International Journal of Engineering & Technology, 7 (3.1) (2018) 110-117

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



An Efficient Improved Weighted Round Robin Load Balancing Algorithm in Cloud Computing

Manikandan N[1*], Pravin A[2]

[1]Research Scholar, Sathyabama Institute of Science and Technology, School of Computing, Assistant Professor, St.joseph's College of Engineering, Chennai, Tamilnadu, India.
[2]Assistant Professor, Sathyabama Institute of Science and Technology, School of Computing, Chennai, Tamilnadu, India.
**Corresponding author E-mail: macs2005ciet@gmail.com*

Abstract

Cloud Computing is an era when various trends showed up. It is the type of development, based solely on the Internet. The computer technology has brought in several changes along the years. SaaS is an architecture that comes under computing technology, involving the use of powerful processors, that helps in transforming data centers, scaling services involving computing technology across all fields. Various trends are opening in the field of Cloud Computing, developed mainly on Internet basis and use of computing technology. The computing architecture of SaaS along with powerful processors, are transforming data centers into a huge scale of pools of computing services. Balancing load efficiently is one of the key areas involved in cloud computing. Services that are of very high quality can be subscribed for lower cost, along with completely reliable network connections with their bandwidth being increased , providing a flexible service residing in remote data centers. Load Balancing is the concept involved in the storage of computing resources. It is the partition of work which requires processing between two or more number of computers and/or CPUs, storage devices which ultimately leads to providing higher efficiency with much faster service. The improved WRR algorithm which is improved has been proposed in order to distribute the workload efficiently, thereby enhancing various other features that are associated with it.

Keywords: *Cloud, Load Balancing, Data Centres, Improved Weighted Round Robin and Hybrid Cloud.*

1. Introduction

The complexities involved in direct hardware management are taken care by moving data into the cloud, thereby making it convenient for the users. The pioneer of Cloud Computing sellers, Amazon S3 and windows Microsoft Azure are both well known examples. This shift of computing platform has eliminated the need for maintaining data on local machines. They offer enormous storage space and computing resources that can be customized. The users depend on their cloud service providers for the availability and integrity of their information.

2. Statement of the Problem

On the websites today suffer a serious problems which involves crashing of servers. Alternative solutions have already been proposed but an efficient solution has not been proposed which looks into servers that are being overloaded. The services must be checked to guarantee availability. The objectives involved include balancing the load efficiently. This in turn must bring about flexibility, scalability and accessibility. Maximum utilization of resources must be achieved as part of the load balancing algorithm that has been proposed. The time taken to wait and the time involved in executing a particular task must be reduced. The system must be reliable for efficiency. The challenges also include building a fault tolerant system which means a failure in a particular system will not affect the entire performance of the system. The overall system must be developed to imbibe the quality of services. The ultimate aim is to yield a faster response time by perfect distribution of workload, which leads to customer satisfaction. An efficient solution is obtained by looking into the performance of the system. The purpose of a cloud environment is to reduce the costs associated with it. Thus overheads involved in load balancing must not be compromised. Future enhancement in load balancing will be accommodated during the scope for further exploration.

3. Objective of the Study

1. To process the entire outline of cloud computing.
2. To analyse the algorithms that has already been proposed.
3. To furnish an efficient solution based on the assortment of algorithms.

3.1. Cloud Computing - An Overview

Cloud itself is categorized into two, Service and Deployment models. The following gives an understanding on the Service Model.

3.2. The Service Models

Software as a Service (SaaS) simply explains about the customers purchase of an application or service that is hosted on the cloud. For instance, Zoho offers services for developing web applications along with cloud backup. One more prominent example for this is the Microsoft that offers various products making it easily available to the end users and only requires the users to update to

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

make use of additional features. Therefore, SaaS provides the access to consume fully serviced software in fully serviced infrastructure making it easier for the end users to access the document from anywhere and at any time.
Platform as a Service (PaaS) is a platform where the customers hire access to the platforms, allowing them to deploy their software and applications on the cloud. The clients hire infrastructure and the programming tools facilitated by the vendors to develop applications. Google App Engine is one good example of Platform as a sservice.
Although the cloud infrastructure is managed by the vendor, processing, storage, network services and other resources that device on the basis of computation, are hired by the customers and done through Infrastructure as a service (IaaS). Cloud Amazon's leading service provider has been a major provider in IaaS.

3.3. The Deployment Models

Cloud service opened to all is referred as Public cloud . A third party is responsible for providing the entire cloud service and the customer can enjoy the benefits of the cloud without bearing the cost of setting up the cloud. For e.g. Gmail application is built on public cloud such that a user from any part of the world will be allowed to access the application. It also provides flexibility in opening the application from anywhere and at anytime. However, this type of cloud suffers serious disadvantages as it paves way for unauthorized information access on the cloud.
A particular organisation has a specific private cloud. Its cloud infrastructure is facilitated by the organization. Private cloud has an benefits of the fact that the information on the private cloud is specific to a particular organization and it is up to the organization to decide who can access the application and who can modify the applications. Although the privacy of the data is ensured in private cloud the cost of hosting the entire infrastructure is borne by the organization.
Hybrid cloud is referred to as a combination of both public and private cloud.
One of the important aspects that renders to the quality of service is the perspective of efficient load balancing.

