iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** IX **Month of publication:** September 2017

DOI: http://doi.org/10.22214/ijraset.2017.9025

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue IX, September 2017- Available at www.ijraset.com

A Load Balancing Based Improved Task Scheduling Algorithm in Cloud Computing

Davneet Singh Chawla[1], Dr. Kanwalvir Singh Dhindsa[2]
M.Tech Research Scholar[1], Professor (CSE)[2], Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib, I.K.G. Punjab Technical University, Jalandhar, Punjab

Abstract*: The concept of cloud computing creates new opportunities for Business and IT enterprises to achieve their objectives. Cloud caters to many needs of the organizations like storage, servers and applications. Cloud computing is extremely beneficial as it is a demand and pay per use service. In cloud computing, specifically there are range of jobs that need to be completed with the to be had assets to achieve most fulfilling performance, less cost of processing, short average waiting time, shortest response time, and efficient utilization of resources etc. Normally errands are scheduled by client instructions or prerequisites. Load balancing is one of the major challenges in cloud computing which is required to distribute the dynamic workload across more than one nodes to make sure that no single node is overloaded. It helps in optimum utilization of resources and hence in enhancing the performance of the system. There are few existing scheduling algorithms which could keep load balancing and provide better methods via efficient process scheduling and resource allocation strategies as well. It becomes important to efficiently utilize the resources which are available so as to maximize the profits with optimized scheduling and load balancing algorithms. This paper discusses some of the scheduling and load balancing algorithms in cloud computing and results have been derived thereafter.

***Keywords*: Cloud Computing, Job Scheduling, Load Balancing, Efficiency, Performance, Cost**

I. INTRODUCTION

The most recent advancements in cloud computing are a rising strategy nowadays is cloud computing. As of late it is discovered that investigators have interest in utilizing cloud for carrying out technical applications and also the enormous associations are on the edge of changing over to hybrid cloud. Numerous applications which are very complex need parallel processing for executing the jobs efficiently. Because of the synchronization and communication among processes which run parallel, there is a reduction in usage of resources of CPU. It is fundamental for a data center to accomplish the use of hubs while keeping up the level of responsiveness of jobs which are running parallel. The cloud computing is pulling in an expanded number of uses to keep running in the data centers which are remote. Numerous intricate applications necessitate capabilities of parallel processing. A portion of the applications which are running parallel demonstrate a decline in usage of resources of CPU at whatever point there is an expansion in parallelism only when there is no planning of jobs accurately then it lessens the execution of a computer.

In Cloud Computing, scheduling plays a vital role in efficiently managing the computer services; It is the succession of captivating decisions about the allocation of available resources and /or capacity to jobs and /or customers well on time. Thousands of users share cloud services by submitting their thousands of computing tasks to the cloud computing environment [1]. Scheduling of these thousands of tasks is a clash or crisis to the cloud environment. The scheduling crisis in cloud has been found out to be difficult to work out, in the case of massive composite jobs like workflows. At the equal time, the scheduling techniques consciousness on throughput, performance, area, cost of time and improve the class of carrier of the complete cloud computing surroundings. Scheduling procedure in cloud is split into three tiers particularly; Resource discovering and filtering, Resource choice, Task allocation. In Resource discovering and filtering the datacenter broking finds out the resources present within the community system and collects repute statistics approximately the assets. In Resource selection, the goal resource is selected based on the requirements of useful resource and objective. This is a decision making stage. In task allocation, the task is allocated to selected resource. Job Scheduling is utilized to assign definite jobs to specific assets specifically time.

The needs of job scheduling in cloud computing are load balance, quality of service, economic principles, best running time, throughput. In environment of cloud computing, issue of scheduling of job is a greatest and testing issue. Subsequently the job scheduler ought to be rapid. The scheduling of job in cloud computing is principally centers to enhance the effective utilization of resource like lessening in completion time, memory and bandwidth. A productive strategy of job scheduling must intend for yielding less time of response in such a way that the execution of jobs submitted happens inside a conceivable least time and hence

there would be a happening of event of in-time where reallocation of jobs is done. Subsequently, few dismissals of jobs happen and additional quantity of jobs could be put forward to the cloud by the customers which at last show expanding results in quickening the business execution of the cloud. [1]

A. Scheduling in Cloud Computing

The needs of job scheduling in cloud computing are load balance, quality of service, economic principles, best running time, throughput. In environment of cloud computing, issue of scheduling of job is a greatest and testing issue. Subsequently the job scheduler ought to be rapid. The scheduling of job in cloud computing is principally centers to enhance the effective utilization of resource like lessening in completion time, memory and bandwidth. A productive strategy of job scheduling must intend for yielding less time of response in such a way that the execution of jobs submitted happens inside a conceivable least time and hence there would be a happening of event of in-time where reallocation of jobs is done. Subsequently, few dismissals of jobs happens and additional quantity of jobs could be put forward to the cloud by the customers which at last show expanding results in quickening the business execution of the cloud. [1]

