



ENHANCED DYNAMIC RESOURCE ALLOCATION SCHEME BASED ON PACKAGE LEVEL ACCESS IN CLOUD COMPUTING

Manpreet Kaur⁽¹⁾, Dr. Rajinder Singh⁽²⁾

⁽¹⁾ Research Scholar, Department of Computer Science & Engineering, GGSCET(Guru Kashi University), Talwandi Sabo, Bathinda, Punjab.
manpreet12395@gmail.com

⁽²⁾ Assistant Professor, University College of Computer Applications, Guru Kashi University, Talwandi Sabo, Bathinda, Punjab.
rajneel2807@gmail.com

ABSTRACT

Cloud computing is distributed computing, storing, sharing and accessing data over the Internet. It provides a pool of shared resources to the users available on the basis of pay as you go service that means users pay only for those services which are used by him according to their access times. This research work deals with the balancing of work load in cloud environment. Load balancing is one of the essential factors to enhance the working performance of the cloud service provider. 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. We propose an improved load balancing algorithm for job scheduling in the cloud environment using load distribution table in which the current status, current package, VM Capacity and the number of cloudlets submitted to each and every virtual machine will be stored. Submit the job of the user to the datacenter broker. Data center broker will first find the suitable Vm according to the requirements of the cloudlet and will match and find the most suitable Vm according to its availability or the machine with least load in the distribution table. 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 cloudsims environment to demonstrate the results. The main contributions of the research work is to balance the entire system load while trying to minimize the make span of a given set of jobs. 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 (Al-Rayis, 2013) 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 (Al-Rayis, 2013) ties together all these entities and delivers them as a single integrated entity under its own sophisticated management.

Cloud is a term used as a metaphor for the wide area networks (like internet) or any such large networked environment. It came partly from the cloud-like symbol used to represent the complexities of the networks in the schematic diagrams. It represents all the complexities of the network which may include everything from cables, routers, servers, data centers and all such other devices. Computing started off with the mainframe era. There were big mainframes and everyone connected to them via "dumb" terminals. This old model of business computing was frustrating for the people sitting at the dumb terminals because they could do only what they were "authorized" to do. They were dependent on the computer administrators to give them permission or to fix their problems. They had no way of staying up to the latest innovations. The personal computer was a rebellion against the tyranny of centralized computing (Bendiab, 2010) operations. There was a kind of freedom in the use of personal computers. But this was later replaced by server architectures with enterprise servers and others showing up in the industry. This made sure that the computing was done and it did not eat up any of the resources that one had with him. All the computing was performed at servers. Internet grew in the lap of these servers. With cloud computing we have come a full circle. We come back to the centralized computing infrastructure. But this time it is something which can easily be accessed via the internet and something over which we have all the control. Cloud computing is Internet ("cloud") based development and use of computer technology ("computing"). It is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. Users need not have knowledge of, expertise in, or control over the technology infrastructure "in the cloud" that supports them.

Cloud computing is often confused with grid computing ("a form of distributed computing whereby a 'super and virtual computer' is composed of a cluster of networked, loosely-coupled computers, acting in concert to perform very large tasks"), utility computing (the "packaging of computing resources, such as computation and storage, as a metered service similar to a traditional public utility such as electricity") and autonomic computing ("computer systems capable of self-management").



Figure 1. Model of Cloud Computing

TYPES OF CLOUDS

Clouds are divided into 4 categories: -

- 1) Public Cloud: - Public cloud (Deldari, 2006) allows users to access the cloud publicly. It is access by interfaces using internet browsers. Users pay only for that time duration in which they use the service, i.e., pay-per-use.
- 2) Private Cloud: - A private clouds (Desai, 2013) operation is with in an organization's internal enterprise data center. The main advantage here is that it is very easier to manage security in public cloud. Example of private cloud in our daily life is intranet.
- 3) Hybrid Cloud: - It is a combination of public cloud (Elzeki, 2012) and private cloud. .It provide more secure way to control all data and applications .It allows the party to access information over the internet. It allows the organization to serve its needs in the private cloud and if some occasional need occurs it asks the public cloud for some computing resources.
- 4) Community Cloud: -When cloud infrastructure construct by many organizations jointly, such cloud model is called as a community cloud. The cloud infrastructure could be hosted by a third-party provider or within one of the organizations in the community.

