Comparative Analysis of Existing Load Balancing Algorithms in Cloud Computing

Shilpa B Kodli[1] & Vinita[2] & Ashwini R S[3]
[1]Assisstant Professor, [2]PG Student, [3]PG Student,
Visvesvaraya Technological University, Department of MCA,
PG Centre Kusnoor road Kalaburagi 585104, Kalaburgi, India.

Received: June 18, 2018 Accepted: August 01, 2018

ABSTRACT *Cloud computing is a novel technology leads several new challenges to all organizations worldwide. Cloud computing supports virtual machines (VMs) to host multiple applications simultaneously. Balancing the large numbers of applications in the heterogeneous cloud environment becomes challenging as the hypervisor scheduling controls all VMs. When the scheduler allocates tasks to the overloaded VMs, the performance of the cloud system degrades. In this paper, we present a novel load balancing approach to organizing the virtualized resources of the data center efficiently. In our approach, the load to a VM scales up and down according to the resource capacity of the VM. The proposed scheme minimizes the make span of the system, maximizes resource utilization and reduces the overall energy consumption. We have evaluated our approach in CloudSim simulation environment, and our devised approach has reduced the waiting time compared to existing approaches and optimized the make span of the cloud data center.*

Keywords: *Cloud Computing, Load Balancing, Makespan, Task Allocation, Virtual Machine, Resource Utilization.*

I. Introduction

In cloud computing users can access resources all the time through internet. It provides online resources and online storage to the users. In Cloud computing cloud provider outsources all the resources to their client.

Cloud computing means storing and accessing data and programs over the internet instead of your computer's hard drive. Cloud computing means on demand delivery of IT resources via the internet with pay-as-you-go pricing. It provides a solution of IT infrastructure in low cost.

Cloud computing has four types of models [1] private, public, hybrid and community cloud.

Private cloud- The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers.

Public cloud- The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or organization, or some combination of them.

Hybrid cloud- The cloud infrastructure is a composition of two or more distinct cloud infrastructure that remains unique entities, but is bound together by standardized technology that enables data and application portability.

Community cloud- The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns.

In [2] the cloud model has three services:

Saas (Software as a service) – is a method for delivering software applications over the internet, on demand and typically on a subscription basis. Saas provides a complete software solution that to you purchase on a pay-as-you-go basis from a cloud service provider.

Paas (Platform as a service) – refers to cloud computing services that supply an on-demand environment for developing, testing, delivering, and managing software applications.

Iaas (Infrastructure as a service) – is the most basic category of cloud computing services. It is an instant computing infrastructure, provisioned and managed over the internet.

In cloud computing, if users are increasing load will also be increased, the increase in the number of users will lead to poor performance in terms of resource usage, if the cloud provider is not configured with any good mechanism for load balancing and also the capacity of cloud servers would not be utilized properly.

II. LOAD BALANCING IN CLOUD COMPUTING

Load balancing is the process of improving the performance of the system by shifting of workload among the processors. Load balancing is a new approach that assists networks and resources by

providing a high throughput and least response time [3]. In cloud platforms, resource allocation (or load balancing) takes place majority at two levels.

- At first level: The load balancer assigns the requested instances to physical computers at the time of uploading an application attempting to balance the computational load of multiple applications across physical computers.
- At second level: When an application receives multiple incoming requests, each of these requests must be assigned to a specific application instance to balance the computational load across a set of instances of the same application.

Based on the current state of the system load balancing is classified as:

- **Static Load Balancing**
  In the static load balancing algorithm the decision of shifting the load does not depend on the current state of the system. It requires knowledge about the applications and resources of the system. The performance of the virtual machines is determined at the time of job arrival. The master processor assigns the workload to other slave processors according to their performance. The assigned work is thus performed by the slave processors and the result is returned to the master processor.
  Static load balancing algorithms are not preemptive and therefore each machine has at least one task assigned for itself. Its aims in minimizing the execution time of the task and limit communication overhead and delays.
- **Dynamic Load Balancing**
  In this type of load balancing algorithms the current state of the system is used to make any decision for load balancing, thus the shifting of the load is depend on the current state of the system. It allows for processes to move from an over utilized machine to an underutilized machine dynamically for faster execution.
  This means that it allows for process preemption which is not supported in Static load balancing approach. An important advantage of this approach is that its decision for balancing the load is based on the current state of the system which helps in improving the overall performance of the system by migrating the load dynamically.