3.4. Introduction to Load Balancing

The load balancing refers to a process in which the amount of work done by a single computer is divided among two or more computers which benefits the user to get their task served quicker and every computers that are involved finishes their task in same amount of time.

Load balancing can be implemented in terms of hardware, software, or a combination of both. There are certain Web sites that experience a great deal of traffic. These sites make using of load balancer algorithms to serve their purpose. These algorithms balance the load, so that no site suffers delay during searches.

Multiple servers are involved during the load balancing process. The same process can be explained in simple terms such that, when two servers balances the work load, then a third server also comes into account to determine which server the work has to be assigned. This is generally done by programming them with various scheduling algorithms for efficient workload assignment. Serving many of parallel requests from users or clients and restore the results for their searches in a faster and reliable manner is the prime motive of every modernized high-traffic website. Scaling a very high volume of searches cost-effectively, the best practice of modern computing requires the addition of more servers. The servers that are involved are usually combined with failover and backup services. In some other approaches, the servers are said to be distributed over different several geographic locations.

Distribution of workload and balancing it between two or more Cloud servers is done through Load Balancers. Therefore, the infrastructure can be shaped to meet activity spikes, thereby optimizing the allocation of resources that ensure a minimal response time. A load balancer is highly recommended in cases that require guaranteed continuity of services, handling high traffic and facing sudden request spikes. Overloads can be prevented by increasing the number of balanced Cloud servers.

A load balancer is like the traffic cop supervising the servers and routing clients requests across the servers that are made to fulfil requests that maximizes the utilization of speed and capacity. No other servers should be overloaded which may retard the performance and is done by utilizing speed and capacity. When a single server malfunctions, the traffic is diverted to rest of online servers by load balancer. Each time requests are sent from load balancer which starts sending requests to server that newly joins the group. There are various task of load balancer such as, distribution of requests sent by client or network load, more effectively across multiple servers. There must be assurance of higher availability and reliability by sending requests to servers that are connected online. The most important task of load balancer to provide flexibility by adding or removing servers as demand dictates.

Asha, M. L. & Neethu Myshri, R. (2014), in paper on "Performance Evaluation of Round Robin Algorithm in Cloud Environment" has underlined the performance of RR algorithm in cloud based environment. A novel methodology of using a cloud analyst toolkit has been selected for the purpose of simulation and to understand the response of cloud computing and its deployment models. The algorithm that has been proposed is quite similar to that of the Throttled algorithm. The balancing is done by giving the requests to every server in a round-robin manner. Round Robin is one of the traditional load balancing scheduling algorithms, used to distribute the workload among servers. This algorithm works best for clusters consisting of servers with identical specifications. It chooses a node randomly and allocates the task in a circular fashion. Even though, it is the simplest method for distributing the request sent by the clients across a group of servers, round robin algorithm suffers non- uniformity in distributing the workload due to the specifications of the servers being identical. In addition, Round Robin algorithm will not take into account of the resource capabilities, priorities, and the size of the tasks. So, the higher priority and the lengthy tasks end up with the higher response times sometimes even leading to overloading of servers.

Priyanka Singh, Palak Baaga & Saurabh Gupta (2016), in the study on Assorted Load Balancing Algorithms in Cloud Computing: A Survey, has given a detailed analysis in the various algorithms that were proposed earlier. An exploration into the assorted algorithms has been made to find a proper solution for load balancing in cloud computing environment and compare those algorithms based on various parameters. The merits and demerits of the algorithms are discussed. One such algorithm is the Minimum Connections Algorithm. This algorithm considers the number of current connections every server will have. When a client tries to connect, the load balancer will eventually try to determine which server will have the minimum number of connections and then assign the newer connections to that server. Since the connection held by each server is taken into consideration, which is referred to as the Least Connections Algorithm. It simply avoids the server from getting overloaded.

D. Saranya & L. Sankara Maheswari (2015), Load Balancing Algorithms in Cloud Computing is dealt in this paper: A review of study on the various load balancing algorithms in cloud computing is done thoroughly. A important type of load balancing algorithm is opportunistic load balancing algorithm which does not consider the present workload of VM and normally keeps every node busy. It quickly deals with unexpected tasks and randomly to the current

node, where every task is assigned randomly. This algorithm provides a load balancing schedule but does not produce a good result. The tasks are processed in a slow manner, where the on-going execution time of the node is not calculated.

Gang Liu, Jing Li & Jianchao Xu (2013), in the article on "An Improved Min-Min Algorithm in Cloud Computing" explains that scheduling of users tasks is a very sophisticated process in cloud computing environment. According to the Min-Min algorithm, the parameters that are required to be known in advance for every tasks, waiting in a queue includes both execution time and completion time. The unexecuted tasks with the minimum execution time are first assigned to the processors and the rest are handled afterward. This is done in accordance with the capability to complete the job in a specified completion time. The unexecuted tasks that are already waiting in the queue for a long time might experience starvation. However, the main disadvantage of Min-Min algorithm is that the longer tasks will not be scheduled. Hence in this paper an Improved Min-Min algorithm which takes the features of previous Min-Min algorithm and also takes into consideration three constraints. The three constraints are quality of service, the dynamic priority model and the cost of service. The conclusion is given by comparison with the traditional min-min algorithm where the experimental results show it can make long tasks execute at reasonable time, increase resource utilization rate and meet users' requirements.