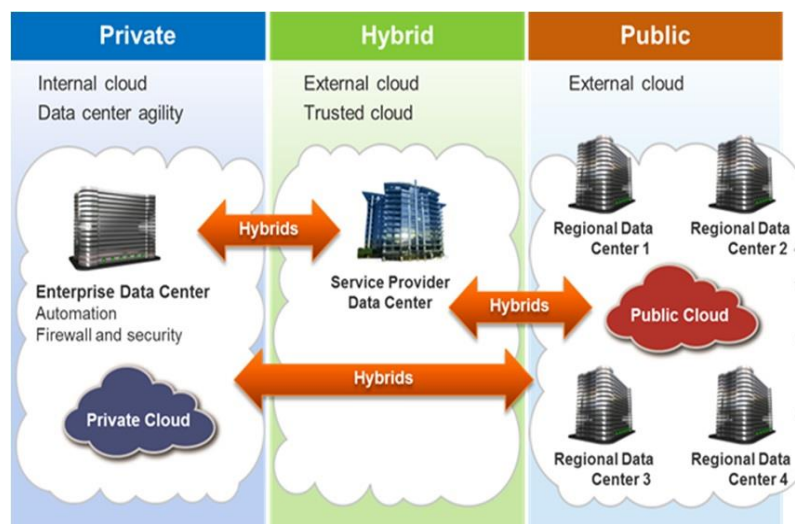


Figure 2. Types of Cloud

SERVICES OF CLOUD MODEL

There are different types of services are providing by cloud models like: Software as a Service(SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) (Buzato, 2011) which are deployed as public cloud, private cloud, community cloud and hybrid clouds.

- 1) Software as a Service (SaaS): - The capability provided to the consumer is to use some applications which is running on a cloud infrastructure. The applications are accessible from many devices through an interface such as a web browser (e.g., web-based email). The consumer does not control the cloud infrastructure which includes network, and servers, all operating systems, and provides storages.

2) Platform as a Service (PaaS): - PaaS (Birattari, 2006) provides all the resources that are required for implementation of applications and all services completely from the Internet. In this no downloading or installing is required of any software. The capability provided to the consumer is to deploy onto the cloud infrastructure. Consumer uses all the applications by using different programming languages and tools which are provide by the provider. Any consumer has not any control on cloud infrastructure including all networks, servers and operating systems, but has control over the applications which they deployed.

3) Infrastructure as a Service (IaaS): - The capability provided to the consumer is to access all the processing, storage, networks and other many fundamental computing resources. Consumer is able to deploy arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed application, and possibly limited control of select networking components. Infrastructure Providers manage a large set of computing resources, which include as storing and processing capacity. In this virtualization is used to split, assign and dynamically resize these resources. They deploy only those software stacks that run their required services. It can be used to avoid buying, housing, and managing the basic hardware and software infrastructure components, scales up and down quickly to meet demand.

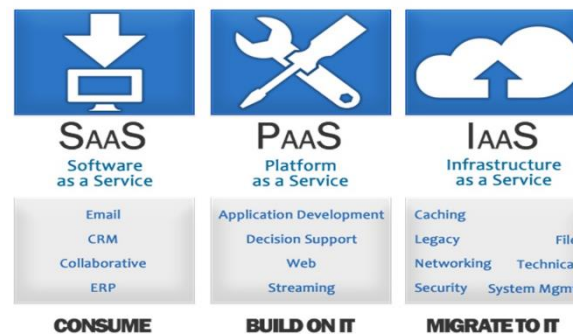


Figure 3. Example of Services provided by Cloud

LOAD BALANCING

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. Load balancing plans to optimize the usage of resources, maximize overall success ratio, minimize waiting time interval, and evade overloading of the resources. By the utilization of multiple algorithms and mechanisms with load balancing rather than one algorithm might increase reliability and efficiency. Load balancing within the cloud differs from classical thinking on load balancing design and implementation by misusage of data center servers to perform the requests on the basis of first come first serve basis. The older load balancing algorithm allocates the requests according to the incoming requests of the client. Load balancing is one amongst the central problems in cloud computing. It's a mechanism that distributes the dynamic workload equally across over the nodes or virtual machines within the whole cloud server to avoid a state of conflict wherever some virtual machines are measured as heavily loaded whereas others nodes or hosts are measured as idle or doing very little work. It helps to realize a high client satisfaction and resource utilization magnitude relation, consequently increasing the performance and resource utility of the system.