III. METRICS FOR LOAD BALANCING

There are some parameters

1) **Response time: -** The time taken by load balancing algorithm to respond for a particular task. This parameter should be minimized.
2) **Resource utilization: -** It is the degree to which resource is utilized. A good load balancing algorithm should provide maximum resource utilization.
3) **Performance: -** It is the effectiveness of system. For a system to be effective the response time of task be reduced while maintaining acceptable delay. If all above parameters are satisfied then performance also improved.
4) **Scalability: -** It is the ability of system to perform with finite no. of nodes in system.

IV. Literature survey

Load balancing refers to efficiently distributing incoming network traffic across a group of backend servers. Load balancers are used to **increase** capacity and reliability of applications. In computing, **load balancing** improves the distribution of workloads across multiple computing resources. A load balancer is a device that spreads network or application traffic across multiple servers or computers.

There is a growing interest around the utilization of cloud computing in education. As organizations involved in the area typically face severe budget restrictions, there is a need for cost optimization mechanisms that explore unique features of digital learning environments. In[4] authors introduced a method based on Maximum Likelihood Estimation that considers heterogeneity of IT infrastructure in order to devise resource allocation plans that maximize platform utilization for educational environments. They performed experiments using modelled datasets from real digital teaching solutions and obtained cost reductions of up to 30%, compared with conservative resource allocation strategies.

Energy demand in data centre industry is growing rapidly as computing technology changes and Information Technology (IT) professionals seek to maximize performance of data centres. A multitude of methods have been used to estimate and quantify energy intensity. In [5] M. Uddin et.al, discussed

Techniques to implement in green data centres to achieve energy efficiency and reduce global warming effects.

Load management in cloud data centers must take into account 1) hardware diversity of hosts, 2) heterogeneous user requirements, 3) volatile resource usage profiles of virtual machines (VMs), 4) fluctuating load patterns, and 5) energy consumption. In [6] authors proposed distributed problem solving techniques for load management in data centers supported by VM live migration. Collaborative agents are endowed with a load balancing protocol and an energy-aware consolidation protocol to balance and consolidate heterogeneous loads in a distributed manner while reducing energy consumption costs. Agents are provided with 1) policies for deciding when to migrate VMs, 2) a set of heuristics for selecting the VMs to be migrated, 3) a set of host selection heuristics for determining where to migrate VMs, and 4) policies for determining when to turn off/on hosts.

In [7] D. B. LD et.al proposed an algorithm named honey bee behaviour inspired load balancing (HBB-LB), which aims to achieve well balanced load across virtual machines for maximizing the throughput.

In [8] D. Puthal et.al introduces the phenomenon of cloud computing was proposed, there is an unceasing interes for research across the globe. Cloud computing has been seen as unitary of the technology that poses the next-generation computing revolution and rapidly becomes the hottest topic in the field of IT. This fast move towards Cloud computing has fuelled concerns on a fundamental point for the success of information systems, communication, virtualization, data availability and integrity, public auditing, scientific application, and information security. Therefore, cloud computing research has attracted tremendous interest in recent years.

In [9] S. K. Mishra introduces the energy aware model which includes description of physical hosts, virtual machines and service requests (tasks) submitted by users. An Energy Aware Task Consolidation (EATC) algorithm is developed where heterogeneity also affects the performance and show significant improvement in energy savings.