O.M. Elzeki, M.Z.Reshad & M.A.Elsoud (2012), in the study on "Improved Max-Min Algorithm in Cloud Computing" have put forth an Improved Max-Min Algorithm for the purpose of clearing all the limitations of the earlier version of Max-Min Algorithm. The general Max-Min algorithm deals with assigning tasks with longer execution time to the best servers first, along with which the rest of the smaller tasks to be executed are assigned to all other servers. This ensures concurrent execution of tasks and does not suffer starvation, thus serving with minimum completion time improving make span. For example, task A from the meta-task is said to be assigned to the best server since it takes longer execution time than the others whilst task B and C comparatively having lower execution time than task A and is given to the other servers. Thus, the smaller tasks B and C are executed concurrently along with task A. The proposed algorithm in this paper has brought in a unique variation of Max-Min algorithm. The algorithm is basis of step wise study of impact of RASA algorithm in task scheduling and atom concept of Max-Min strategy. A higher and newer version of Max-Min algorithm is put forward to out power scheduling map, which might be similar to RASA map in total time for submitted jobs. In alternate to complete time as in selection basis, improved MAX-Min is based on execution time. Experimental outcomes show accessibility of load balance in small cloud computing environment and total small make span in large-scale distributed systems and cloud computing. Using improved Min-Max algorithm in cloud computing turns scheduling tasks with comparable lower make span rather tha RAS and original Max-Min.

3.5. Tools Used

Eclipse framework: Eclipse is a popular tool for the Java Integrated Development Framework. It is a platform which is designed from the scratch to construct a integrated web and application development tools. Eclipse supports faster development of integrated features on the basis of plug-in model.

It supports a wide range of platforms and can be accustomed to run on any platform.

The architecture exists at the core of the Eclipse software for dynamic discovery, loading, and running of plug-ins. The platform is also responsible for the logistics of running the code. The User Interface of the Eclipse also supports a standard user navigation model. The plug-in ensures that each tasks that are performed are the best.

An open architecture is traced by the eclipse such that all plug-in development mainly focuses on individual area of expertise. Its design supports the backend development by repository expert and built ting end user tools by usability expert substantial newer features and integration levels are added without affecting other tools. A newer model of common workbench is used by the eclipse platform to compose tools from end users point of view in which substantial newer features and integration levels can be added.

The Eclipse platform uses a newer model of a common workbench to compose the tools from the end user's point of view. The plug-ins is provided by using the extension points.

This tool is built among each lower level of the plug-in represents the extensions of plug-in which again defines their own extensions for customization. A wide range of functionality is supported by capability of extension for developers. Elements like files and data are controlled by common platform resource model. Working with tools and providing management of integrated resources in which the eclipse gives user an evident idea to create with plug-ins. this architecture benefits the plug-in developers. Complexity of various runtime environments like different operating systems is handled by this platform.

Eclipse gives the users an evident idea to work with the tools, and provides management of integrated resources which they create with plug-ins.

Plug-in developers are also benefited from this architecture. The platform handles the complexity of different runtime environments, such as different operating systems. This helps the plug-in developers to focus on a specific task without bothering about the integrated issues.

Eclipse Requirements: The Eclipse requires the Java Runtime Environment and the Java Development kit. The Java Runtime Environment is needed to run the java programs and the Java Development Kit is a software bundle that can be used to develop the java application.

The Proposed Algorithm: The proposed algorithm is the Improved Weighted Round Robin algorithm. The algorithm takes its origin from the Weighted Round Robin algorithm. The Weighted Round Robin algorithm works by assigning the maximum weight to the server with the best specifications. For e.g. if there are two servers one with the quad core processors and increased processing speed and the other server with dual core processors and comparatively minimum processing speed then the server with the best specification is assigned the maximum weight. In short the Weighted Round Robin algorithm is a special case of the Round Robin algorithm that works well with servers of different specifications.

The Improved Weighted Round Robin algorithm works such that in addition to considering the specification of the servers it also considers the execution time of the tasks such that it assigns the task with the maximum execution time to the server with the maximum weight. This ensures that the load is distributed evenly among the servers thereby minimizing the response time.