LITERATURE OVERVIEW

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.

Nguyen KhacChien et al. (2016) has proposed a load balancing algorithm which is used to enhance the performance of the cloud environment based on the method of estimating the end of service time. They have succeeded in enhancing the service time and response time of the user. S.Yakhchi et al. (2015) discusses that the energy consumption has become a major challenge in cloud computing infrastructures. They proposed a novel power aware load balancing method, named ICAMMT to manage power consumption in cloud computing data centers. We have exploited the Imperialism Competitive Algorithm (ICA) for detecting over utilized hosts and then we migrate one or several virtual machines of these hosts to the other hosts to decrease their utilization. Surbhi Kapoor et al. (2015) aims at achieving high user satisfaction by minimizing response time of the tasks and improving resource utilization through even and fair allocation of cloud resources. The traditional Throttled load balancing algorithm is a good approach for load balancing in cloud computing as it distributes the incoming jobs evenly among the VMs. But the major drawback is that this algorithm works well for environments with homogeneous VMS, does not considers the resource specific demands of the tasks and has additional overhead of scanning the entire list of VMs every time a task comes. The issues have been addressed by proposing an algorithm Cluster based load balancing which works well in heterogeneous nodes environment, considers resource specific demands of the tasks and reduces scanning overhead by dividing the machines into clusters. ShikhaGarg et al. (2015) aims to distribute workload among multiple cloud systems or nodes to get better resource utilization. It is the prominent



means to achieve efficient resource sharing and utilization. Load balancing has become a challenge issue now in cloud computing systems. To meet the user's huge number of demands, there is a need of distributed solution because practically it is not always possible or cost efficient to handle one or more idle services. ReenaPanwar et al. (2015) describes that the cloud computing has become essential buzzword in the Information Technology and is a next stage the evolution of Internet, The Load balancing problem of cloud computing is an important problem and critical component adequate operations in cloud computing system and it can also prevent the rapid development of cloud computing. Mohamed Belkhouraf et al. (2015) aims to deliver different services for users, such as infrastructure, platform or software with a reasonable and more and more decreasing cost for the clients. To achieve those goals, some matters have to be addressed, mainly using the available resources in an effective way in order to improve the overall performance, while taking into consideration the security and the availability sides of the cloud. Hence, one of the most studied aspects by researchers is load balancing in cloud computing especially for the big distributed cloud systems that deal with many clients and big amounts of data and requests. The proposed approach mainly ensures a better overall performance with efficient load balancing, the continuous availability and a security aspect. Lu Kang et al. (2015) improves the weighted least connections scheduling algorithm, and designs the Adaptive Scheduling Algorithm Based on Minimum Traffic (ASAMT). ASAMT conducts the real-time minimum load scheduling to the node service requests and configures the available idle resources in advance to ensure the service QoS requirements. Being adopted for simulation of the traffic scheduling algorithm, OPNET is applied to the cloud computing architecture. Hiren H. Bhatt et al. (2015) presents a Flexible load sharing algorithm (FLS) which introduces the third function. The third function makes partition the system in to domain. This function is helpful for the selection of other nodes which are present in the same domain. By applying the flexible load sharing to the particular domains in to the distribute system, the performance can be improved when any node is in overloaded situation. Desai et al. (2013) 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. Al-Rayis et al. (2013) 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). In the hierarchical load balancing architecture, a main load balancer (parent) receives all job requests, and then it spreads them to other connected load balancers (children) where each load balancer in the tree may use a different algorithm.