V. Existing load balancing algorithm

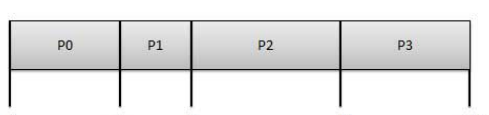
Distributed computing model gives business arranged administrations to an expansive scope of cloud client's on-request finished the Internet. The cloud framework has a few server farms, and every datum focus has an alternate number of heterogeneous servers. The cloud client presents their requests to the cloud service provider (CSP) for the execution reason. The two server farms (i.e., cloud assets) and client necessities are normally heterogeneous. The CSP assigns the submitted client solicitations to a limited number of virtual machines (VMs). Distributed computing framework underpins virtualization strategy. This strategy exhibits a virtualized part of physical assets connected to instantiate virtual machines. A VM Machine (VMM) deals with the physical resources and keeps up partition between VMs. Each VM is independent with its working system.

❖ FCFS Algorithm:

First come, first served (FCFS) is an operating system process scheduling algorithm and a network routing management mechanism that automatically executes queued requests and processes by the order of their arrival.

- Jobs are executed on first come, first serve basis.
- It is a non-preemptive, pre-emptive scheduling algorithm.
- Easy to understand and implement.
- Its implementation is based on FIFO queue.
- Poor in performance as average wait time is high.

Process	P0	P1	P2	P3
Arrival Time	0	1	2	3
Execution Time	5	3	8	6

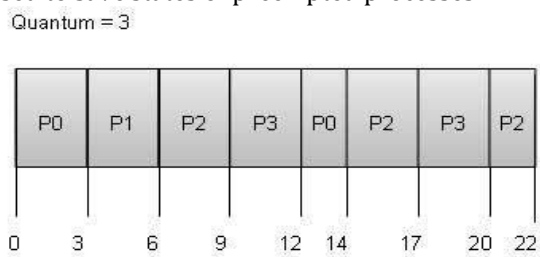


❖ Round-Robin Algorithm:

This method is quite same as the FCFS but the difference is the in this case the processor will not process the whole job (process) at a time. Instead, it will complete an amount of job (quantum) at a

turn and then will go to the next process and so on. When all job has got a turn, it will again start from the first job and work for a quantum of time/cycle on each job and proceed.

- Round Robin is the preemptive process scheduling algorithm.
- Each process is provided a fix time to execute, it is called a **quantum**.
- Once a process is executed for a given time period, it is preempted and other process executes for a given time period.
- Context switching is used to save states of preempted processes.



❖ **Dynamic Load Balancing Algorithm:**

Dynamic load balancing algorithms take into account the different attributes of the nodes' capabilities and network bandwidth. Most of these algorithms rely on a combination of knowledge based on prior gathered information about the nodes in the Cloud and run-time properties collected as the selected nodes process the task's components. These algorithms assign the tasks and may dynamically reassign them to the nodes based on the attributes gathered and calculated. Such algorithms require constant monitoring of the nodes and task progress and are usually harder to implement. However, they are more accurate and could result in more efficient load balancing.

Dynamic Load Balancing depends on the current state of the system. If any node is overloaded then its load is shifted to the under loaded node. So real time communication is performed here.

VI. simulation setup

In this we have used the cloud analyst tool. We have taken six regions, two data centers, many user bases and 5 virtual machines for each data centers. The output of simulation is shown graphically.

Cloud Analyst is built on top of CloudSim toolkit, by extending its functionalities with the introduction of concepts that model Internet and Internet Application behavior. Cloud Analyst has the following main components:

- GUI Package: The front-end is the graphical user interface to control screen transitions and related functionalities.
- Simulation: This important component enables the development and execution of simulation by retaining the simulation parameters.
- User Base: It is used to model the users and users' traffic.
- DataCenter Controller: This module tackles with the activities related to the data center.
- Internet: This is used to exhibit the Internet and traffic routing**.**

