RESEARCH ARTICLE OPEN ACCESS

A Highest Response Ratio Next(HRRN) Algorithm Based Load Balancing Policy For Cloud Computing

Rakesh Kumar Sanodiya, Dr. Sanjeev Sharma, Dr. Varsha Sharma
Department of School of Information Technology
Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal
M.P - India

ABSTRACT

Cloud Computing is a business oriented online shop where IT resources and IT services are sold on demand using pay per use model. The main objective of the service provider is to gain maximum profit by using cloud computing resources efficiently. The virtual machine (VM) resources scheduling strategy in cloud computing online shop mainly takes the current state of the system but randomly takes system variation and historical data, which always leads to load imbalance of the system. Load balancing which is one of the main challenges in Cloud computing, distributes the dynamic workload across multiple nodes to ensure that no single resource is either overwhelmed or underutilized. This paper proposes a load balancing policy using Highest Response Ration Next (HRRN) algorithm. The algorithm blooms to balance the load of the cloud infrastructure. The proposed load balancing strategy has been analyzed with the Cloud simulator. Analysis results for a typical cloud computing environment shows that the proposed algorithm outperformed the existing approaches like First Come First Serve (FCFS), Round Robin (RR), Equally Spread Current Execution Load and Throttled.

Keywords:- Cloud Computing, Load Balancing, HRRN, FCFS, Round Robin, Virtulization.

1. INTRODUCTION

The boom in cloud computing has been sort of astounding; in just five years cloud computing has reshaped the way ICT organization are given and used. There are as various definitions, groupings and developments for cloud computing as the amount of foundations accepting it, and this number is on the climb. Cloud computing is described by the US Government's National Institute of Standards and Technology (NIST) [1] as an ICT sourcing and transport model for engaging favorable, on-investment framework access to a conferred pool of configurable computing resources (e.g. frameworks, servers, stockpiling, applications and organizations) that can be immediately provisioned and released with immaterial organization tries and organization supplier cooperation's. Virtualization advances are the key enabling specialists of cloud computing by giving suppliers a versatile system for managing their advantages. Virtual system (VI) organization is a key concern when building cloud circumstances and speaks to different challenges. Among the essential challenges in VI organization is to execute a powerful load balancer prepared for consistently pass on the workload among open cloud resources [2]. In view of the exponential advancement of cloud computing, it has been extensively gotten by the business and thusly making a quick expansion in availability of advantage in the Internet. As the degree of cloud scales up cloud computing organization suppliers obliges treatment of huge claims. The crucial test then becomes to keep the execution same or better at whatever point such a change happens. Thusly paying little respect to extraordinary inevitable destiny of Cloud Computing, various separating issues still need to be researched for its optimal affirmation [3]. One of these issues is Load adjusting. At the point when all is said in done terms, the load adjusting is a framework to course the workload of the structure transversely over diverse servers to ensure that no server is discovered up with dealing with a staggering workload while an interchange server is unmoving. Thusly, a load balancer can be considered as an opposite go-between that passes on framework or application development over different servers. Load balancers development finishing three standard targets. In any case, improving general structure execution by landing at high resource utilization extent. Second, staying far from structure bottleneck that happens as a result of load unevenness. Finally, finishing high suppliers' and customers' satisfaction by striving to construct the structure throughput and decay the vocation taking care of time [4]. Basically, load balancers can be passed on concentrated around three different architectures: The united load adjusting development demonstrating which joins a central load balancer to settle on the decision for the entire system as for which cloud resource should take what workload and concentrated around which algorithm(s) [5]. This building outline has the known inclination of capable organization plan however encounters poor flexibility and constitutes a singular reason for bafflement. The decentralized load adjusting building outline has no central load balancer to suitable workload among open resources; rather, occupation requesting are apportioned on landing, similarly among various load balancers where each of them may run an interchange figuring to appropriate occupations to resources. This structural building offers uncommon flexibility and adaptability. On the other hand, it yields poor load balance among shrouded resources [6].

In this paper Highest Response Ration Next (HRRN) count) has been used as a sensitive computing procedure, which uses the instrument of basic decision system. Cloud analyst - A Cloudsim based Visual Modeler has been used for reenactment and examination of the estimation. The

execution of the figuring is differentiated and Four ordinarily used booking count like First Come First Serve (FCFS), Round Robing (Rr),equally Spread Current Execution Load and Throttled . Whatever is left of paper is dealt with as takes after. Territory 2 proposes the HRRN computation for load adjusting. Territory 3 shows the amusement results and its examination with a blueprint of Cloudanalyst in Section 3.1 with the end goal of zenith. Finally, Section 4 completions up this paper.

II. OVERVIEWOF LOAD BALANCING

The goal of load adjusting is enhancing the execution by adjusting the load among these different assets (system joins, focal transforming units, plate drive) to accomplish ideal asset usage, greatest throughput, most extreme reaction time, and maintaining a strategic distance from overload [8]. To convey load on distinctive frameworks we utilize by and large customary calculations like who's utilized as a part of web servers, however these calculations don't generally give the normal execution with expansive scale and unique structure of administration arranged server farms [9]. To defeat the deficiencies of these calculations, load adjusting has been broadly concentrated on via analysts and executed by machine sellers in conveyed frameworks. All in all, load-adjusting calculations take after two noteworthy groupings [10]:

• Depending on how the charge is conveyed and how procedures are allotted to hubs (the framework load);

• Depending on the data status of the hubs (System Topology).

In the first case it outlined as incorporated approach, appropriated approach or mixture approach in the second case as static approach, dynamic or adaptive approach. [11]

A) CLASSIFICATION ACCORDING TO SYSTEM LOAD

a) Centralized approach: In this approach, a single node is responsible for managing the distribution within the whole system.
b) Distributed approach: In this approach, each node independently builds its own load vector by collecting the load information of other nodes. Decisions are made locally using local load vectors. This approach is more suitable for widely distributed systems such as cloud computing.
c) Mixed approach: A combination between the two approaches to take advantage of each approach.

B) CLASSIFICATION ACCORDING TO THE SYSTEM TOPOLOGY

a) Static Approach: This approach is generally defined in the design or implementation of the system.
b) Dynamic Approach: This approach takes into account the current state of the system during load balancing decisions.
This approach is more suitable for widely distributed systems such as cloud computing.
c) Adaptive approach: This approach adapts the load distribution to system status changes, by changing their parameters dynamically and even their algorithms. This approach is able to offer better performance when the system state changes frequently [11], [12]. This approach is more suitable for widely distributed systems such as cloud computing.

III. LOAD BALANCING PARAMETERS IN CLOUDS

The factors that always be considered in various load balancing techniques in cloud computing are as follows Detailed description of the load balancing factor is as follows:

• **Response Time -** It is the amount of time taken to provide the response by some load balancing algorithm in a distributed environment. This parameter should be minimized. It is represented as R (t).
Formula to calculate the Response Time is:
R (t) = Finish Time - Start Time.
= T (f) – T(s)
Where T(f) is finish time and T(s) is start time.

• **Communication Time** - It is defined as time taken by number of hops to travel in the communication channel. It is represented by C (t). Formula to calculate the Communication Time is: C(t)=2(Number of hops*Time to traverse between hops)

• **Processing Time -** It is defined as the difference between Communication Time and Response Time. It is represented by P (t). Formula to calculate the Processing Time is: P (t) = Response Time- Communication Time
= R (t) – C(t)

• **Throughput -** is used to calculate the number of tasks whose execution has been completed. It should be high to improve the reliability and the performance of the system. It is represented as Th (Vi).
Th (Vi) = (Cloudlet length*Number of cloudlets)/ Response Time
Th(Vi)= [Length (Ci) – Ni] / R(t)
Where Length (Ci) is cloudlet length and Ni is number of cloudlets for specific virtual machine.

• **Network Delay -** Delay in sending request and receiving response. It is the time taken by the network to send the number of cloudlets to particular VM and time taken by the VM to receive the cloudlets.
D (t) = No. of cloudlets / Rate of transmission

=N/r

Where "r" is the rate of transmission.

IV. HRRN FOR LOAD BALANCING IN CLOUD COMPUTING

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 [13]

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**]

V. SIMULATION RESULTS AND ANALYSIS

The proposed HRRN algorithm is simulated by considering a scenario of "Internet Banking" of an international bank in a simulation toolkit Cloud Analyst.

5.1. Cloud Analyst

To support the base and application-level prerequisites emerging from Cloud computing standard, for example, demonstrating of on interest virtualization empowered asset test systems are needed. Cloud Analyst has been utilized as a part of this paper as a simulation apparatus.
A depiction of the GUI of Cloud Analyst simulation tool compartment is indicated in figure 1(a) and its building design in portrayed figure figure2 (b).

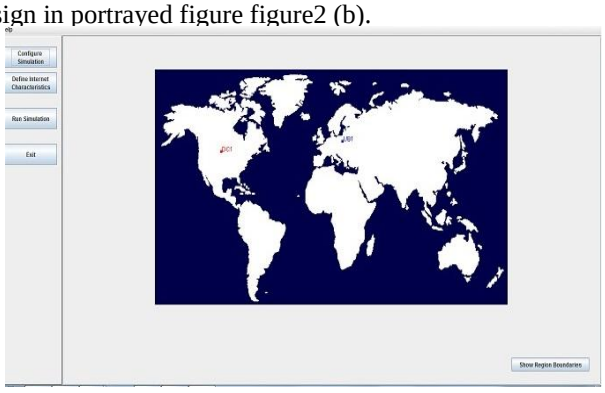

Figure. 1:(a)

Fig. 1: Snapshot of Cloud Analyst (a) GUI of Cloud Analyst.

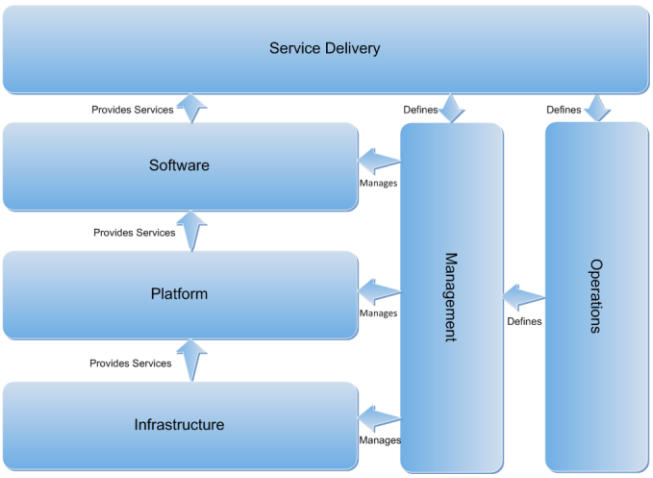

Figure 2:(b)

Fig 2: Diagram of Cloud Architecture (b) Architecture of Cloud Analyst build on CloudSim

Cloud Analyst created on Cloudsim is a GUI based simulation device. Cloudsim encourages demonstrating, simulation and other experimentation on cloud automatically. Cloud Analyst utilizes the functionalities of Cloudsim and does a GUI based simulation. It permits setting of parameters for setting a simulation environment to study any exploration issue of cloud. In view of the parameters the instrument processes, the simulation come about likewise demonstrates to them in graphical structure.

A speculative arrangement has been produced utilizing Cloud Analyst. Where, the world is isolated into 6 "Regions" that concur with the 6 principle mainlands in the World. Six "User bases" demonstrating a gathering of users speaking to the six noteworthy mainlands of the world is considered. A solitary time zone has been considered for the all the user bases and it is accepted that there are shifted number of online enlisted users amid crest hours, out of which stand out twentieth is internet amid the off-top hours. Table 1 rundowns the points of interest of user bases utilized for experimentation. Each one reproduced "server farm has" has a specific measure of virtual machines (Vms) committed for the application. Each of the Machines has 4 GB of RAM and 100gb of capacity and each one machine has 4 Cpus, and every CPU has a limit force of 10000 MIPS.

5.2. Simulation setup

A few situations are considered for experimentation beginning with just a solitary incorporated cloud Data Center (DC). Along these lines all user asks for around the globe are handled by this single DC having 25, 50 and 75 Vms of Cloud Configuration (Ccs) designated to the application. This simulation setup is depicted in Table 2

with figured general normal Response Time (RT) in ms for HRRN, FCFS and ESCE. An execution investigation chart of the same is portrayed in figure 3. Next two Dcs are viewed as each one having a mix of 25, 50 and 75 Vms as given in Table 3 and execution investigation is accounted for in figure 4.

S .No.	User Base	Region	Online users during Peak hrs.	Online users during off-peak hrs.
1.	UB1	India	47	25
2.	UB2	America	60	20
3.	UB3	SriLanka	30	10
4.	UB4	Pakistan	35	12
5.	UB5	Australia	40	23

Table 1: Configuration of simulation environment

5.3. Complexity analysis

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.

Response Ratio= (W+S)/S
Where W=Waiting Time
S=Service Time or Burst Time

5.3.1 Results using one data center

Cloud Conf.	DC specification	RT using HRRN	RT using FCFS	RT using ESCE
CC1	Each with 25 VMs	90.1	95.1	91.
CC2	Each with 50 VMs	85.2	89.0	86.0
CC3	Each with 75 VMs	81.0	84.0	82.4

Table 2: Simulation scenario and calculated overall average response time (RT) in (ms)

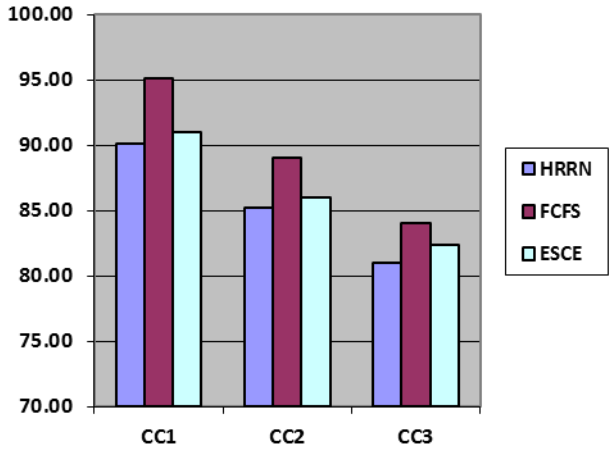

Fig. 3: Performance analysis of proposed HRRN with FCFS and ESCE

5.3.2 Results using two data centers

Cloud Conf.	DC specification	RT using HRRN	RT using FCFS	RT using ESCE
CC1	2 DCs with 25 VMs	95.1	99.1	96.0
CC2	2 DCs with 50 VMs	90.2	92.0	91.0
CC3	2 DCs with 75 VMs	88.0	84.0	82.4
CC4	2 DCs with 25, 50 VMs	87.0	83.0	84.0
CC5	2 DCs with 25, 75 VMs	85.0	86.2	86.0
CC6	2 DCs with 75, 50 VMs	84.0	86.1	85.1

Table 3: Simulation scenario and calculated overall average response time (RT) in (ms)

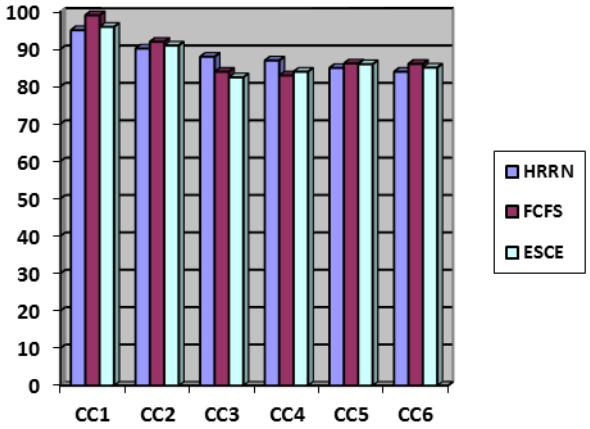

Fig. 4: Performance analysis of proposed HRRN with FCFS and ESCE

VI. CONCLUSION

In this paper, a HRRN based load balancing strategy for Cloud Computing has been developed to provide an efficient utilization of resource in the cloud environment. The proposed strategy for load balancing not only outperforms a few existing techniques but also guarantees the requirement of customer jobs, as analyzed. Assuming the jobs with the same priority can be accommodated for fulfilling their need based on the calculated response ration time. A very simple approach of HRRN has been used as well. The variation of the response time strategies could be applied as a future work for getting more efficient and tuned results.

REFERENCES

[1] P. Mell, T. Grance, " The NIST Definition of Cloud Computing." National Institute of Standards and Technology, Internet: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, [December 11th, 2012].

[2] Ektemal Al-Rayis Computer Science Department Imam Muhammad Ibn Saud Islamic University, "Performance Analysis of Load Balancing Architectures in Cloud Computing",2013 European Modelling Symposium.

[3] A. Vouk, "Cloud computing- issues, research and implementations", in Proc. of Information Technology Interfaces, pp. 31-40, 2008.

[4] N. Kansal, I. Chana. "Cloud Load Balancing Techniques: A Step Towards Green Computing." IJCSI International Journal of Computer Science Issues, Vol. 9, No 1, pp. 238-246, January 2012.

[5] Gaurav R. et al. "Comparative Analysis of Load Balancing Algorithms in Cloud Computing." International Journal of Advanced Research in Computer Engineering & Technology, Vol. 1, No. 3, pp.120-124, May 2012.

[6] B. Addis, D. Ardagna, B. Panicucci, M. S. Squillante, L.Zhang, "A Hierarchical Approach for the Resource Management of Very Large Cloud Platforms, IEEE Transactions On Dependable And Secure Computing, pp. 253-272, Vol. 10, No. 5, 2013.

[7] B. Wickremasinghe, R. N. Calheiros and R. Buyya, "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications", in Proc. of Proceedings of the 24th International Conference on Advanced Information Networking and Applications (AINA 2010), Perth, Australia, pp.446-452, 2010.

[8] A.KHIYAITA Information Security Research Team-ENSIAS University Mohammed V Souissi Rabat Morocco" Load Balancing Cloud Computing : State of Art" IEEE 2012

[9] Yi Lu, Qiaomin Xie, Gabriel Kliot, Alan Geller, James R. Larus, Albert Greenberg, Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services, IFIP PERFORMANCE 2011 29th International Symposium on Computer Performance, Modeling, Measurements and Evaluation 2011, 18-20 October, 2011, Amsterdam, Netherlands ;

[10] Elarbi Badidi, Architecture et services pour la distribution de charge dans les systèmes distribués objet, Université de Montréal Faculté des études supérieures, these doctorale, 20 juillet 2000 ;

[11] N. Shivaratri, P. Krueger, and M. Singhal. Load distributing for locally distributed systems.IEEE Computer, 25(12), pp. 33-44, December 1992 ;

[12] T.L. Casavant and J.G. Kuhl. A Taxonomy of Scheduling in General- Purpose Distributed Computing Systems. IEEE Transactions on Software Engineering, 14(2), pp. 141-154, February 1988

[13] http://en.wikipedia.org/wiki?curid=6659305