. Elzeki , M. Z. Reshad and M. A. Elsoud , "Improved Max-Min Algorithm in Cloud Computing," International Journal of Computer Applications (0975 – 8887), pp. 22-27, 2012.

[16] B. Kruekaew and W. Kimpan, "Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony," Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol I,IMECS 2014, 2014.

[17] R.-S. Chang, J.-S. Chang and P.-S. Lin, "An ant algorithm for balanced job scheduling in grids," Future Generation Computer Systems 25 (2009) 20–27, pp. 21-27, 2009.

[18] Z. Chaczko, V. Mahadevan, S. Aslanzadeh and C. Mcdermid, "Availability and Load Balancing in Cloud Computing," International Conference on Computer and Software Modeling IPCSIT vol.14 (2011) © (2011) IACSIT Press, Singapore, pp. 134-140, 2011.

[19] R. K. S, S. V and V. M, "Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud," Special Issue of International Journal of Computer Applications (0975 – 8887) on Advanced Computing and Communication Technologies for HPC Applications - ACCTHPCA, June 2012, pp. 31-35, 2012.

[20] Kumar Nishant, , P. Sharma, V. Krishna, N. and R. Rastogi, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization," IEEE, pp. 3-9, 2012.

DOI:10.24297/ijct.v16i6.6304