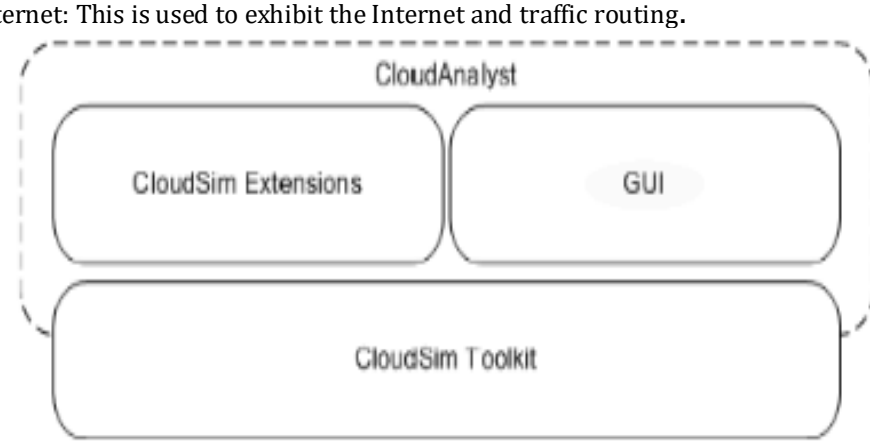


Fig1: Cloud Analyst Architecture

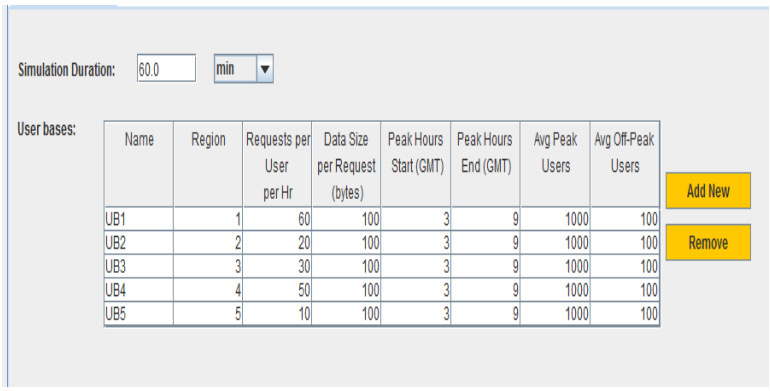


Fig2: User base cinfiguration

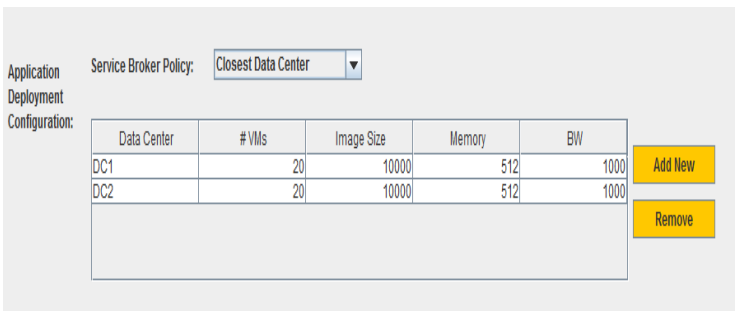


Fig3: Data center configuration

Table1: Input values

Parameters	DLBA	Round Robin	FCFS
Data center	2	2	2
UB	5	5	5
VM	20	20	20

VII. RESULT AND ANALYSIS

The result shows as,

Table 2: Result

Parameter	DLBA	Round Robin	FCFS
Data center	2	2	2
UB	5	5	5
VM	20	20	20
Response Time	200.34	221.88	264.56
Total Cost	130.74	133.52	140.23