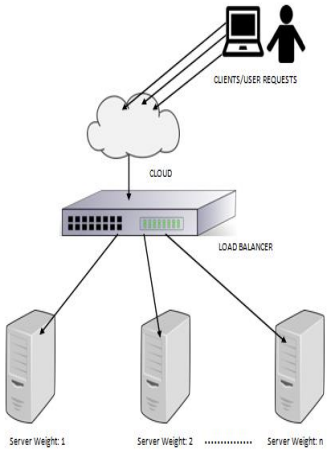


Fig. 1: Architecture diagram of the proposed system

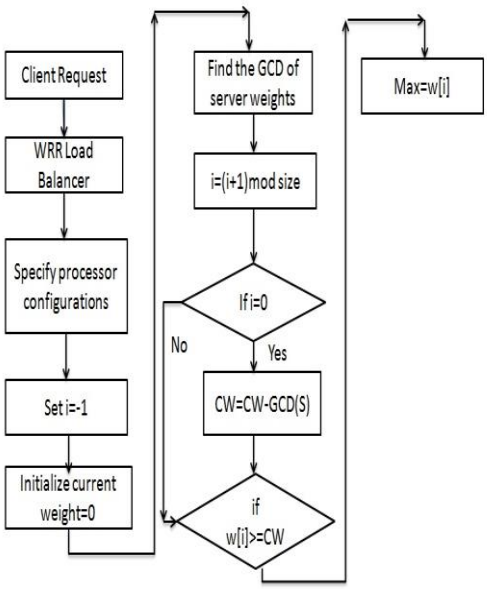


Fig. 2: Data Flow Diagram of the Proposed System
Choosing the appropriate load balancing algorithm

Pseudo Code

Let S = {S1, S2, ……..Sn } be the servers.
Let w = {w1,w2,……..wn } be the weights assigned to the servers.
Let T= {T1, T2,……,Tn}be the tasks arranged in order of the maximum execution time
Let C be the number of tasks assigned to the processors.
{
If w1 = Maximum weight
{
Assign task T1 to server with weight with w1
C++;
If (C > weight w1)
{
Assign task to the next server with maximum weight
}
If (S==NULL)
{
Determine Completion time
}}

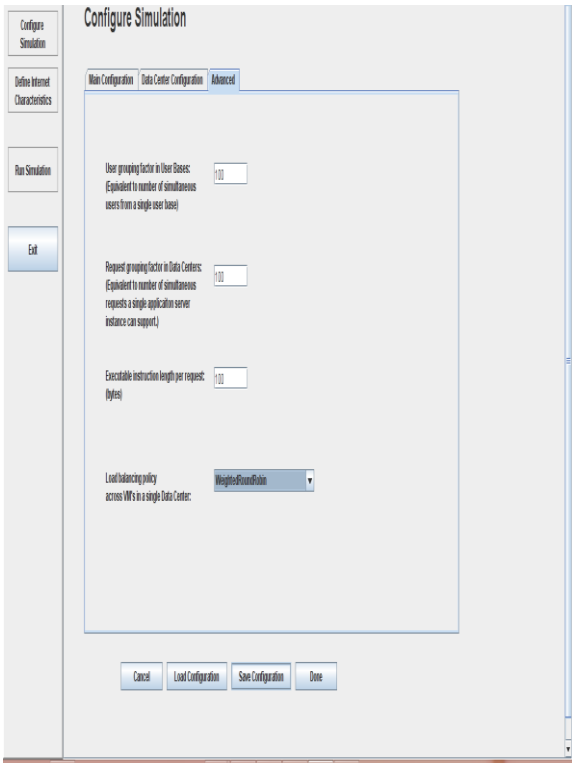


Fig. 3: Load Balancing Policy Application

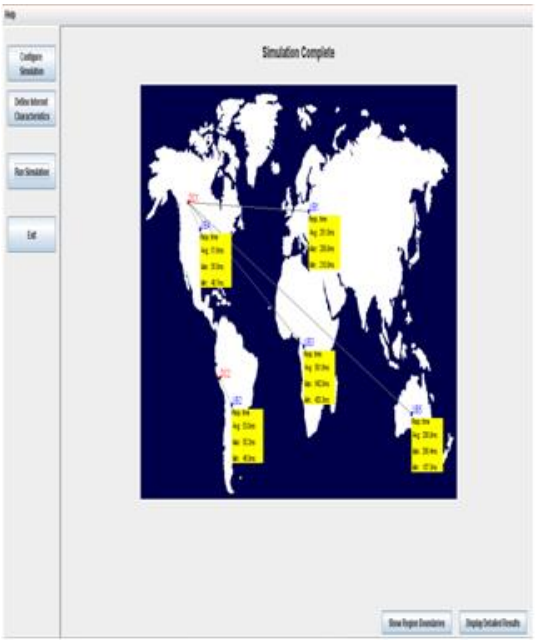


Fig. 4: Running simulation

Calculating the average response time on considering the execution time

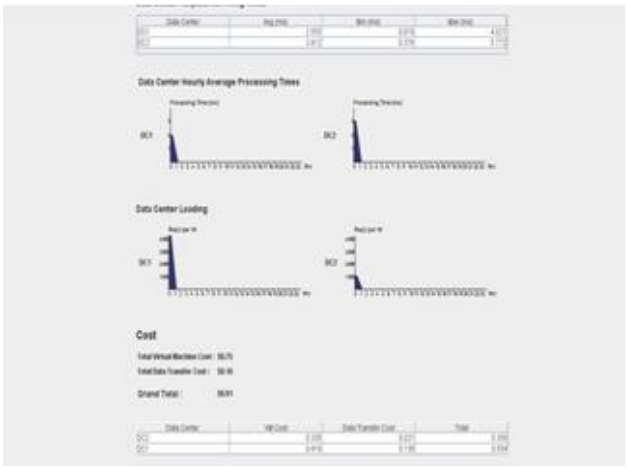


Fig. 5: Response Time Calculation

Comparing with the weighted round robin algorithm without considering the execution time of the tasks.

