*International Journal of Application or Innovation in Engineering & Management (IJAIEM)*
**Web Site: www.ijaiem.org Email: editor@ijaiem.org**
**Volume 3, Issue 10, October 2014** **ISSN 2319 - 4847**

# Comparison of Highest Response Ratio Next Algorithm with First Come First Served in a CloudComputing

**Rakesh Kumar Sanodiya, Dr.Varsha Sharma**

RGPV Bhopal, India Department of SOIT
RGPV Bhopal, India

**ABSTRAC**

***"Cloud computing" is a term, which involves virtualization, distributed computing, networking, software and Web services. This new paradigm has experienced a fantastic rise in recent years. Because of itsinfancy, it remains a model to be developed. In particular, it must offer the same features of services than traditional systems. The cloud computing is large distributed systems that employ distributed resources to deliver a service to end users by implementing several technologies. Hence providing acceptable response time for end users, presents a major challenge for cloud computing. Virtual machine (VM) is a key component of cloud computingtechnology. Therefore developing an optimal scheduling mechanism for balancing VM operations at cloud computing framework is an intriguing issue for cloud computing service performance.Load balancing is the method of distributing the load among all the processors or every node in the system so that every node or processor gets equal amount of load at any instant of time.In view of the load balancing problem in VM resources scheduling, this paper presents a scheduling strategy on load sharing of VM resources based on Load management algorithm as well as it also provides simulated results based on the scheduling algorithm like FCFS and HRRN algorithms. Our algorithm maintains the state of all compute nodes, and based on utilization percentages, decides the number of compute nodes that should be operating. We show that our Load management algorithm provides adequate availability to compute node resources while decreasing the overall power consumed by the local cloud as compared to using any other load balancing techniques that are power aware.***

**Keywords:-**Cloud Computing, HRRN, FCFS, Load Balancing, Task, Scheduling, Cloud Storage, Replications, VM.

## 1. INTRODUCTION

Today, computing becomes steadily more important and more used. The amount of data exchanged over the network or stored on a computer is in constant increasing. Thus, the processing of this increasing massof data requires more computer equipment to meet the different needs of organizations. To better capitalize their investment, the overequipped organizations open their infrastructure to others byexploiting the Internet and related technologies like Web 2.0 and other emerging technologies such as virtualization by creating a new computing model: the cloud computing[1].Cloud computing is starting to provide an environmentwhereby Web Services can realise their initially promised potential. Up to the present time, Web Services within Service Oriented Architectures (SOA) have been used in alimited way within business boundaries for integration of applications [5]. The predicted widespread availability and uptake of web-delivered services has not occurred to anygreat scale [6]. Commonly cited reasons include; high complexity and technical expertise required, large expense of implementation and maintenance, and the inflexibilityand lack of widely accepted standards for defining service cooperation, identification and orchestration [7]. These concerns arise as a consequence of associated servicearchitecture management and maintenance difficulties. The scale and complexity of these systems makes centralized governance of specific servers infeasible; requiring effective distributed solutions. Distributedgovernance, achieved through local knowledge, is a vital prerequisite in order to enable the vision inherent in the Internet of Services/Things (IoS/T) model ofservice/hardware provision [17]. CLOUD computing enables developers to automatically deploy applications during task allocation and storage distribution by using distributed computing technologies in numerous servers [2,3]. Load management in cloud computing systems is really a challenge now. Always a distributed solution is required. Because it is not always practically feasible or cost efficient to maintain one or more idle services just as to fulfillthe required demands. Jobs can't be assigned to appropriate servers and clients individually for efficient load balancing as cloud is a very complex structure and components are present throughout a wide spread area. Here some uncertainty is attached while jobs are assigned. This paper considers some of the methods of load balancing in large scale Cloud systems. Our aim is to provide an evaluation and comparative study of these approaches, demonstrating different distributed algorithms for load balancing and to improve the different performance parameters like throughput, latency etc. for the clouds of different sizes. As the whole Internet can be viewed as a cloud of many connection-less and connection-oriented services, thus concept of load balancing in Wireless sensor networks (WSN) proposed in [18] can also be applied to cloud computing systems as WSN is analogous to a cloud having no. of master computers (Servers) and no. of slave computers (Clients) joined in a complex structure. A comparative study of different algorithms has been carried out using divisible load scheduling theory proposed in [4].

## 2. CLOUD COMPUTING

Cloud computing, as a current commercial offering,started to become apparent in late 2007 [8]. It wasintended to enable computing across widespread anddiverse resources, rather than on local machines or atremote server farms. Although there is no standarddefinition of Cloud Computing, most authors seem toagree that it consists of clusters of distributed computers(Clouds) providing on-demand resources or services overa network with the scale and reliability of a data centre [9];notions familiar from resource virtualisation and Grid computing. Where these clusters supply instances of ondemandCloud computing; provision may be comprised ofsoftware (e.g. Software as a Service, SaaS) or of thephysical resources (e.g. Platform as a Service, PaaS). TheAmazon Elastic Compute Cloud (Amazon EC2) [10] is anexample of such an approach, where a computing platformis provided. In common with many commercialapproaches provision is the primary objective;management and governance handled via redundancy orreplication, scaling capacity up or down as required. Incontrast the authors proposed a Cloud Coordinationframework in 2005 with the notion of a Cloud being asystem of loose boundaries, which interacts and mergeswith other systems [11]. This definition of a Cloud is refined to a federation of interacting services and resources, which share and pool resources for greater efficiency. Thus governance, in general, and scalability are handled as part of the separated coordination framework. This separation permits sophisticated implementations ofmanagement techniques, such as load balancing.Until recently the major works on load balancingassumed homogeneous nodes. This is obviously unrealisticfor most instances of Cloud computing, as defined herein,where dynamic and heterogeneous systems are necessaryto provide on-demand resources or services. In theAmazon EC2, dynamic load balancing is handled byreplicating instances of the specific middleware platformfor Web services. This is achieved through a trafficanalyser, which tracks the time taken to process a clientrequest. New instances of the platform are started whenthe load increases beyond pre-defined thresholds [12].Therefore, combinations of rules prescribe thecircumstances and solution for load balancing. As thesystems increase in size and complexity, these rule setsbecome unwieldy and it may not be possible to maintain aviable monitoring and response cycle to manage thecomputational workload. In short, the size of these systemsmay exceed the capabilities of attached meta-systems tomaintain a sufficiently agile and efficiently organized loadbalancing (or general management) rule-set. When somany management rules are defined within a system, thereare likely to be conflicts amongst the rules; interactionsand impact are in general very difficult to analyses. Forinstance, the execution of one rule may cause an event,triggering another rule or set of rules, dependent on current state. These rules may in turn trigger further rules andthere is a potential for an infinite cascade of policyexecution to occur. Additionally these rules are static innature; there is usually no provision for rule refinement oranalysis. A system rule requiring alteration or adjustmentnecessitates the system or component being taken offline,reprogrammed and deployed back into the system.Thus, as an example management task; a load balancingsystem is required that self-regulates the load within the Cloud's entities without necessarily having to have full knowledge of the system. Such self-organised regulationmay be delivered through distributed algorithms; directlyimplemented from naturally observed behaviour,specifically engineered to maintain a globally-balancedload, or directly altering the topology of the system to enhance the natural pattern of load distribution.

## 3. LOAD BALANCING

The goal of load balancing is improving the performanceby balancing the load among these various resources (networklinks, central processing units, disk drives…) to achieveoptimal resource utilization, maximum throughput, maximumresponse time, and avoiding overload.To distribute load on different systems we use generallytraditional algorithms like who's used in web servers, but thesealgorithms do not always give the expected performance withlarge scale and distinct structure of service-oriented datacenters [14]. To overcome the shortcomings of thesealgorithms, load balancing has been widely studied byresearchers and implemented by computer vendors indistributed systems.

**A. Goals of Load balancing:**As given in [15], the goals of load balancing are:

1. To improve the performance substantially
2. To have a backup plan in case the system fails even partially
3. To maintain the system stability
4. To accommodate future modification in the system

**B. Types of Load balancing algorithms**

a. First Come First Server.
b. Highest Response Ratio Next.
c. Round Robin.

## 4.PROBLEM DEFINATION

The random arrival of load in cloud computing can cause some server to be heavily loaded while other server is idle or only lightly loaded. Our HRRN algorithm will equally divide the load to all the Nodes. Efficient scheduling and resource allocation is a critical characteristic of cloud computing based on which the performance of the system is estimated. The considered characteristics have an impact on cost optimization, which can be obtained by improved response time and processing time.

## 5. DESIGN MODEL

To handle the random selection based load distributedproblem, we have proposed a scheduling algorithm and compared it with the existing first come first server scheduling to estimate response time, processing time, which is having an impact on cost .A Comparison of Dynamic Load Balancing Algorithms.
**HRRN**:- HRRN is a non-preemptive discipline, similar to shortest job next, in which the priority of each job is dependent on its estimated run time, and also the amount of time it has spent waiting, jobs gain higher priority the longer they wait, which prevents the longer they wait, which prevents indefinite postponement. In fact, the jobs that have spent a long time waiting compete against those estimated to have short run times.

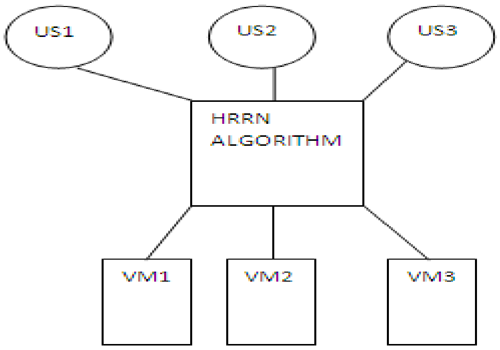


**Figure 1**.Equally spread Active execution load to the cloud system

**HRRN Formula:**

Priority=(Waiting time + Estimated run time)/estimated run time
Priority=1+ waiting time/ estimated run time

**HRRN Algorithm: LOAD ALGORITHM ACTIVE VM LOAD BALANCER [START]**
**Step 1**: Insert all the virtual machines which want to share the load.
**Step 2**: Find out the Response Ratio of all the virtual machines by applying the following formula.
**Response Ratio**=(W+S)/S
Where W=Waiting Time
S=Service Time or Burst Time
**Step 3**: Select one of the virtual machine among the virtual machines for those we found Response ratio.
**Step 4**: Give the load to that virtual machine which I have selected.
**Step 5**: After completion go to the step 1: [END]

## 6.PERFORMANCE ANALYSIS

Here we are going to use Cloud analysis tool for a period of one hour to evaluate the proposed algorithm for the number of users and data centers. Set simulation according to table 1 and 2.
**UserBase**:- The design model use the user base to represent the single user but ideally a user base should be used to represent a large numbers of users for efficiency of simulation.

**Table1.** User Base

| User Base | Region |
|---|---|
| UB1 | 0 |
| UB2 | 1 |
| UB3 | 2 |
| UB4 | 3 |
| UB5 | 4 |
| UB6 | 5 |

**DataCenter:-** Datacenter manages the data management activities virtual machines creation and destruction and does the routing of user requests received from user base via the internet to virtual machines.

**Table2.** Data Center

| DATACENTER | VM M1 | VM M2 | VM M3 | VM M4 | VM M5 | VM M6 |
|---|---|---|---|---|---|---|
| DC1 | 1 | 10 | 25 | 50 | 75 | 100 |
| DC2 | 1 | 10 | 25 | 50 | 75 | 100 |
| DC3 | 1 | 10 | 25 | 50 | 75 | 100 |
| DC4 | 1 | 10 | 25 | 50 | 75 | 100 |
| DC5 | 1 | 10 | 25 | 50 | 75 | 100 |
| DC6 | 1 | 10 | 25 | 50 | 75 | 100 |

**First Come First Server Algorithm Time and Cost For the given virtual machines:-**

**Table3.** FCFS

| FCFS WITH VM | TIME | COS |
|---|---|---|
| FCFS-1 | 67.00 | 19.0 |
| FCFS-10 | 50.45 | 20.0 |
| FCFS-25 | 64.00 | 40.0 |
| FCFS-50 | 67.47 | 23.8 |
| FCFS-75 | 50.73 | 33.4 |
| FCFS-100 | 80.00 | 37.0 |

**Highest Response Ratio Next AlgorithmTime and Cost For the given virtual machines:**

**Table4.** HRRN

| HRRN WITH VM | TIME | COST |
|---|---|---|
| HRRN-1 | 67.00 | 19.09 |
| HRRN -10 | 50.45 | 20.01 |
| HRRN -25 | 64.00 | 40.01 |
| HRRN -50 | 67.47 | 23.89 |
| HRRN -75 | 50.73 | 33.45 |
| HRRN -100 | 80.00 | 37.00 |

**Table5.** Comparison b/w FCFS Time and HRRN Time

| VIRTUAL MACHINE | HRRN TIME | FCFS TIME |
|---|---|---|
| R 1 | 57.00 | 67.00 |
| R 10 | 40.45 | 50.45 |
| R 25 | 60.00 | 64.00 |
| R 50 | 61.47 | 67.47 |
| R 75 | 40.73 | 50.73 |
| R 100 | 70.00 | 80.00 |

**Table5.** Comparison b/w FCFS Cost and HRRN Cost

| VIRTUAL MACHINE | HRRN Cost | FCFS Cost |
|---|---|---|
| R 1 | 18.00 | 19.09 |
| R 10 | 17.45 | 20.01 |
| R 25 | 36.00 | 40.01 |
| R 50 | 20.47 | 23.89 |
| R 75 | 30.73 | 33.45 |
| R 100 | 35.00 | 37.00 |

## 7.CONCLUSION& FUTURE WORK

To develop products for every IT engineer Cost and Time are the key challenges. These can increase the businessperformance in the cloud. Current Techniques in Cloud Computing leading to increased operational cost and time. This paper aimstowards the development of enhanced strategies throughimproved job scheduling and resource allocationtechniques for overcoming the above-stated issues. Here,Highest Response Ratio Next Algorithmdynamically allocates the resources to the job in queueleading reduced cost in data transfer and virtual machineformation. The simulation results show overall time andcost results and comparison of load balancing algorithms. This time my paper work solve the load distributing problemon various nodes of a distributed system to improve bothresource utilization and job response time while alsoavoiding a situation where some of the nodes are heavilyloaded while other nodes are idle or doing very little work.Load balancing ensures that all the processor in the systemor every node in the network does approximately the equalamount of work at any instant of time. The simulated resultsprovided in this paper based on scheduling algorithm HRRN(Highest Response Ratio Next) load .HRRN Schedulingalgorithms handle the random selection based loaddistributed problem First Come First Serve , we have proposed HRRNscheduling algorithm and compared it with the FCFSscheduling to estimate response time, processing time,which is having an impact on cost. My future work is basedon overloading server or overflow server load. In futureovercome the server overflow problem using algorithm andimprove the load distribution performance.

## 8.RESULT

When we are comparing with the table and graph, overall response time and data centre processing time is improved. it is also seen that the virtual machine time and data transfer time in HRRN is much better when compared to FCFS and round robin.

## REFERENCES

[1] A.KHIYAITA, M.ZBAKH, and H. EL BAKKALI. Load Balancing Cloud Computing : State of Art.IEEE Computer, 25(12), pp. 33-44, December2012 ;

[2] M.D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis and A. Vakali, ”Cloud Computing: Distributed Internet Computing for IT and Scientific Research,” IEEE Internet Computing, Vol.13, No.5, pp.10-13, 2009.

[3] B. Ahlgren, P.A. Aranda, P. Chemouil, S. Oueslati, L.M. Correia, H. Karl, M. Sollner and A. Welin, ”Content, Connectivity and Cloud: Ingredientsfor the Network of the Future,” IEEE Communications Magazine, Vol.49, No.7, pp.62-70, 2011.

[4]Jiann-Liang Chen, YanuariusTeofilusLarosa and Pei-Jia Yang, Optimal QoS Load Balancing Mechanism forVirtual Machines Scheduling in Eucalyptus CloudComputing Platform 2012 2nd Baltic Congress on Future Internet Communications

[5] C.Schroth and T. Janner, (2007). Web 2.0 and SOA: ConvergingConcepts Enabling the Internet of Services. IEEE IT Professional Vol.9, No.3 (pp.36-41), June 2007.

[6] P.A Laplante, Zhang Jia and J.Voas, “What's in a Name?Distinguishing between SaaS and SOA," IT Professional , vol.10, no.3, pp.46-50, May-June 2008

[7] W.M.P. van der Aalst, Don't go with the flow: Web servicescomposition standards exposed. IEEE Intelligent Systems, 18(1):72-76, 2003.

[8] IBM. IBM Introduces Ready to Use Cloud Computing. IBM PressRelease, 15th November 2007. http://www03.ibm.com/press/us/en/pressrelease/22613.wss

[9] R.L. Grossman, "The Case for Cloud Computing," IT Professional,vol.11, no.2, pp.23-27, March-April 2009

[10] Amazon Elastic Compute Cloud (EC2),http://www.amazon.com/gp/browse.html?node=201590011

[11] P. Miseldine and A.Taleb-Bendiab, A Programmatic Approach toApplying Sympathetic and Parasympathetic Autonomic Systems toSoftware Design, in Self-Organisation and Autonomic Informatics (1) (Ed: H. Czap et al) pp: 293-303, IOS Press, Amsterdam, 2005.

[12] A. Azeez, Auto-Scaling Axis2 Web services on Amazon EC2.ApacheCon Europe 2009, Amsterdam, March 2009.

13] Martin Randles, A. Taleb-Bendiab and David Lamb, Cross LayerDynamics in Self-Organising Service Oriented Architectures.IWSOS, Lecture Notes in Computer Science, 5343, pp. 293-298, Springer, 2008.

[14] S. Nakrani and C. Tovey, On Honey Bees and Dynamic ServerAllocation in Internet Hosting Centers. Adaptive Behavior 12, pp: 223-240 (2004).

[15] Yi Lu, QiaominXie, Gabriel Kliot, Alan Geller, James R. Larus, AlbertGreenberg, Join-Idle-Queue: A Novel Load Balancing Algorithm forDynamically Scalable Web Services, IFIP PERFORMANCE 2011 29thInternational Symposium on Computer Performance, Modeling,Measurements andEvaluation 2011, 18-20 October, 2011, Amsterdam,Netherlands ;

[16] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Sys-tems, IJCSNS International Journal of Computer Science and Network Security,VOL.10 No.6, June 2010.

[17] Peter S. Pacheco, ”Parallel Programming with MPI”, Morgan Kaufmann Publishers Edition 2008

[18] MequanintMoges, Thomas G.Robertazzi, ”Wireless Sensor Networks: Scheduling forMeasurement and Data Reporting”, August 31, 2005