Job scheduling is the procedure of assigning resources (or machines, such as drills, milling machines, lathes etc.) R= {R1, R2 …, Rm} to jobs (or activities) A= {A1, A2 …, An} over sometime for fulfilling the foreordained goal. For the most part, the allocation may conceivably be compelled by priority imperatives, asset requirements, and issue particular limitations. The most fundamental requirements is priority imperatives, which demonstrate the job or activity has a predefined preparing request through the resources Oi {Oi=oi1, oi2,… , oin}, the established resource limitations is the restricted limit of machine or human asset, and confinements like non-seizure, the hinder of procedure of production are all the issue particular limitations, which are nearly connected with the genuine condition. For the viable environment, perhaps is uncertain, dynamic or complex, the issues of job scheduling are exceptionally different.

With computing systems shifting to cloud-based systems continuously, this system works on a pay-as-you-use basis. Several researches have made an attempt to outline the scheduling problem on cloud systems due to workflow hassle, which is further divided into two ranges: service-level (platform layer and static scheduling) and task-level (unified resource layer and dynamic scheduling). In different forms, the grid computing, the user can install their applications on the virtual machines and decipher a way to execute their programs on the cloud computing machine [2]. For those purposes, despite the fact that both grid computing and cloud computing are heterogeneous, the important troubles they face are very distinctive. A good example is latency of data transfer and the cost on these environments. That is why some studies added more considerations to their definitions of scheduling on cloud.

B. Load Balancing

In cloud computing, Load balancing gives a productive solution for different issues dwelling in usage and set up of cloud computing environment. Load balancing must consider two noteworthy tasks, initial one is the resource allocation or provisioning of resource and other is scheduling of task in environment which is distributed. Effective scheduling of resources and provisioning of resources and in addition tasks will guarantee:

Resources are accessible on demand easily.

Resources are effectively utilized beneath provision of load which is high/low.

Saving of energy is done if there arise of occurrence of low load

Cost of utilizing resources is reduced.

Load balancing is a generally new procedure that encourages resources and networks by furnishing a greatest throughput and having least time of response. Partitioning the traffic in between servers, information could be transmitted and got immediately with no delay. Various algorithms types are accessible that assists traffic loaded in between servers which are accessible. An essential case of load balancing in our day by day life could be identified with websites. Clients could encounter timeouts, delays and conceivable long framework responses with no load balancing. The solutions of load balancing normally apply servers which are repetitive which support a superior distribution of traffic of communication in such a manner that the accessibility of website is decisively settled.

Keeping in mind the end goal for balancing the resources requests it is imperative for recognizing a few major objectives of algorithms of load balancing:

Cost effectiveness

Scalability and flexibility

Priority
Load balancing algorithms are divided into two categories mainly

C. Static

The network traffic is divided evenly among the nodes or servers in static algorithms. Static approach needs a prior knowledge of system sources, so that the decision choice of shifting of the weight does not depend any longer on the current country of the device. Static approach is best suitable for a device having low version in load.

D. Dynamic

The server that's both free or having least load within the entire community or device is found out and fixed up for assigning load in dynamic algorithm. Here modern-day nation of the machine is used to make selections to control the load. But for this actual-time communication with community is required which can elevate the traffic in the system.
The rest of the paper is organized as: Section II describes the literature review. Section III describes the proposed methodology. Section IV presents the algorithms of various procedures for calculating the total credit. Section V shows the Results and Discussion part. Section VI and VII brings the conclusion and future extent of the paper.