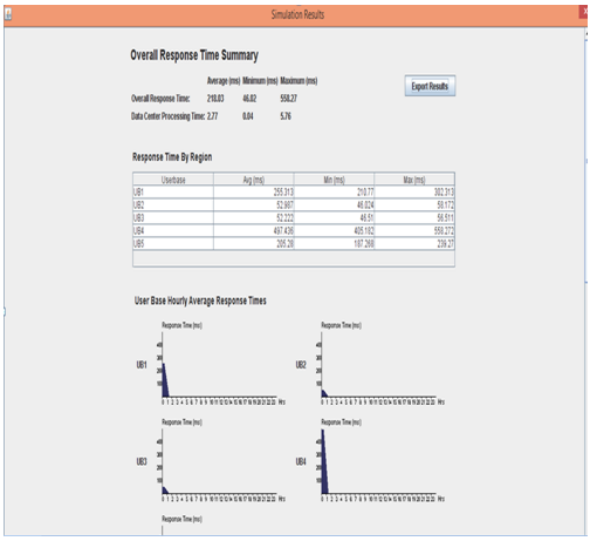


Fig. 6: Results Comparison

The general maximum response time for the particular task is given by,

$$Mi = Ei + Bi + \sum [(Mi + Fk) / Tk]* Ek \quad , \text{for all, } k \in p(i) \qquad (1)$$

$$Ri = Mi + Fimax \qquad (2)$$

Where i is the task given,Ri is the response time Fi is the jitter factor Bi is the blocking factor
Ti is the time taken
Ei is the execution time
p(i) is the task with highest priority task than i
To check if the response time is less than deadline or not
From (1) & (2)

$$Ri = Ei + \sum [Ri / Tk]* Ek \qquad \text{for all, } k \in p(i) \qquad (3)$$

This equation can be iterated on R for n numbers

$$Rin+1 = Ei + \sum [Rin / Tj] * Ek \quad \text{for all, } k \in p(i) \qquad (4)$$

The jitter factor is given by Fi= Fimax - Fimin
Therefore in practical calculation of response time of the tasks the formula is

$$Min+1 = Ei + Bi + \sum [(Min + Fk) / Tk]* Ek \qquad (5)$$

$$Ri = Mi + Fimax$$

In case of distributed systems the response time slightly changes. It is because in such distributed systems there is a broadcast bus to which the nodes are connected. Each node can access the other one via bus. The main difference in the distributed system is that they are non-preemptive.so the blocking factor will be,

$$Bi = \max Ei \quad \text{for all } Ei \in l(i)$$

Where El is the time of transmission
l(i) is the frames with low priority than i
So for this the new response time will be

$$Min+1 = Ei + Bi + \sum [(Min + Fk + Tb) / Tk] * Ek \qquad (6)$$

Now equation (2) is modified to,

$$Ri = Mi + Fimax + Ei$$

Where, Tb is the time taken to transfer one bit
Weighted Round Robin Algorithm:
It is the scheduling algorithm used in switching networks based on the static weights of the queues.
Let the fraction weight be Θn for each queue. Let us consider that the sum of weight of all queues is equal to1.

$$\sum n=1N \quad \Theta n = 1$$

The weights of the queues are in fraction. So it is necessary to determine number of integer block to be given to each queue. So by multiplying a constant integer C we can obtain integer weight Wn. So the integer weight of nth queue will be,

$$Wn = [\Theta n * C] \qquad (7)$$

If there are N connections, the round robin length L is given by,

$$L = \sum n=1N \quad Wn = C$$

The rate given to the nth connection is,

$$rn = (Wn /L) * r \qquad (8)$$

This gives the latency Ø for the nth connection,

$$Øn = (L – Wn) * (Mn / r) = L(1- \Theta n) * (Mn /r) \qquad (9)$$

Where Mn is the maximum length.
From this equation it is inferred that if Θn decreases the latency will increase for increase in N.
To calculate the delay time,

$$dn < bn + [Qn(t) / rn] + Fn$$

Where, Qn(t) is the queue size
Fn is the worst case fair index
From the above equation the worst case fair index for round robin algorithm is given by,

$$Fn = dn - bn – [Qn(t) / rn] = (L – Wn + 1) * (Mn / r) \qquad (10)$$

As the number of connection increases, the worst case fair index also increases.
As the number of connection increases, the worst case fair index also increases. In this study, five regions are assigned with user base distinctly and each user base contains 1000 users appropriately. As per literature review, Round Robin scheduling gives better results in terms of average, minimum, maximum, response time, data transfer time and cost on comparing other scheduling algorithms. In this proposed work an improved WRR scheduling is used and algorithm produces comparatively good results.