MOTIVATION FOR RESEARCH

Sharing resources, software and applications are the important functions of cloud computing that are having an objective to reduce the overall cost like the capital cost and operational cost that are attached with them. Moreover, the performance in terms of processing time, execution time, turnaround time and the waiting time should be achieved considerably. So there are various number of measures and numerous technical challenges that has to be addressed like fault tolerance, virtual machine migration, high accessibility, server consolidation and measurability. However central issue is that the load balancing. It's the mechanism of distributing the load among various number of nodes of a distributed system to boost each resource utilization and job time interval whereas additionally avoiding a state of conflict wherever a number of the nodes known as virtual machines are heavily loaded whereas alternative virtual machines are idle or doing little or no work. It additionally ensures that each one of the processor within the system or every node within the network will process the equal amount of load at any instant of time. The term load balancing is completed with the assistance of load balancers wherever every incoming request is redirected and is executed in minimum response time and the output is given to consumer who makes the request. Supported parameters like availability of the machine or current load, the load balancer uses numerous different mechanisms to work out that server ought to handle and forwards the request on to the chosen server. Clients request for the virtual machine and cloud broker handles the client request according to available virtual machine. If the VM is idle, then broker allocate that VM to user for the processing and if VM is not free then incoming requests of the client are send into the waiting state until the resources are free.

1. The current algorithm will work only in the homogeneous cloud system where all the resources are of same configuration.
2. Cloudlets are not assigned to the virtual machine according to their capacities. There may be a scenario where a cloudlet with highest priority is assigned to the machine with lowest capacity in the host.
3. The processing capacity (No of processors / MIPS) is not considered for assigning the VM to a job.
4. Every time before allocation, extra overhead is involved in parsing the table of virtual machines from top to bottom.

PROPOSED 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. The objective is to achieve accuracy, performance of servers and the cloud environment can be maintained. The cloudlets are divided into 3 classes as mentioned below:

- **Cancellable:** These requests can be scheduled at any time after their arrival time. It need not be resumed later. Cancellable leases do not guarantee the deadline.
- **Suspendable:** Leases of this type can be suspended at any time but should be resumed later. This type of lease guarantees the execution but not in a specific deadline. Suspendable leases are flexible in start time and can be scheduled at any time after their ready time. In the case of preemption, these leases should be rescheduled to find another free time-slot for the remainder of their execution.
- **Non-Preemptable:** The leases associated with such requests cannot be preempted at all.

STEPS

- Open the Cloud Sim Simulator in NetBeans IDE of Java and create the heterogeneous virtual machines of different MIPS and processing power in multiple data centers with different hosts.
- The virtual machines are having service level packages with different resources inside it.
- Create the multiple number of user cloudlets of different packages and size with different types of leases.
- The Datacenter broker inside the cloud simulator will retrieve the number of cloudlets and number of virtual machines created by the user.
- The DCB will group the Cloudlets according to VM package and sort the cloudlets in ascending order of deadline and will also group the VM's that have similar packages and sort the VM's in descending order of processing power.
- The broker will maintain the LOAD distribution table in which the current status, current package, VM Capacity and the number of cloudlets submitted to each and every Virtual machine will be stored.
- Based on the deadline, Cloudlets are allocated to VM's according to package. The cloudlet with minimum deadline (highest priority) is assigned to a virtual machine with highest MIPS.
- If the virtual machine is not available, then the algorithm will find the low priority job according to its lease type. The low priority job of particular lease type will be suspended and will be resumed later on.
- Calculate the different QOS Parameters like waiting time, execution time, turnaround time, processing cost etc.
- Repeat the same procedure for all the remaining cloudlets.

In this strategy, current system state plays major role while making decisions. Despite the fact that dynamic load balancing has higher run rime complexity than 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.