II. LITERATURE REVIEW

Ettikyala and Latha [10] have proposed a task scheduling algorithm based on task length and speed of VMs. They have designed a rank based task scheduler which effectively utilizes resources and provides high performance. This algorithm has been tested using CloudSim toolkit with varying lengths of tasks and varying MIPS (speed) of VMs. The results have shown that Rank based task scheduler gives high performance than existing space-shared and time-shared task schedulers.
Tripathy and Patra [8] have discussed the scheduling in cloud computing. Cloud computing is a developing innovation. It process colossal measure of information so mechanism of scheduling fills in as a key part in the cloud computing. Here priority is assigned to the job which gives better execution to the PC and it is attempted best to minimize the switching time and waiting time.
Tareghian and Bornaee [3] have proposed an algorithm to improve the issue of job scheduling in environment of cloud computing. Recent research works on cloud computing have mostly considered one criterion. In this paper a multi-objective scheduling scheme is investigated and a static method for distributing different requests in cloud platform is proposed.
Bey et al. [11] have proposed a task scheduling technique for cloud computing. A new strategy of task scheduling is proposed in view of the aggregate order for allocation of resource to enhance the algorithm of Min-Min. The main concentration is on minimizing the aggregate executing time (makespan) of scheduling of task and expanding the utilization of assets. The results of experiment show that the proposed approach allows more versatile assets distribution for free occupations booking in the distributed computing environment.
Kun-lun et al. [5] have proposed improved algorithm of GEP for scheduling of task in cloud computing. An enhanced algorithm of GEP is proposed with twofold functions of fitness (DF-GEP), which communicates a decent meeting, through experiments contrasted and GA and common GEP algorithm by utilizing the programming model of Map/Reduce.
Vig et al. [4] have proposed a proficient disseminated approach for burden adjusting in distributed computing. This paper displays a few strategies of burden adjusting in various cloud environment, existing burden adjusting strategy furthermore talks about different parameters like adaptability, execution, asset usage, adaptation to non-critical failure, acquainted overhead.
Dave et al. [7] have presented a review on various algorithms of job scheduling in cloud computing. The algorithms of scheduling are executed by taking into account different parameters, for example, resource availability, bandwidth, performance, physical distances, computational time, priority, cost, latency, resource utilization, and throughput.
Singh et al. [9] have given a review on task scheduling in cloud computing. In this paper comparative study of distinctive algorithms is performed for their adaptability, feasibility, suitability with regards to cloud situation, after that hybrid approach is proposed that can be received to upgrade the current stage further.
Zhu et al. [6] have discussed job scheduling for cloud computing integrated with wireless sensor network. The powerful data processing and data storage capabilities of cloud computing and the ubiquitous ability of data gathering of WSN (wireless sensor network) complement one another in integration of CC-WSN, which is drawing in developing enthusiasm from both the industry and academia.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue IX, September 2017- Available at www.ijraset.com

III. PROPOSED METHODOLOGY

There are numerous algorithms for task scheduling available, that are utilized in the environment of cloud computing for better resource scheduling. Scheduling is assigning fundamentally number of resources to the tasks in a manner that there will be maximum utilization of resource, minimum time for total processing and minimum time of waiting. The traditional or any rule-based scheduling algorithms are widely implemented on cloud computing systems because they are simple to understand and easy to implement and also provide most suitable and best solution for scheduling problem. But they all suffer from the problem of not being able to find the suitable and best solution in a reasonable time, for too complex or large problems. Job scheduling is a procedure of mapping tasks of users to the suitable resources and their execution.

In our proposed methodology, we have analysed various low level algorithms in cloud computing like FIFO, SJF and analyse their processing times. Then develop an improved scheduling algorithm using QOS parameters for virtual machine. In this, tasks are scheduled using the priorities which are calculated using weights assigned to each task and virtual machines are sorted using MIPS. After that, the tasks are mapped to virtual machines using some grouping factor in order to optimize the processing and average waiting time.

A. Proposed Algorithm

1) Initialize the Cloudsim package by creating the datacenter, broker, virtual machines and cloudlets
2) Initialize the virtual machines list.
3) Initialize the tasks list.
4) Sort the virtual machines using QOS parameters (MIPS and Granularity size).
5) Sort the task list using priorities calculated using weights by using following procedure:
6) In this credit to task is assigned using 3 parameters which are weights based on task length difference, priority of the task, deadline of the task.

$$Totalcredit_i = Credit\_Length_i * Credit\_Priority_i * Credit\_deadline_i$$

7) Assigning the tasks to the virtual machines in groups. Like if there are 30 tasks and we have 10 virtual machines then assign each virtual machine 3 tasks based on the calculated credit of each task.
8) This process of allocation will be repeated for all tasks.

Load balancing of virtual machines is acquired by firstly mapping tasks to VM's and after that all the VMs to resources which are host, utilizing the method System Load balancing which is task based. This algorithm guarantees the load balancing of system by the means of transferring extra tasks only from a VM which is overloaded rather than migration of the whole overloaded VM. Then, Perform task scheduling which is sending the cloudlet to the first available virtual machine. After then evaluation of performance parameters of Scheduling Strategies which considers processing time, processing cost and waiting time.