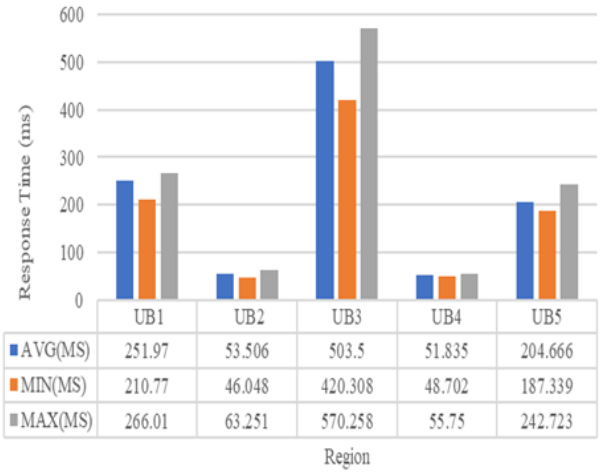


Fig. 7: Overall Response Time of improved WRR

In the above given figure 7, the overall response time of improved WRR in UB1 with categories average, minimum, maximum is 251.97 ms, 210.77 ms and 266.01 ms respectively. With constant change, the end value at UB5 in categories of average, minimum and maximum is 204.666ms ,187.339 ms, 249.723 ms. The X-axis is given in terms of response time (ms) and Y -axis is given in terms of Region.

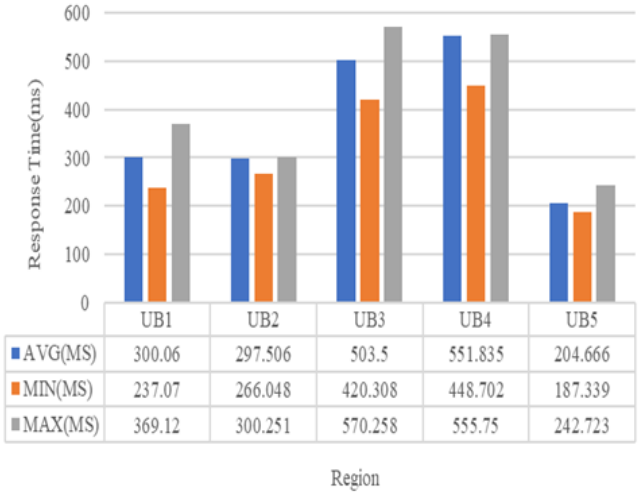


Fig. 8: Overall Response Time in RR

The overall response time in RR, the values in UB1, with categories like average time, minimum and maximum time is 300.06ms, 237.07ms and is 369.12 ms. With regular intervals the result value at UB5 in average time is 204.666 ms and minimum time (ms) is 187.339 and maximum time (ms) is 292.323 ms respectively. Comparing fig(7) improved WRR with response time with RR fig(8), the best overall response time is produced in improved WRR.

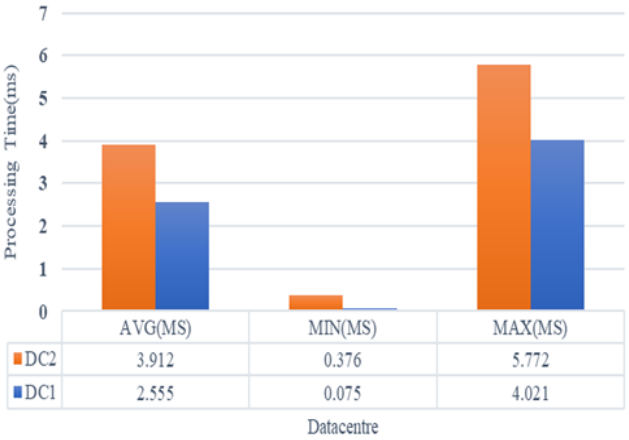


Fig. 9: Data Processing Time of Improved WRR

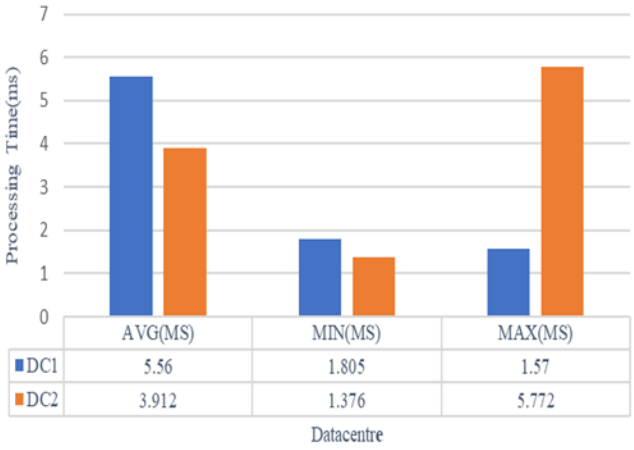


Fig. 10: Data Processing Time of WRR

The above figure 9 explains data processing time of improved WRR in terms of X-axis as data centre and Y-axis as processing time. The values of processing time of DC1 (datacentre) in average, minimum and maximum are 2.555ms, 0.075ms and 4.021ms and the values of DC2(datacentre) are 3.912ms, 0.376 ms and 5.772 ms.

The figure 10 highlights the data processing time in RR with same X -axis as data centre and Y-axis as processing time, the values of DC1 in x-axis average ,minimum and maximum are 5.56 m,1.805,1.57 ms and DC2 values are 3.912 ms,1.376 ms and 5.772 ms. Comparing the improved WRR in Fig.9 with RR in fig.10 the best data transferred time is produced in improved WRR.