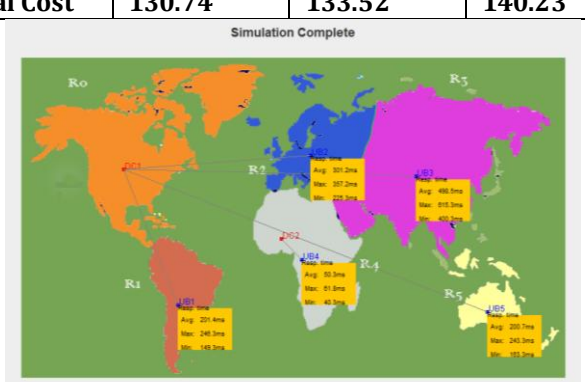


Fig4: simulation result

e ISSN 2348 –1269, Print ISSN 2349-5138
Cosmos Impact Factor 4.236

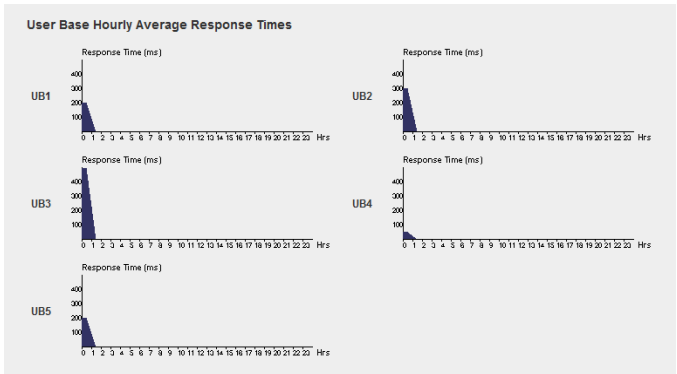


Fig5: User Base hourly response time

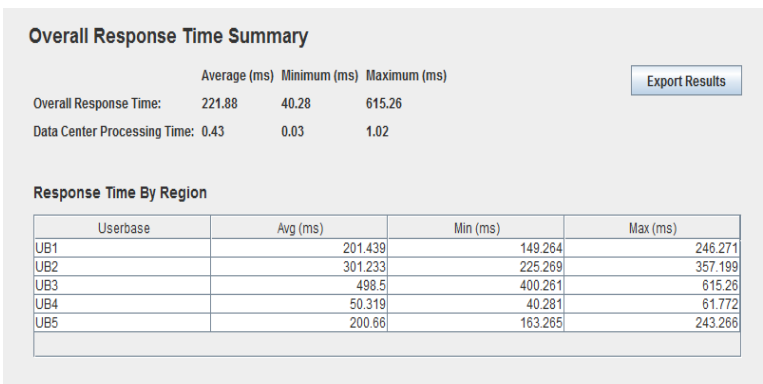


Fig6: response time using region

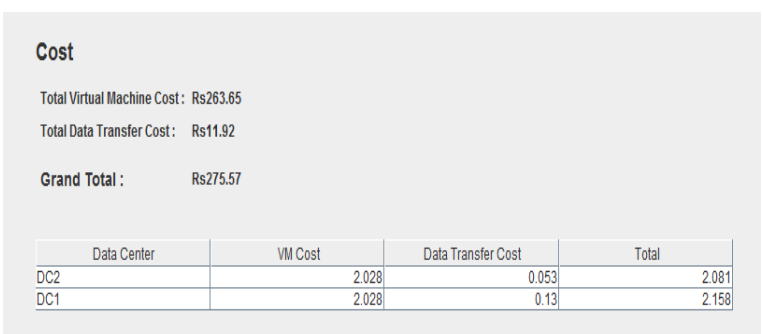


Fig7: Total cost of data centers

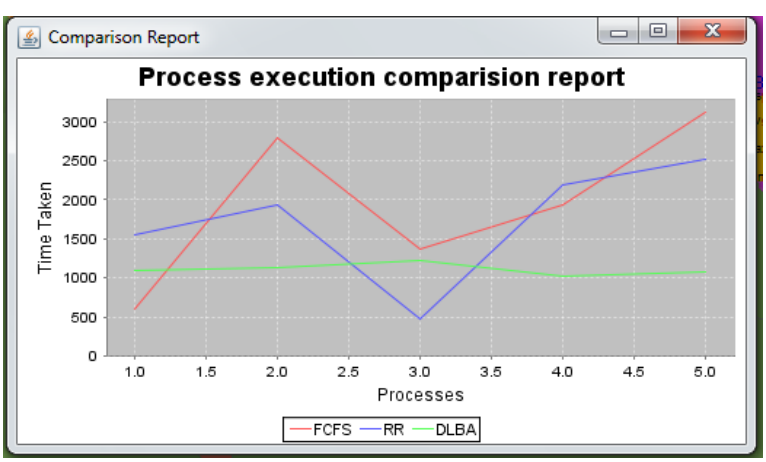


Fig8: Simulation Graph

We have compared the DLBA makespan with the Round- Robin (RR) and First-Come-First-Serve(FCFS) algorithms, the DLBA response time is better than the Round-Robin and FCFS. And the Round-Robin is better than the FCFS.

VIII. CONCLUSION

In this paper we have compared the dynamic Load Balancing algorithm which reduces the server burden and handles the multiple requests by properly scheduling the tasks. Here we have used Round-Robin algorithm to schedule the processes. The DLBA response time is good results from the Round-Robin and the FCFS. The system has a good makespan compare to other scheduling algorithms. In future we are trying to improve dynamic load balancing algorithm.

References

1. Shyam Patidar; Dheeraj Rane; Pritesh Jain "A Survey Paper on Cloud Computing" in proceeding of Second International Conference on Advanced Computing & Communication Technologies, 2012.
2. Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAWHILL Edition 2010.
3. R. Shimonski, Windows 2000 And Windows Server 2003, Clustering and Load Balancing Emeryville, McGrow-Hill Professional Publishing, CA, USA, 2003.
4. M. A. Netto, C. Vecchiola, M. Kirley, C. A. Varela, and R. Buyya. "Use of run time predictions for automatic Co-allocation of multicluster resources for iterative parallel applications." Journal of Parallel and Distributed Computing, Vol. 71(10), pp. 1388-1399, 2011.