IV. ALGORITHMS OF VARIOUS PROCEDURES PERFORMED FOR CALCULATING TOTAL CREDIT

Procedure 1: Credit based on Length of task[12]

For all submitted tasks in the set; Ti
Task length difference (TLD)= absolute value (average length – length of particular task)
If $TLD_i \leq$ value_1
then credit =5
else if value_1 < $TLD_i \leq$ value_2
then credit =4
else if value_2 < $TLD_i \leq$ value_3
then credit =3
else if value_3 < $TLD_i \leq$ value_4
then credit =2
else value_4 > TLD_i
then credit =1
End For
where value_1= high_len / 5;
value_2= high_len / 4;
value_3=value2+value1;
value_4=value3+value2;

Procedure 2: Priority credits assigning to task

For all submitted tasks in the set; Ti
Find out highest priority task
Choose the divisible factor for priority
$credit\_Priority_i = TaskPriority_i / divisiblefactor_i$
End For

Procedure 3: Deadline of the task

For all submitted tasks in the set; Ti
Find out MAXMIPS of the VM from the virtual machine list
$Deadline\_Task_i = Credit\_Length_i * Credit\_Priority_i / MIPS_{MAX}$
End For

Procedure 4: Adaptive Load Balancing based on the deadline time

For all submitted tasks in the set; T_i
Find the execution time of Ti and the deadline time of T_i
Calculate $U_i = E_i / D_i$
If (T_i (U_i)) < threshold
Insert T_i into EDF List
Else
Insert T_i into AEDF List
End If
End For

For all submitted tasks in the EDF List; T_i
Schedule Task using EDF
End For

For all submitted tasks in the AEDF List; T_i
Schedule T_i with Min (Di)
Calculate NT = Ei + ST
If $(T_{(i+1)} E_{(i+1)}) + NT \leq D_{(i+1)}$
ScheduleTi+1
EndIf
End For

The major goal of above ALB algorithm is to optimize the load balancing in cloud environment also this method is intended to utilize the cloud resources efficiently. Thus the concluding results of ALB method show the better performance.

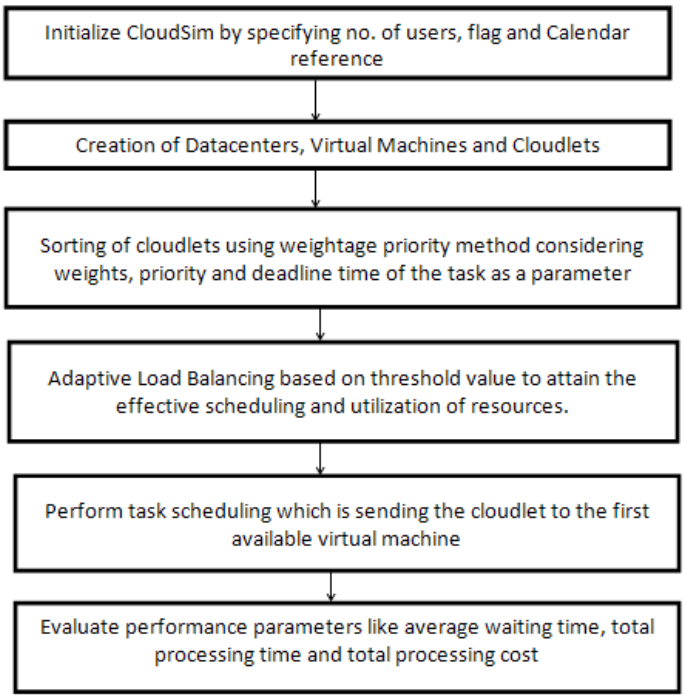


Fig 1: Flowchart of Proposed Algorithm

V. RESULTS AND DISCUSSION

The proposed methodology is implemented with the help of Cloudsim and NetbeansIDE8.0. Cloudsim is the knowledge source that provides the simulation environment of cloud computing and also provides the primary classes describing users, virtual machines, data centres, and applications. In our proposed methodology, various scheduling strategies are there whose performance is evaluated on the basis of following parameters:

Processing Time

$$Processing_{time} = cloudlet_{length}/vm_{MIPS} * no\_ofPES$$

Processing Cost

$$Processing_{cost} = Datacenter_{costpermemory} * VM_{Ram}$$

Waiting Time

$$Waiting_{time} = \sum_{i=1}^{no\ of\ cloudlets} cloudlet_i.waitingtime()$$

A. Creating a scenario in which 30 cloudlets are mapped to 5 virtual machines:

1) *First Come First Served (FCFS) Algorithm for Broker:* First Come First Serve algorithm indicates that the jobs are completed as in line with the order of task arriving time. The FCFS set of rules may additionally further breed the convoy impact which commonly takes location when there may be a job with a massive quantity of workload inside the activity queue. In this state of affairs, all of the jobs which can be queued at the back need to wait a long time for the long activity to finish.

```
total Processing time : 1412.286585655065
total Processing Cost : 824.5799999999998
Average Waiting Time : 0.5430101913810258
```

Fig 2: Results for 30 Cloudlets and 5 Virtual machines using FCFS Broker

30 Cloudlets and 5 Virtual Machines	
Total Processing Time	1412.286
Total Processing Cost	824.579
Average Waiting Time	0.543