Fig. 11: Cost of Improved WRR

The figure 11 features the usage of improved WRR's VM cost, data transfer cost and Total cost. The x-axis data centre, y-axis cost and total cost and the numbers are 0.418 ms,0.36 ms, and 0.422 ms in DC1 and the numbers in DC2 are 0.335,0.021 and 0.35 respectively.

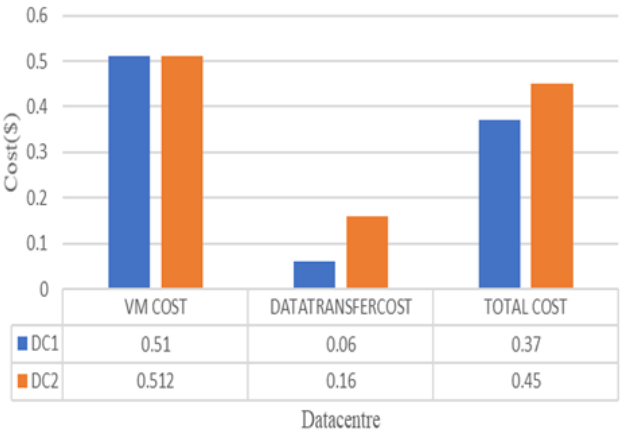


Fig. 12: Cost of WRR

The figure 12 projects the numbers of RR's in cost, data transfer cost and total cost. In data centre1 the values of VM(virtual machine) is 0.51 ms and data transfer cost is 0.06 ms and total cost is 0.37and the values of DC2 with VM cost is 0.512 ms and data transfer cost is 0.16 ms an total cost is 0.45ms.
Comparing the figure 11 with 12 the minimum cost is given in improved WRR and it is explained in the above figures.

4. Conclusion

The paper has mainly focused on the issues of balancing the load efficiently and also minimizing the response time. The algorithm Improved Weighted Round Robin has been comparatively efficient in balancing the load, avoiding the servers being overloaded or under loaded and also minimizing the response time thereby improving performance. This algorithm works by assigning a maximum weight to the server with the best specification and also subsequently calculating the execution time of the tasks such that the task with the maximum execution time goes to the server with the maximum weight. The algorithm also avoids starvation to a greater extent. The Improved Weighted Round Robin algorithm works well on varying load with the requirements already specified.

5. Future Enhancement

The proposed algorithm takes into consideration the processing capacity, load on each Vm, both the size of the task and the execution time of the task into account. The future work can be that in addition to considering these parameters it can also maintain the state of each Vm. The states of the Vm can be flagged as busy or idle. If the state of the Vm is kept as busy then it implies that the server is said to be busy processing a request and will not be able to handle any other request. If the state is flagged as idle then it means that the server is currently not handling any request and is in a state to accept the request.

References

[1] Al Nuaimi, K., Mohamed, N., Al Nuaimi, M., & Al-Jaroodi, J. (2012, December). A survey of load balancing in cloud computing:Challenges and algorithms. In Network Cloud Computing and Applications(NCCA), 2012 Second Symposium on (pp.137-142). IEEE.

[2] Ali, A. D., &Belal, M. A.(2007). Multiple ant colonies optimization for load balancing in distributed systems. Proceedings of ICTA, 2007.

[3] Asha, M. L. & Neethu Myshri,R. (2014), Performance Evaluation of Round Robin Algorithm in Cloud Environment. International Journal of Computer Applications (pp.12-16). ICICT-2014.

[4] Aslam, S., & Shah, M. A. (2015, December). Load balancing algorithm in cloud computing, A survey of modern techniques. In 2015 National Software Engineering Conference (NSEC) (pp. 30-35). IEEE.

[5] Behal, V., & Kumar, A.(2014). Comparative Study of Load Balancing Algorithms in Cloud Environment using CloudAnalyst. International Journal of Computer Applications, 97(1).

[6] Dimri,S. C.(2015). Various Load Balancing Algorithms in Cloud Environment.

[7] Domanal, S. G., & Reddy, G. R. M. (2014, January). Optimal load balancing in cloud computing by efficient utilization of virtual machines. In2014 Sixth International Conference on Communication System and Networks (COMSNETS) (pp. 1-4). IEEE.

[8] Elzeki, O.M., Reshad, M.Z. & Elsoud, M.A.(2012), Improved Max-Min Algorithm in Cloud Computing.Vol. 50-No 12,July 2012, pp.22-27.

[9] Gang Liu, Jing Li & Jianchao Xu, An Improved Min-Min Algorithm in Cloud Computing.Proceedings of the 2012 International Conference of Modern ComputerScience and Applications (pp.47-50),Springer Berlin Heidelberg.

[10] Gopinath, P. G., & Vasudevan, S. K.(2015). An in-depth analysis and studyof Load balancing techniquesin the cloud computing environment. Procedia Computer Science, 50, 427-432.

[11] Gopinath, P.G., & Vasudevan's. K.(2015). An in-depth analysis and study of Load balancing techniques in the cloud computing environment. Procedia Computer Science, 50, 427-432.