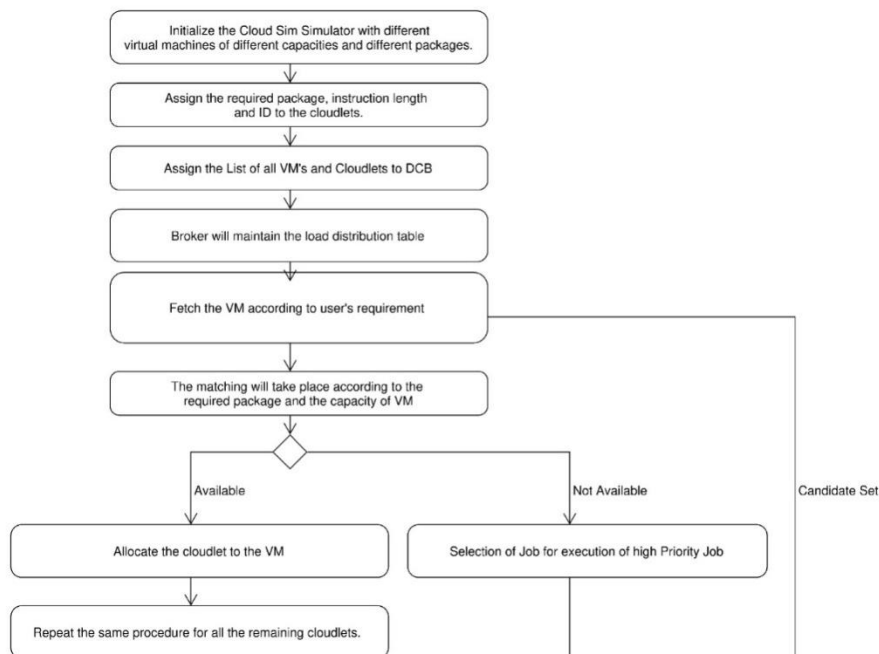


Figure 4. Flowchart of Proposed Methodology

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.

SIMULATION RESULTS

In the table 1, we have listed the total number of cloudlets taken in each experiment. Multiple number of experiments have been conducted inside cloudsims simulator in java and every time number of cloudlets i.e. load on the cloud has been increased. The cloudlets are further classified into 3 parts: Cancellable cloudlets, suspendable cloudlets and non-premtable cloudlets.

Table 1. Different Types of Cloudlets

Sno.	Total No of Cloudlets	Cancellable Cloudlets	Suspendable Cloudlets	Non Premtable Cloudlets
1	5	2	2	1
2	10	3	4	3
3	40	15	15	10
4	60	20	20	20
5	100	30	35	35
6	150	50	50	50
7	200	70	60	70
8	300	100	100	100
9	400	150	100	150
10	500	180	140	180
11	700	250	250	200
12	1000	300	400	300
13	2000	600	700	700
14	3000	1000	1000	1000
15	5000	1400	2000	1600

The table 2 shows the results of the existing work. The processing time, waiting time and the processing cost has been computed for the evaluation of results. The processing time and waiting time is mentioned in seconds and the processing cost is calculated in rupees.

Table 2. Results of Existing Work

Sno.	Total No of Cloudlets	Total Processing Time (Seconds)	Total Waiting Time (seconds)	Total Processing Cost (Rupees)
1	5	13	8	38
2	10	25	22	75
3	40	92	296	281
4	60	135	654	412
5	100	227	1918	693
6	150	338	4333	1030
7	200	452	7836	1379
8	300	675	17698	2059
9	400	902	31842	2752
10	500	1127	49817	3438
11	700	1577	97821	4811



12	1000	2252	200307	6871
13	2000	4502	802365	13736
14	3000	6750	1804998	20594
15	5000	11252	5027268	34330

In the table 3, 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 3. Results of Proposed Work

Sno.	No of Cloudlets	Total Processing Time	Total Waiting Time	Total Processing Cost
1	5	6	0	18
2	10	11	4	35
3	40	56	151	170
4	60	80	354	244
5	100	138	1065	421
6	150	205	2427	625
7	200	273	4342	833
8	300	410	10009	1250
9	400	551	18008	1680
10	500	686	28036	2092
11	700	963	55577	2938
12	1000	1376	113771	4197
13	2000	2748	455878	8384
14	3000	4122	1028186	12577
15	5000	6873	2858431	20970

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

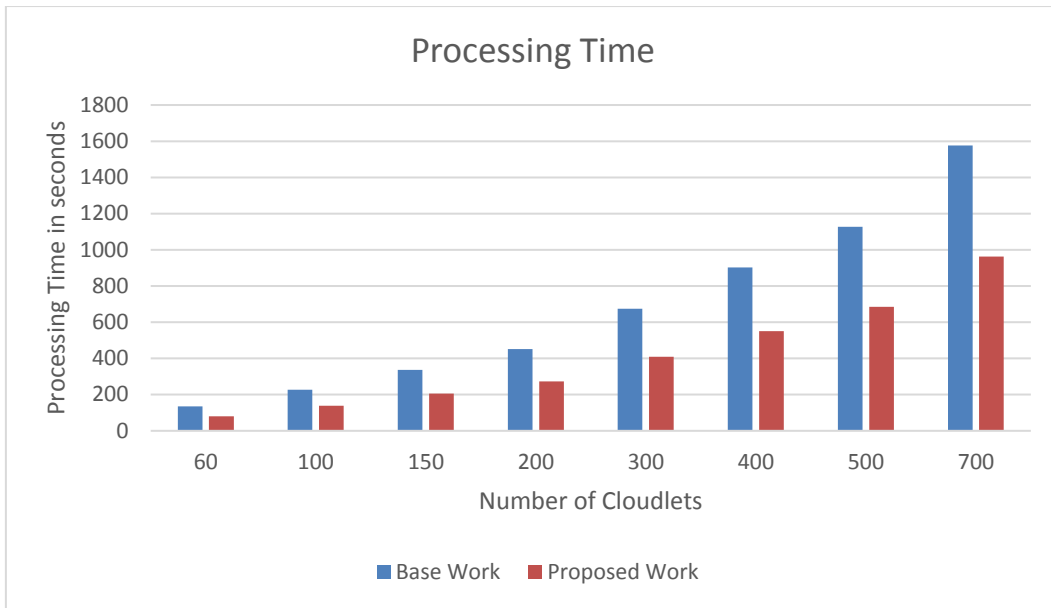


Figure 5. Execution Time of Proposed Algorithm

From the above bar chart in figure 5, it is clear that the processing time has been reduced.

PROCESSING COST

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

$$\text{Processing Cost} = \text{RT} * \text{unit_cost.}$$

where, RT = response time

unit_cost = cost per unit time

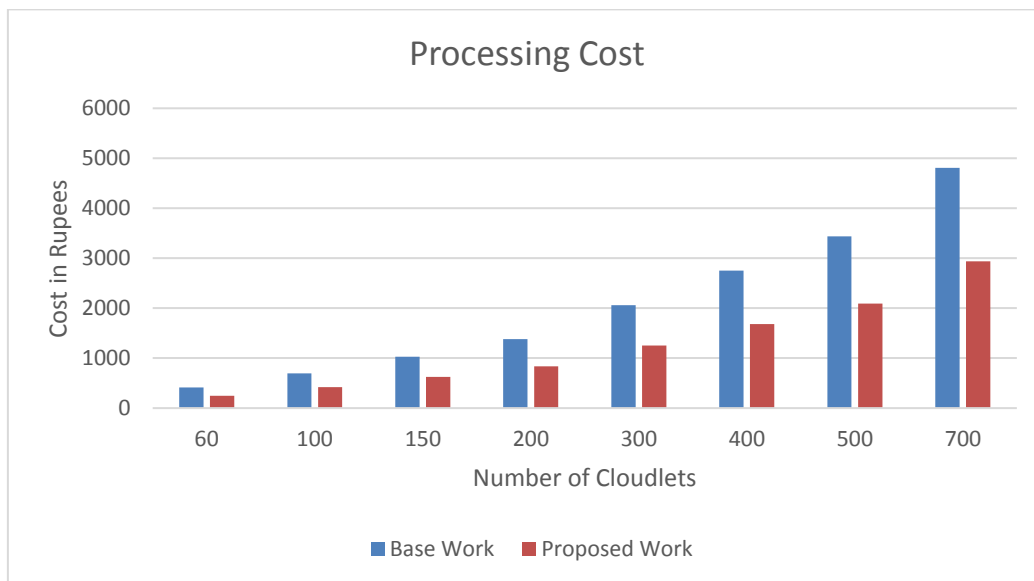


Figure 6. Processing Cost

From the above bar chart in figure 6, it is clear that the processing cost has been reduced. Figure 7 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.

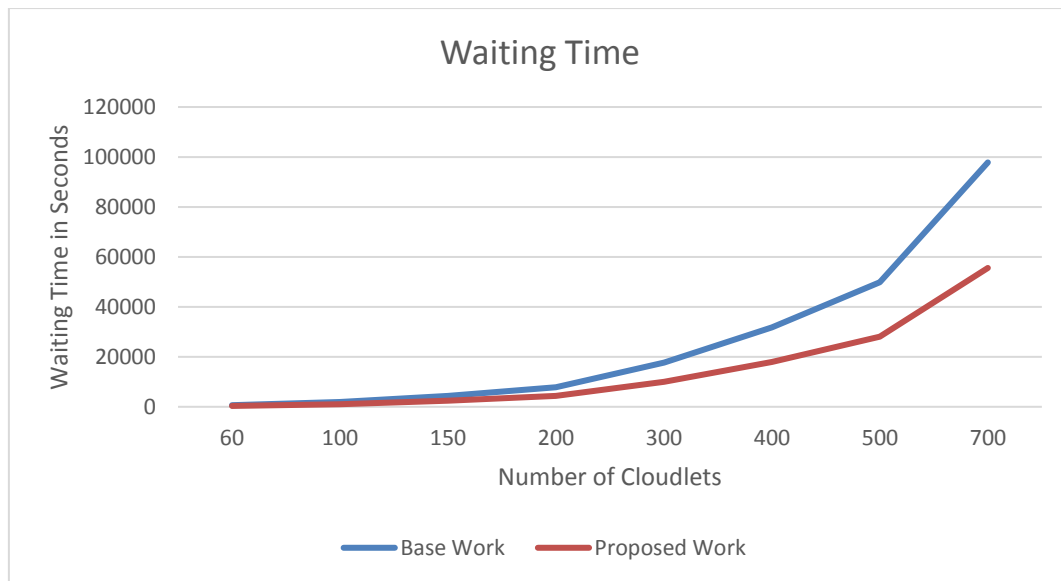


Figure 7. Waiting Time

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 planned load balancer shows wisely that after binding the cloudlets to virtual machines for execution the execution time and waiting time have been reduced so that the required objective is achieved. The model planned is predicated on centralized load equalization strategy. This study is involved with the capacity of the resources or the virtual machines and the task length or instruction size of the cloudlets. By making use of these parameters, we can divide our cloudlets and virtual machine into different classes. Before binding cloudlets to VM, load balancer initially calculates the remaining capacities of all VMs so dispatch the cloudlet to the most powerful VM. Moreover, as load is initial readjusted among VMs, the load distribution method is honest. So as to evaluate performance of the planned model, simulation study has been place through numerous check conditions. It's been found that the model works well in distributing the entire workload or the multiple number of cloudlets to the virtual machines. 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] Al-Rayis, Ektemal, "Performance Analysis of load balancing architectures in cloud computing", IEEE, European Modeling Symposium, pp. 520-524, ISBN 978-1-4799-2577-3, November 2013.
- [2] BhoiUendraBhoi, "Enhanced max-min Task scheduling Algorithm in cloud computing", International Journal of Application or Innovation in Engineering & Management (IJAEM), Vol. 2, Issue 4, ISSN 2319 - 4847, April 2013.
- [3] Bhadani A. and Chaudhary S., "Performance evaluation of web servers using central load balancing policy over virtual machines on cloud", COMPUTE '10, Proceedings of the Third Annual ACM Bangalore Conference, Article no. 16, ISBN 978-1-4503-0001-8, 2010.
- [4] Bendiab A., Randles M., Lamb D., "A comparative study into distributed load balancing algorithms for cloud computing", IEEE, 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 551-556, April 2010.
- [5] M. Birattari and T. Stutzle, "Ant Colony Optimization-Artificial Ants as a Computational Intelligence Technique", IEEE, Computational Intelligence Magazine, Vol. 1, Issue 4, pp. 28-39, November 2006.
- [6] Buzato L.E, Nakai A.M., Madeira E, 5th Latin-American Symposium on Dependable Computing, pp. 156-165, 2011.
- [7] Babu, "Honey bee behavior inspired load balancing of tasks in cloud computing environments", Elsevier, Applied Soft Computing Journal, ISSN : 1568-4946, 2013.