Table 1: Results for 30 Cloudlets and 5 Virtual machines using FCFS Broker

2) *Shortest Job First (SJF) Algorithm for Broker:* Shortest Job First algorithm schedules the cloudlet with the least execution time. Highest precedence is assigned to the jobs with minimum execution time and located first in queue while the lowest precedence is assigned to the activity with the most execution time.

```
total Processing time : 1411.7450031778553
total Processing Cost : 824.5800000000004
Average Waiting Time : 0.5393697385558187
```

Fig 3: Results for 30 Cloudlets and 5 Virtual machines using SJF Broker

30 Cloudlets and 5 Virtual Machines	
Total Processing Time	1411.745
Total Processing Cost	824.580
Average Waiting Time	0.539

Table 2: Results for 30 Cloudlets and 5 Virtual machines using SJF Broker

3) *Credit Based Algorithm for Broker:* Credit for each task is calculated based on the task length, task priority and the speed of the VMs. Each task is assigned a credit before the broker starts scheduling the tasks. This leads to efficient scheduling of the tasks.

```
total Processing time : 1403.0331445592267
total Processing Cost : 824.5799999999999
Average Waiting Time : 0.48610655365755606
```

Fig 4: Results for 30 Cloudlets and 5 Virtual machines using Credit Based Algorithm

30 Cloudlets and 5 Virtual Machines	
Total Processing Time	1403.033
Total Processing Cost	824.579
Average Waiting Time	0.486

Table 3: Results for 30 Cloudlets and 5 Virtual machines using Credit Based Algorithm

4) *Credit Based Adaptive Load Balancing Algorithm for Broker:* Credit based scheduling reaches a bottle neck when a lot of tasks have similar credits. It becomes important to load balance the virtual machines for better job allocation. Load balancing helps in taking off the load from virtual machines with a lot of jobs and assigning tasks to other virtual machines that have jobs below the calculated threshold value.

```
total Processing time : 1397.3788446518747
total Processing Cost : 698.8069999999996
Average Waiting Time : 0.435504490651544
```

Fig 5: Results for 30 Cloudlets and 5 Virtual machines using Credit Based Adaptive Load Balancing Algorithm

30 Cloudlets and 5 Virtual Machines	
Total Processing Time	1397.378
Total Processing Cost	698.806
Average Waiting Time	0.435

Table 4: Results for 30 Cloudlets and 5 Virtual machines using Credit Based Adaptive Load Balancing Algorithm

B. Comparison of different algorithms used for broker:

1) Processing Time: The Fig. 6 below shows a comparison of the processing times of the four scheduling techniques used to assign cloudlets to the virtual machines in the simulated environment. As seen from the results, there is a significant difference in the processing times of the conventional FCFS and SJF algorithms and the Credit Based algorithm used for scheduling. Further it is seen that Credit Based Adaptive Load Balancing algorithm has performed much better than the other three algorithms used.