[12] Haidri, R.A., Katti, C.P. and Saxena, P.C., 2014, July. Aload balancing strategy forCloud Computing environment. In Signal Propagation and Computer Technology (ICSPCT), 2014 International Conference on (pp. 636-641). IEEE.

[13] Haryani, N., & Jagli, D. (2014). Dynamic Method for Load Balancing in Cloud Computing. IOSR Journal of Computer Engineering (IOSR-JCE), 16(4), 23-28.

[14] Kashyap,D., & Viradiya, J.(2014). A survey of various load balancing algorithms in cloud computing. International Journal of Scientific and Technology Research, 3 (11), 115-19.

[15] Katyal, M., &Mishra, A.(2014). A comparative study of load balancing algorithms in cloud computing environment. arXiv preprint arXiv:1403.6918.

[16] Kaur, R., & Luthra, P.(2012,December). Load Balancing in Cloud Computing. In Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing, ITC.

[17] Kherani, F. F, & Vania, J (2014). Load Balancing in cloud computing.

[18] Mao, Y., Chen, X. and Li, X., 2014. Max–min task scheduling algorithm for load balance in cloud computing. In Proceedings of International Conference on Computer Science and Information Technology (pp. 457-465). Springer India.

[19] Mishra, R., & Jaiswal, A. (2012). Ant colony optimization: Asolution ofload balancing in cloud. International Journal of Web & Semantic Technology,3(2), 33.

[20] More, M. S. D., & Mohanpurkar, A. Load Balancing Strategy Based On Cloud Partitioning Concept.

[21] Nadaph, A., & Maral, V. Cloud Computing–Partitioning Algorithm And Load Balancing Algorithm.

[22] Nitika,M., Shaveta,M., &Raj,M.G.(2012). Comparative analysis of load balancing algorithms in cloudcomputing. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 1(3), pp-120.

[23] Padhy, R.P.(2011). Load balancing in cloud computing systems (Doctoraldissertation,National Institute of Technology, Rourkela).

[24] Pagar,K.,&Patil, S.A Survey on Load Balancing Techniques in Cloud Computing.

[25] Panwar, R., & Mallick, B. (2015).A Comparative Study of Load Balancing Algorithms in Cloud Computing. InternationalJournal of Computer Applications, 117(24).

[26] Priyanka Singh, Palak Baaga & Saurabh Gupta (2016), Assorted Load Balancing Algorithm in Cloud Computing: A Survey. International Journal of Computer Applications. 143-No 7, June 2016. pp.34-40.

[27] Rajan, R.G., & Jeyakrishnan, V. (2013). A survey on load balancing in cloud computing environments. International Journal of Advanced Research in Computer and Communication Engineering, 2(12), 4726 - 4728.

[28] Raddles, M., Lamb, D., & Taleb-Bendiab, A.(2010, April). A comparative study into distributed load balancing algorithms for cloud computing. In Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on (pp.551-556). IEEE.

[29] Ray, S, & De Sarkar, A. (2012). Execution analysis of load balancing algorithms in cloud computing environment. International Journal on Cloud Computing: Services and Architecture (IJCCSA), 2(5), 1-13.

[30] Sadhu,R.,&Vania, Survey on Various Load Balancing Techniques in Cloud Computing Environment.

[31] Sandhiya, S., Saranya, D., Archana, S., & Jayasudha, M. Achieving Mutual Trust and Empowering Dynamic Data in Cloud Storage.

[32] Santra, S. and Mali, K., 2015, September. A new approach to survey on load balancing in VM in cloud computing: Using CloudSim. In Computer, Communication and Control (IC4), 2015 International Conference on (pp. 1-5). IEEE.

[33] Saranya, D.& Sankara Maheswari, L.(2015),Load Balancing Algorithms in Cloud Computing: A Review. International Journal of Advanced Research in Computer Science and Software Engineering 5(7),July-2015,pp. 1107-1111.

[34] Sarddar, D., Bose, R., & Sahana, S. (2015). An Enhanced Cloud Network Load Balancing Approach Using Hierarchical Search Optimization Technique.International Journal of Hybrid Information Technology, 8(3), 9-20.

[35] Sethi, S.,Sahu, A., & Jena, S. K.(2012).Efficient load balancing in cloud computing using fuzzy logic. IOSR Journal of Engineering, 2(7),65-71.

[36] Shah,N.,&Farik, M.Static Load Balancing Algorithms In Cloud Computing: Challenges & Solutions.

[37] Shoja, H., Nahid, H. and Azizi, R., 2014, July. A comparative survey on load balancing algorithms in cloud computing. In Computing, Communication and Networking Technologies (ICCCNT),2014 International Conference on (pp. 1-5). IEEE

[38] M.Joseph and P.Pandya,Finding response time in real time system, the computer journal,29(5):390-396, 1986.

[39] Zuber Patel and Upena Dalal, Design and Implementation of low latency weighted round robin scheduling for high speed networks.

[40] G. Ramprabu, S. Nagarajan, "Design and Analysis of Novel Modified Cross Layer Controller for WMSN", Indian Journal of Science and Technology, Vol 8(5), March 2015, pp.438-444.