International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 1 | ISSN : 2456-3307

Naïve Shared Based Approach of Load Balancing Named As Hybrid Algorithm in Cloud Computing

Nuzhat Fatima, Anugrah , Samendra
Rama University of Technology and Engineering, Kanpur, Uttar Pradesh, India

ABSTRACT

Load balancing play a crucial role in cloud computing in terms of performance. This functionality of load balancing is divided into two functions, first, the allocation of resources and second provision of resources along with task scheduling among distributed system. Many load balancing algorithms such as, FCFS, Round Robin, and Throttled, Equally spread current execution etc. are used for balancing load of cloud computing and make the clouding issues removal. Each algorithm has some points of disadvantage. Proposed algorithm is named as hybrid algorithm that is a combination of Throttled and Current Execution algorithms that equally spread. Proposed algorithm overcomes the drawbacks of throttled and equally spread current execution algorithm.

Keywords: Load Balancing, Shared Based Approach, Hybrid Approach, CloudSim, Throttled, Round Robin, Task Scheduling,

I. INTRODUCTION

Cloud computing is typically the world wide net based mostly computing that presents infrastructure as service (IaaS), platform as service (PaaS), package as Service (SaaS). Within IaaS computing infrastructure may be sent to be a facilitate towards the requester. In current form infrastructure associated with Virtual Machine (VM).These model typically square measure developed seeable from an honest subscription basis utilizing value equally you-use model to be able to customers, regardless concerning their location. The cloud programming plays terribly vital role with policies to be able to manage your order by involving operates for you to presumably be vital role. The programming refers for the set performed by associate computing system. It has been totally different individuals related to programming algorithmic function existing throughout distributed system.

The various objectives of optimization criteria can be [1]:

- **CPU Utilization:** The total percentage of time used for which CPU was utilized was not idle.
- **Throughput:** Total no. of tasks executed (or requests served) per unit time.
- **Response Time:** The total time spent in the waiting queue till it gets the first time to use the CPU.
- **Waiting Time:** Total time spent waiting in the ready queue after the first response from CPU.
- **Turnaround Time:** Total time taken to get completely served, include its response time, waiting time and service time.
- **Fairness:** The principle states that every job can get equal share of CPU time.
- **Resource Cost:** The servicing of various cloud consumers used for total cost of the resources acquired.

The main goal is to maximize the CPU Utilization, maximize the Throughput, minimize the Response Time, Waiting time, Turnaround Time, Resource Cost and obey the Fairness principle.

II. METHODS AND MATERIAL

A. Literature Survey

In Current Scenario, the environment of mobile cloud the task is disseminated into same size of small jobs i.e. Cloudlets. These Cloudlets as well as Virtual Machines are scheduled according to the various scheduling policy for e.g. FCFS, Round Robin etc. Cloud Coordinator (CC) [2] divides the task into equal sized cloudlets and passes it to Datacenter (DC). Basically this default job scheduled policy is extremely Time- Consuming, Cost insensitive and inefficient. Cloud services are currently among the top ranked high growth areas in computer services and seeing an acceleration in enterprise adoption with the worldwide market predicted to reach more than$140b in 2014[3]. For secure cloud bursting and aggregation, the author uses encryption scheme of 64-bit cipher. But they are fail to apply this concept in real environment check for the real time simulations on different platforms.

In paper [4] a brand new VM fill up Balancing Algorithm is actually Weighted Active Monitoring populate Balancing Algorithm applying CloudSim tools, due to the Datacenter to help efficiently load balance requests between ones exhibited virtual devices assigning the weight. In paper [5] author proposed a good algorithm can be ant colony optimization that random optimization search approach is usually obtained pertaining to allocating your current incoming jobs on the virtual machine.
In 2014[6], the problem of load balancing in cloud bursting was discovered and various scheduling algorithms are proposed to solve the problem but the practical implementation of that solutions is still remaining. In 2015[7], automatic cloud bursting was suggested which allows computer resources to be dynamically reconfigured to meet users' demands but the launching of virtual machines on commercial cloud takes much more time than on fermicloud. Hence further more tests in virtual machines' performance on commercial cloud needed to be done in the future.

B. Components of Cloud System

A typical Cloud modeled applying to CloudSim that involves after four entities Datacenters, Hosts, Virtual m/c in addition application form shown in figure 1.

1. **Datacenter:** Datacenter asset of hosts and can be responsible regarding managing virtual models (VMs) (e.g., VM provisioning). It behaves similar to IaaS provider from finding requests with regard to VMs via brokers.
2. **Datacenter Broker:** This class represents the broker acting on behalf of a user. It modifies a couple of mechanisms: ones mechanism for submitting VMs.
3. **Host:** Host executes actions regarding management of VMs (e.g., creation along with destruction) and update task processing to be able to VMs. a good host possesses the defined policy to provisioning memory, processing elements, and also bandwidth to virtual machines.
4. **VM:** This represents the software implementation of a machine that executes applications called virtual machine (VM) which functions to be a physical machine. Each virtual machine divides your own resources received by the host among tasks working from it.
5. **Cloudlet:** CloudSim represents your complexity of the application in relation to their computational requirements. The class is managed through the International Journal of Scientific Research in Science, Engineering and Technology (ijsrset.com) 634 scheduling policy that will be implemented inside Datacenter Broker Class.

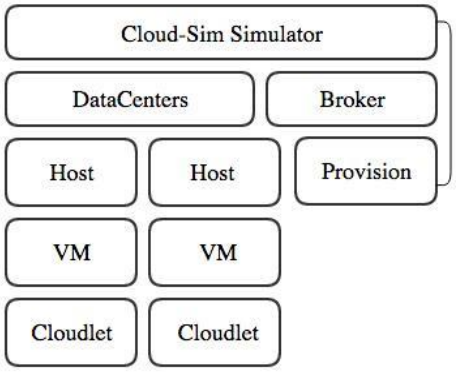

Figure 1. Components of Cloud System

C. LOAD Balancing Algorithms:

Some exiting load-balancing algorithms are listed as under: Token Routing, Round Robin, Randomized, Central Queue, Throttled, Equally spread current execution.

We work on throttled and equal load sharing algorithms.

- **Throttled:** The first method starts by maintain a listing of all the VMs every row is on an individual basis indexed for the fast search process. If match is found on the idea of size and handiness of the machine, then the load balancer accepts the request of the consumer and allocates that VM to the consumer.

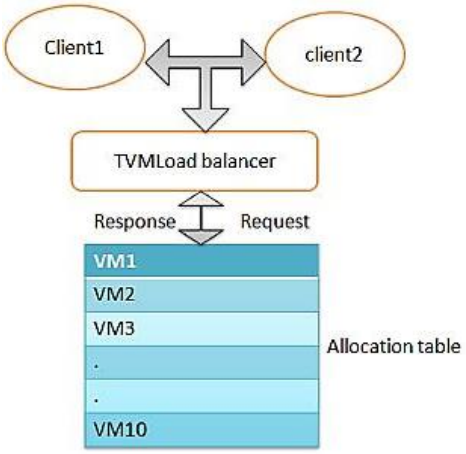

Figure 2. Throttled Algorithm

- **Equally Spread Current Execution:** This algorithmic rule distributes the load indiscriminately by first checkingthe dimensions of the method and so transferring the load to a Virtual Machine that is gently loaded. The load balancer maintains a queue of the roles that require using and are presently mistreatment the services of the virtual machine.

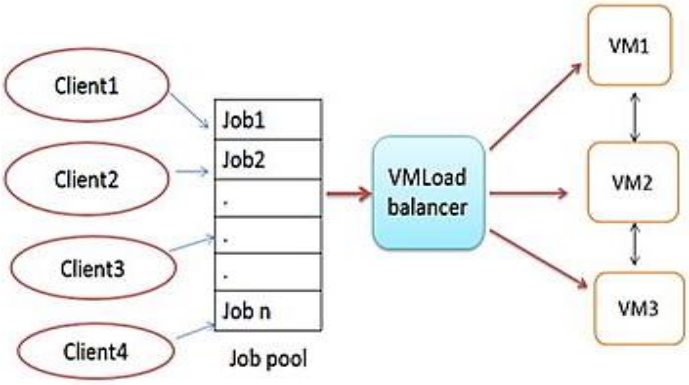

Figure 3. Block diagram of Equally Spread Current Execution Algorithm

D. Problem in existing system

Since the job arrival pattern is not predictable and the capacities of each node in the cloud differ, for load balancing problem, workload control is important to improve system performance and maintain stability. Load balancing schemes depending on whether the system dynamics are important can be either static or dynamic. Static schemes do not use the system information and are less complex while dynamic schemes will bring additional costs for the system

Load balancing schemes depending on whether the system dynamics are crucial can be either static or dynamic. Static schemes do not use the system information and are less complex. Considering that the job arrival pattern isn't predictable and the capacities of each one node in differ, for load managing problem, workload control is important to improve system performance and keep stability.

Cloud-computing environment can be a very complex problem along with load balancing receiving. The job arrival pattern is not predictable and the capacities of each node in the differ for some, for load managing problem, workload control is important to improve system performance and keep stability.

III. ALGORITHM, SIMULATION AND RESULTS

We proposed new hybrid load balancing algorithm, which is combination of throttled and equally spread current execution load balancing algorithm. The method 1st starts by maintain a listing of all the VMs every row is on an individual basis indexed to hurry up the search process. If a match is found on the idea of size and handiness of the machine, then the load balancer accepts the request of the consumer and allocates that VM to the consumer.

The load balancer spreads the load on to completely different nodes, and hence, it's referred to as unfold spectrum technique. The load balancer maintains a queue of the roles that require using and are presently mistreatment the services of the virtual machine. The balancer then unendingly scans this queue and therefore the list of virtual machines.

Algorithm: Hybrid Algorithm

Following is the proposed algorithm. This algorithm maintains an index table of VMs and the state of VM will be either BUSY or FREE.

1. At the start all virtual machines are FREE.
2. Data center controller receives a new request.
3. Data center controller sends queries to the virtual machine load balancer for allocation.
4. Load balancer check for all current allocation count is less than max length of virtual machines list allocation.
    - If virtual machines is not allocation create a new one.
5. Count the active load on each virtual machine.
6. If any virtual machines having least load then return the id of those virtual machines to the virtual machines load balancers.
7. The virtual machines load balancers will allocation the request to one of those virtual machines.
    - If virtual machine is overloaded then the virtual machines load balancer will distribute same of its work to those virtual machines having least work so that every virtual machine will carry equal loaded.
    - When the virtual machines finishes processing the request and the data center controller receives the response cloud, it notifies the virtual machines load balancer of virtual machines reallocation.
8. The data center controller check if there are any waiting request in the queue, if there are, it continues from step4.
9. Continue from step3.

A flow chart of proposed (Hybrid) algorithm is shown in figure 4.

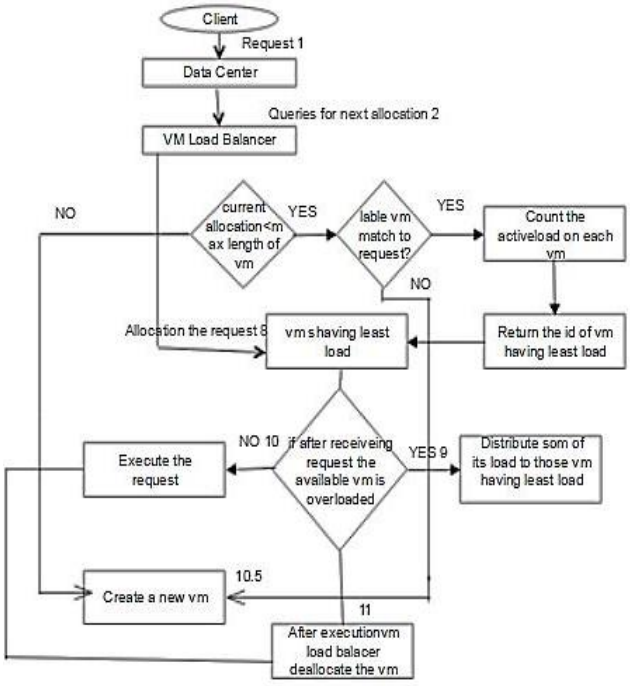

Figure 4. Hybrid Load Balancing Algorithm – A Flow Chart

B. Implementation and Result Analysis

We implemented hybrid algorithm, which is combination of throttled and equally spread current execution balancing algorithm in JAVA. To simulate we use the cloud simulator. Before simulation we configure many parameters as discussed earlier. We implemented three algorithm of load balancing are:

- Hybrid
- Throttled
- Equally spread current execution

Simulation Configuration

- **Main**

Number of User	4
Number of VMs	5
Number of Cloudlets	7

- **Cloudlets**

Length	40000
File Size	300
Output Size	300
Pes Number	1

- **Virtual Machine**

VM Name	Xen
Size	10000
RAM	512
Pes number	1
MIPS	250
BW	1000

- **DaraCenter**

MIPS	1000
RAM	16384
Storage	1000000
BW	100

Below diagrams 5, 6 and 7 shows that execution cloudlets, amount of time needed for execution, it is also showing that which cloudlets assign on which datacenter and virtual machine.

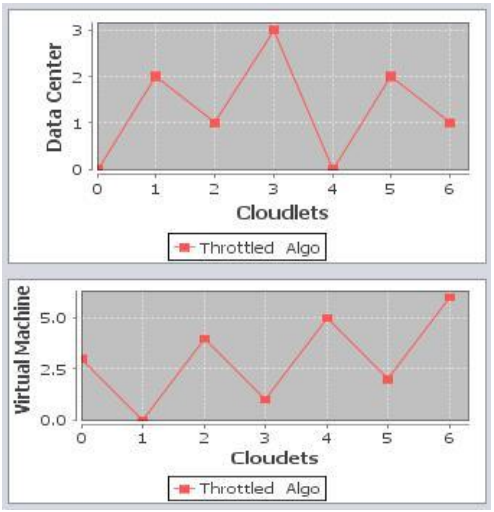

Figure 5. Data Centers and VM Allocation using Throttled Load Balancing Algorithm

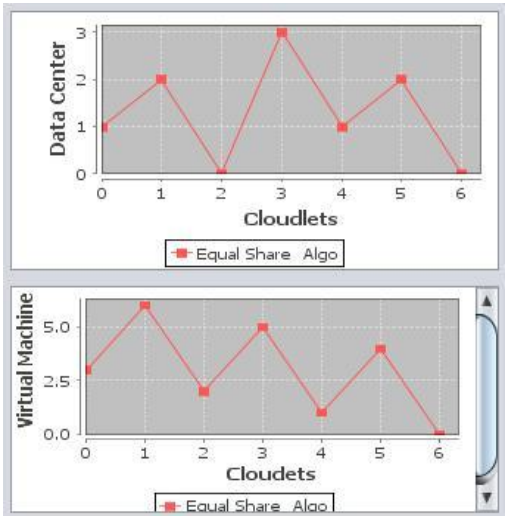

Figure 6. Data Centers and VM Allocation using Equally Load Shared Load Balancing Algorithm

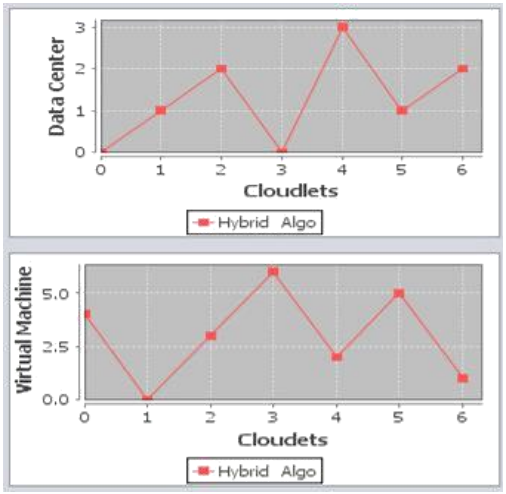

Figure 7. Data Centers and VM Allocation using Hybrid Load Balancing Algorithm

C. Cloudsim Simulator

CloudSim [8] is the many efficient tools you can use with regard to modeling regarding Cloud. During your current lifecycle of a Cloud, CloudSim allows VMs for you to be managed coming from hosts that will inside turn are usually managed by datacenters. CloudSim offers architecture inside four uncomplicated entities.

These types of entities offer consumer to set-up the basic cloud computing environment as well as measure your effectiveness involving fill up balancing algorithms.

Datacenters substance includes the duty of giving Infrastructure level answers for the Cloud Users.

They go about as a home to help a great deal of Host Entities or possibly a considerable measure of occurrences hosts' elements total to enable application to shape the singular Datacenter element. Hosts with Cloud are typically Physical Servers The thought have pre-designed preparing capacities. Host is really mindful with respect to giving Software level SERVICE towards Cloud Users. Hosts have their specific stockpiling and memory. Processing features regarding hosts is usually expressed throughout MIPS (million instructions per second)

IV. CONCLUSION

In this paper, we have simulated three dynamic load balancing algorithms named equally spread Current Execution, Throttled, and hybrid. Here we have taken 10 cloudlets, 4 data centers and 4 VMs for simulation. All three algorithms simulation performed with the help of CloudSim simulator on java language and compare data center service time, allocation of data center, allocation of VM, response time, and, total cost of each data center. As a result, we found that for Hybrid algorithm, data center allocation and cost of each data center is more accurately than two others. As a future work we have two objectives, first, our project is improving efficiency and reduce cost of data centers and VM using new adaptive algorithm, and second, to improve our algorithm for heterogeneous environment such as big data.

V. REFERENCES

[1]. Purohitet (2014), "Load balancing in public cloud by division of cloud based on the geographical location", International Journal of Research in Engineering and Technology

[2]. GaochaoXu, Junjie Pang, and Xiaodong Fu (2013), "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud", Tsinghua Science and Technology.

[3]. K. Nishant, et all, (2012), "Load balancing of nodes in cloud using ant colony optimization", 14th International Conference on Computer Modeling and Simulation (UKSim), Cambridgeshire, United Kingdom.

[4]. Bhatiya Wickremansinghe, Rodrigo N. Calheiros and Dr. Rajkumar Buyya (2010), "CloudAnalyst: A CloudSim- based VisulModeller for Analysing Cloud Computing Environments and Applications", IEEE Computer Society.

[5]. Jaspreet Kaur (2012), "Comparison of load balancing algorithm in a Cloud", International Journal of Engineering Research and Applications (IJERA).

[6]. Kunal Mahurkar, Shraddha Katore and Suraj Bhaisade, Pratikawale (2013), "Reducing Cost of Provisioning in Cloud Computing", International Journal of Advance in Computer Science and Cloud Computing.

[7]. Syed Tauhid Zuheri, Tamanna Shamrin and Rusia Tanbin, Firoj Mahmud (2013), "An Efficient Load Balancing Approach in Cloud Environment by using Round Robin Algorithm", International Journal of Artificial and Mechatronics.

[8]. B. Adler (2012), "Load balancing in the cloud: Tools, tips and Tools, tips and techniques", whitepapers