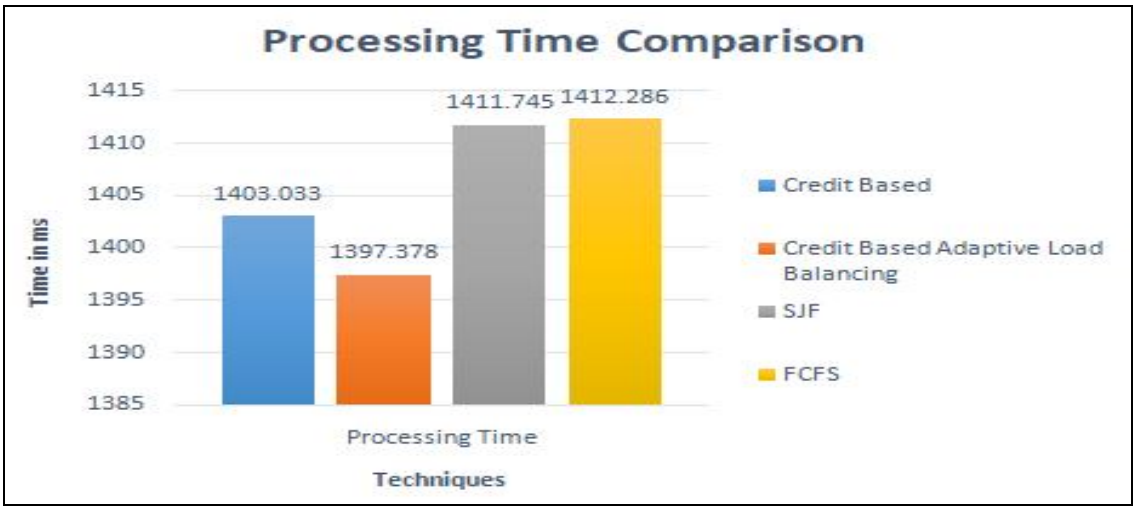


Fig 6: Processing Time Comparison of different algorithms used for the Broker

The SJF algorithm only considers one factor while scheduling which is the length of the jobs while scheduling. This leads to a problem that larger jobs are stuck at the end of the queue and hence this increases the chances that they get assigned to the same VM leading to greater processing time. Similarly FCFS only considers one factor which is the time of arrival of the job while scheduling. This also leads to the problem that similar jobs that arrive together get assigned to the same VM lowering the overall efficiency of the system. Whereas the Credit Based Algorithm considers multiple factors while calculating the credit for each job. This leads to better scheduling of the jobs and hence lower processing time. Sometimes in the Credit Based approach different jobs end up with the same credit. This is overcome by the adaptive load balancing used in Credit Based Adaptive Load Balancing Algorithm and thus it has the lowest processing time of all the four algorithms.

2) Processing Cost: The Fig. 7 below shows a comparison of the processing costs of the four scheduling techniques used to assign cloudlets to the virtual machines in the simulated environment. There is no significant difference in the processing costs of the conventional FCFS and SJF algorithms and the Credit Based algorithm used for scheduling but it is seen that Credit Based Adaptive Load Balancing algorithm has performed considerably well.

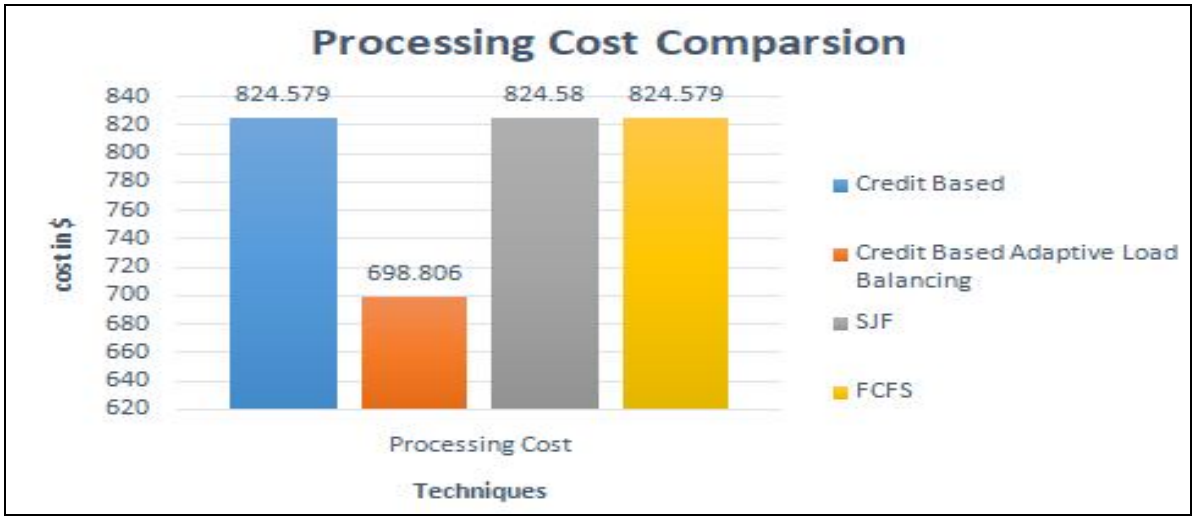


Fig 7: Processing Cost Comparison of different algorithms used for the Broker

The cost of using the resources in SJF, FCFS and Credit Based algorithm is nearly same as the jobs are assigned to the VMs and there is no load balancing performed to optimize the utilization. Since the Credit Based Adaptive Load Balancing algorithm balances the load on all the VMs based on a threshold value, the resources are utilized in an optimized manner and hence the costs are significantly lower.

3) *Average Waiting Time:* The graph below shows a comparison of the average waiting time of the cloudlets using the four scheduling techniques used to assign cloudlets to the virtual machines in the simulated environment. The Credit Based algorithm has performed better than the conventional FCFS and SJF algorithms and it is seen that the average waiting time using the Credit Based Adaptive Load Balancing algorithm is the least among the four algorithms used.