5. M. Uddin and A. A. Rahman. "Techniques to implement in green data centres to achieve energy efficiency and reduce Global warming effects." International Journal of Global Warming, Vol. 3(4), pp. 372-389, 2011.
6. J. O. Gutierrez- Garcia and A. Ramirez-Nafarrate. "Collaborative agents for distributed load management in cloud Data centers using live migration of virtual machines." IEEE Transactions on Services Computing, Vol. 8(6), pp. 916-929, 2015.
7. D. B. LD, and P. V.a Krishna. "Honey bee behavior inspired load balancing of tasks in cloud computing environments." Applied Soft Computing, Vol. 13(5), pp. 2292-2303, 2013.
8. D. Puthal, B. Sahoo, S. Mishra, and S. Swain. "Cloud computing features, issues, and challenges: a big picture." In International Conference on Computational Intelligence and Networks (CINE), pp. 116-123, 2015.
9. S. K. Mishra, R. Deswal, S. Sahoo, and B. Sahoo. "Improving energy consumption in cloud." In 2015 Annual IEEE India Conference (INDICON), pp. 1-6, 2015.
10. Y. Zhao and W. Huang. "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud." In INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on, pp. 170-175, 2009.
11. X. GAO, L. Kong, W. Li, W. Liang, Y. Chen, and G. Chen. "Traffic load balancing schemes for devolved controllers in mega data centers." IEEE Transactions on Parallel and Distributed Systems, Vol. 28(2), pp. 572-585, 2017.
12. M. Mishra, A. Das, P. Kulkarni, and A. Sahoo. "Dynamic resource management using virtual machine migrations." IEEE Communications Magazine, Vol.50 (9), pp. 34-40, 2012.
13. [13]A. Shawish, and M. Salama. "Cloud computing: paradigms and technologies." In Inter-cooperative collective intelligence: Techniques and applications, pp. 39-67, 2014.