- [8] M. Dorigo, and T. Stutzle, "Ant Colony Optimization-Artificial Ants as a Computational Intelligence Technique", IEEE, Computational Intelligence Magazine, Vol. 1, Issue 4, pp. 28-39, November 2006.
- [9] H. Deldari and M. Salehi , "Grid Load Balancing using an Echo System of Intelligent Ants", Proceedings of the 24th IASTED International Conference on Parallel and Distributed Computing and Networks, pp. 47-52, 2006.
- [10] Desai, "A survey of various load balancing techniques and challenges in cloud computing", International Journals of scientific and technology research, Vol. 2, Issue11, ISSN 2277-8616, November 2013.
- [11] O.M. Elzeki, "Improved Max-Min Algorithm in Cloud Computing", International Journal of Computer Applications, Vol. 50, No. 12, ISSN 0975-8887, July 2012.
- [12] Fahringer T, Nae V., Prodan R., " Efficient management of data center resources for massively multiplayer online games" ACM/IEEE conference on Supercomputing, Article no. 10, ISBN: 978-1-4244-2835-9, 2008.
- [13] Fang Y., Wang F. and Ge J., "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Springer Berlin Heidelberg, Proceedings of International Conference, pp. 271-277, ISBN 978-3-642-16514-6, October 2010.
- [14] Gellerb A ,Lua Y., Xiea Q., Kliotb G., Larusb J. R. and Greenber A., "Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services", International Journal on Performance evaluation, 2011.
- [15] F.M. Guo ,J. Sun, S. Xiong and, "A New Pheromone Updating Strategy In Ant Colony Optimization", IEEE, Proceedings of the International Conference on Machine Learning and Cybernetics, Vol. 1, pp. 620-625. ISBN 0-7803-8403-2, August 2004.
- [16] Hu J, Gu J, Sun G and Zhao T, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", IEEE, Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), pp. 89-96, ISBN 978-1-4244-9482-8, December 2010.
- [17] Liu H., Liu S., Meng X., Yang C. and Zhang Y., "LBVS: A Load Balancing Strategy for Virtual Storage", IEEE, International Conference on Service Sciences (ICSS), pp. 257-262, ISBN 978-0-7695-4017-7, May 2010.
- [18] Lamb D. Randles M. and Taleb-Bendiab A., "A comparative study into distributed load balancing algorithms for cloud computing", IEEE, 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 551-556, April 2010.
- [19] Lua Y., Xiea Q., Kliotb G., Gellerb A., Larusb J. R. and Greenber A., "Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services", International Journal on Performance evaluation, 2011.
- [20] P. Lin, R. Chang, J. Chang, "An ant algorithm for balanced job scheduling in grids", Elsevier, Future Generation Computer Systems, pp. 20-27, June 2008.
- [21] Y. Li, "A Bio-inspired Adaptive Job Scheduling Mechanism on a Computational Grid", International Journal of Computer Science and Network Security, Vol. 6, No. 3, March 2006.
- [22] Mohapatra A., Singh, M. Korupolu, , "Server-storage virtualization: integration and load balancing in data centers", International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1-12, ISBN 978-1-4244-2834-2, November 2008.
- [23] Mehta H., Kanungo P. and Chandwani M., International Conference Workshop on Emerging Trends in Technology, pp. 370-375, 2011.
- [24] Nae V., Prodan R. and Fahringer T., " Efficient management of data center resources for massively multiplayer online games" ACM/IEEE conference on Supercomputing, Article no. 10, ISBN: 978-1-4244-2835-9, 2008.
- [25] Klaihem Al Nuaimi, "A survey of load balancing in cloud computing challenges and algorithm", IEEE, Second symposium on network cloud computing and applications, pp. 137-142, 2012.
- [26] Pallis G., IEEE Journal of Internet Computing, Vol. 14, No. 5, pp. 70-73, 2010.
- [27] Randles M., Lamb D. and Taleb-Bendiab A., "A comparative study into distributed load balancing algorithms for cloud computing", IEEE, 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 551-556, April 2010.
- [28] Ren, "The load balancing Algorithm in cloud computing Environment", IEEE, 2nd International Conference on Computer Science and Network Technology, pp. 925-928, ISBN 978-1-4673-2963-7, December 2012.



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

DOI:10.24297/ijct.v16i5.6235