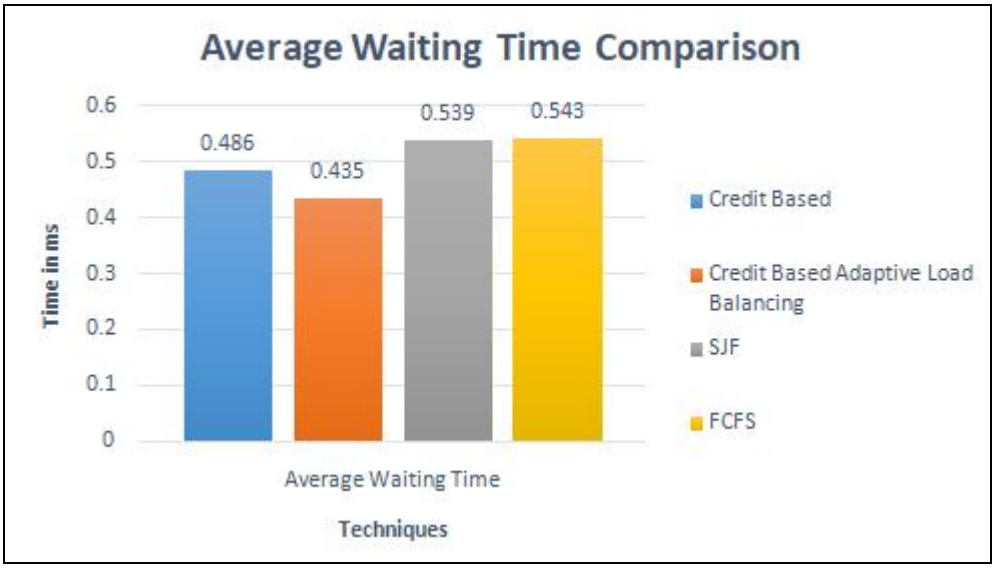


Fig 8: Average Waiting Time Comparison of different algorithms used for the Broker

The average waiting time of the Credit Based algorithm is lower than the SJF and FCFS based broker as multiple factors are considered to calculate the credit assigned to each job. This leads to scheduling of jobs in a manner that the jobs that take nearly same time for execution are scheduled together. This leads to completion of those jobs in similar time span and thus lowering the average waiting time. Similarly the Credit Based Adaptive Load Balancing algorithm performed better as it distributes the dynamic workload across more than one VM such that at any given time all the VMs have equal load based on the given threshold. Due to this more jobs are completed in lesser time and this leads to even lower waiting times.

VI. CONCLUSION

The results of the implementation of Credit Based Adaptive Load Balancing Algorithm on the simulated dataset as interpreted in Chapter 4 has led to some important conclusions. The brief conclusion is as follows:

A. The credit based scheduling that takes into account more than one factor for assigning priority to the jobs performs better than the conventional algorithms like FCFS and SJF used for scheduling jobs.
B. Using Adaptive Loading Balancing with the credit based algorithm further improves the performance of the broker for scheduling the jobs.
C. A credit based scheduling algorithm based on length and priority is proposed considering the user's Quality of Service requirements.
D. In addition to this, the research aims to achieve load balancing on VMs and improved resource utilization.

VII.FUTURE SCOPE

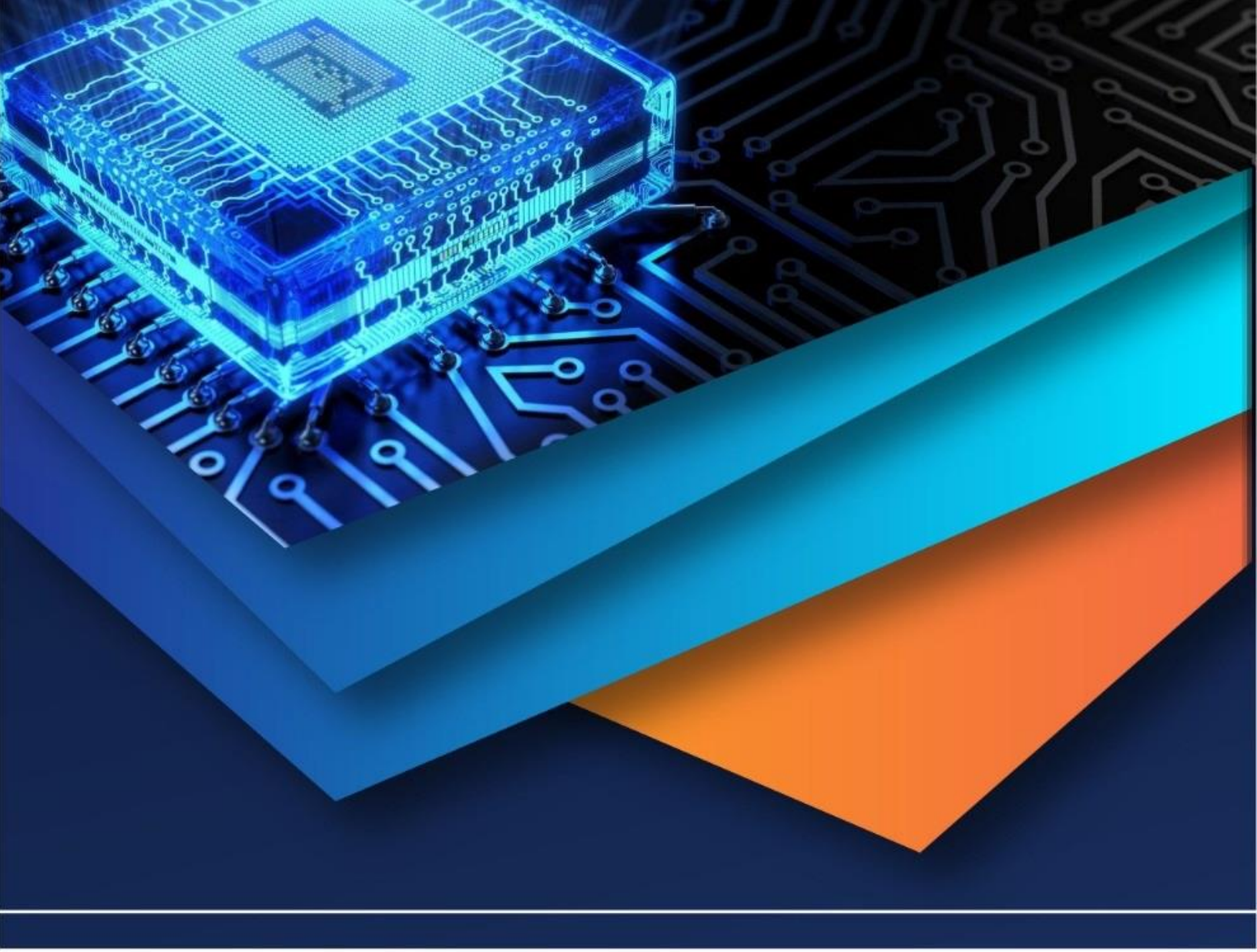
The work can be further extended in future aiming to achieve better performance. The proposed work is using load balancing with credit based scheduling algorithm. In future, the work can be extended as follows:

A. Resource allocation can be performed with evolutionary algorithms.

B. The proposed system has been implemented on simulated data. It can be further implemented in real time scenario.
C. The credit based approach can be replaced with other algorithms like Heuristic Scheduling along with Load Balancing.
D. This work can be extended to study the effects on Grid environment.
E. The proposed system was tested on space shared systems. It can be tested with time shared systems.

REFERENCES

[1] Mell, P. and Grance, T., "The NIST Definition of CloudComputing (Draft)", pp. 1-7, 2011.
[2] Wang, L., Laszewski, G., Kunze, M. and Tao, J., "Cloud Computing: A Perspective study," J. New Generation Computing, pp.1-11, 2010.
[3] Bornaee, Z. and Tareghian, S., "Algorithm to improve job scheduling problem in cloud computing environment," IEEE 2nd International Conference on Knowledge-Based Engineering and Innovation, 2015.
[4] Vig, A., Kushwah, R.S. and Kushwah, S.S., "An Efficient Distributed Approach for Load Balancing in Cloud Computing", International Conference on Computational Intelligence and Communication Networks, IEEE, 2015.
[5] Li, K., Wang, J., Song, J. and Dong, Q., "Improved GEP Algorithm for Task Scheduling in Cloud Computing," Proceedings of 2nd Second International Conference on Advanced Cloud and Big Data, pp. 93–99, 2014.
[6] C. Zhu, Xihua Li, Victor C. M. Leung, X. Hu, and L. T. Yang, "Job Scheduling for Cloud Computing Integrated with Wireless Sensor Network," IEEE 6th International Conference on Cloud Computing Technology and Science, pp. 62–69, 2014.
[7] Dave, Y.P., Shelat, A.S. and Patel, D.S. , "Various job scheduling algorithms in cloud computing: A survey," IEEE International Conference on Information Communication & Embedded Systems, No. 978, 2014.
[8] Tripathy, L. and Patra, R.R. , "Scheduling In Cloud Computing, " International Journal on Cloud Computing: Services and Architecture, Vol. 4, No. 5, pp. 21–27, 2014.
[9] Singh, R.M., Paul, S. and Kumar, A., "Task Scheduling in Cloud Computing : Review," International Journal of Computer Science and Information Technologies, Vol. 5, No. 6, pp. 7940–7944, 2014.
[10] Ettikyala, K. and Latha, Y.V., "Rank Based Efficient Task Scheduler for Cloud Computing," IEEE International Conference on Data Mining and Advanced Computing, pp. 343-346, 2016.
[11] Bey, K.B., Benhammadi, F. and Benaissa R., "Balancing heuristic for independent task scheduling in cloud computing," 12th International Symposium on Programming and Systems, pp. 7–12, 2015.
[12] Thomas, A., Krishnalal, G. and Raj, V.P.J., "Credit based scheduling algorithm in cloud computing environment," Elsevier International Conference on Information and Communication Technologies, Vol. 46, No. 2, pp. 913–920, 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY