

**INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH  
TECHNOLOGY**

**CLUSTER-BASED DECENTRALIZED JOB DISPATCHING FOR THE LARGE-  
SCALE CLOUD**

**Er. Rajdeep Kaur\*, Ms. Amanpreet Kaur**

\* Student, M-Tech (CSE) 4th Semester, CSE Department, Global Institute of Management & Emerging Technologies, Amritsar (PUNJAB)

Assistant Professor, CSE Department, Global Institute of Management & Emerging Technologies, Amritsar (PUNJAB)

DOI: 10.5281/zenodo.58554

---

**ABSTRACT**

Cloud computing is characteristically distinct as a kind of computing that relies on distribution computing resources relatively than having restricted server or private device to switch applications. Cloud technology allows for the routine stipulation and denial of resource as and when it is essential, thus ensuring that the intensity of resource accessible is as strongly matched to current demand as possible. The research work of title “A Framework for implementing Load Balancing and Cluster-Based Decentralized dispatching policy across Virtual Machines in Cloud federations” is based on simulation. This research proposed a frame work on behalf of which a workload is distributed among virtual machines available at various User Bases residing at different Data Centers. The workload is dispersed depending on different policies such as Round Robin, Equally Spread Execution Load and Throttled. The work distribution can takes place depending on single policy at a time. The entire globe is divided into six regions i.e. R0, R1, R2, R3, R4 and R5. The single Data Center and User Base can belong to single region only. After placing the Data Center and User Base to a particular region, its configuration is figured out. User grouping in User Bases and Request grouping in Data centers are the two key features added in this work. Simultaneous users from a single user base forms a user grouping and simultaneous requests that a single server can support at a particular time forms a request grouping in data centers. On behalf of all the parameters response time and processing time is calculated in terms of MAX, MIN and AVG. The same is calculated on region basis also. On behalf of type and extent of hardware, its cost is also calculated in term of VM cost and data transfer cost. At last when simulation is over, overall response is generated and documentation can be achieved.

**KEYWORDS:** Cloud Computing, Virtual Machine, Load balancing, Data centre, User bases, Simulation, Resource scheduling, Service level agreement (SLA).

---

**INTRODUCTION**

Cloud Computing is typically defined as a type of computing that relies on sharing computing resources rather than having local server or personal device to handle applications. In Cloud Computing, the word cloud is used as a metaphor for “the Internet”, so the phrase cloud computing means “a type of Internet-based computing”, where different services such as servers, storage and applications are delivered to an organization computers and devices through the internet. One of the key characteristics of cloud computing is the flexibility that it offers and one of the ways that flexibility is offered is through scalability. This refers to the ability of a system to adapt and scale to changes in workload.



Figure 1- Cloud Computing [5]

The cloud makes it possible for a user to access information from anywhere at any time. While a traditional computer setup requires user to be in the same location as your data storage device, the cloud takes away that step. The cloud removes the need for user to be in the same physical location as the hardware that stores your data. Cloud provider can both own and need to buy the amount of storage space user will use, a business can purchase more space or reduce their subscription as their business grows or as they find they need less storage space. This is especially helpful for businesses that cannot afford the same amount of hardware and storage space as a bigger company. Small companies can store their information in the cloud, removing the cost of purchasing and storing memory devices.

## CLOUD COMPUTING MODELS

Cloud Providers offer services that can be grouped into three categories:

### Software as a Service (SaaS)

In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced. On the customer's side, there is no need for upfront investment in servers or software licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted & maintained. Today SaaS is offered by companies such as Google, Salesforce, Microsoft, Zoho, etc.

### Platform as a Service (PaaS)

Here, a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider's infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySQL and PHP), restricted J2EE, Ruby etc. Google's App Engine, Force.com etc are some of the popular PaaS examples.

### Infrastructure as a Service (IaaS)

IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, GoGrid, 3 Tera, etc.

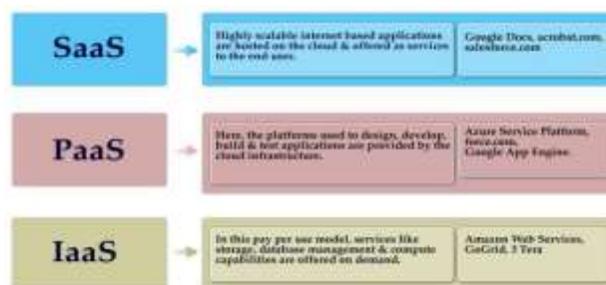


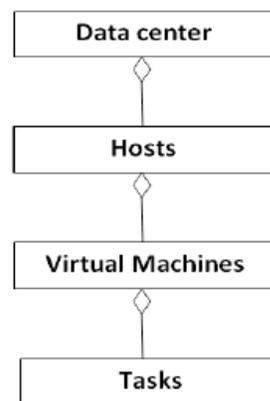
Figure 2- Cloud models [6]

A Virtual Machine (VM) is a software implementation of a computing environment in which an operating system (OS) or program can be installed and run. The Virtual Machine typically emulates a physical computing environment, but requests for CPU, memory, hard disk, network and other hardware resources are managed by a virtualization layer which translates these requests to the underlying physical hardware. VMs are created within a virtualization layer, such as a hypervisor or a virtualization platform that runs on top of a client or server operating system. This operating system is known as the host OS. The virtualization layer can be used to create many individual, isolated VM environments.

### VM LOAD BALANCING

Virtual Machine enables the abstraction of an OS and Application running on it from the hardware. The interior hardware infrastructure services interrelated to the Clouds are modeled in the simulator by a Datacenter element for handling service requests. These requests are application elements sandboxed within VMs, which need to be allocated a share of processing power on Datacenter's host components. Data Center object manages the data center management activities such as VM creation and destruction and does the routing of user requests received from user via the Internet to the VMs. The Data Center Controller, uses a VmLoadBalancer to determine which VM should be assigned the next request for processing. Most common VM load balancer are throttled and active monitoring load balancing algorithms. Throttled Load Balancer maintain a record the state of each Virtual Machine (busy/ideal), if a request arrive concerning the allocation of Virtual Machine, throttled load balancer send the ID of ideal Virtual Machine to the data center controller and data center controller allocates the ideal Virtual Machine. Active Monitoring Load Balancer maintains information about each VMs and the number of requests currently allocated to which VM. When a request to allocate a new VM arrives, it identifies the least loaded VM. If there are more than one, the first identified is selected.

Class diagram of Cloud architecture illustrating relationship between the four basic entities is shown in figure 3. Thus, the object oriented approach of CloudSim can be used to simulate Cloud Computing environment.



*Figure 3- Class diagram of Cloud [8]*

#### Load balancing handles various issues:

- Efficient distribution of user processes on virtual machines.
- Efficient distribution of virtual machines on the physical servers.

#### Load Balancing helps in:

- Improving the performance substantially.
- Having a Reverse up plan in case the system fails even partially.
- Maintenance of system stability.
- Accommodation of future modification.

- Efficient load distribution.
- Cost effectiveness.

## LOAD BALANCING SCHEMES

Load balancing algorithms can be classified into static and dynamic approaches:

### *Static load balancing algorithm*

Static load balancing algorithms assume that a priori information about all the characteristics of the jobs, the computing resources and the communication network are known and provided. Load balancing decisions are made deterministically or probabilistically at compile time and remain constant during runtime. The static approach is attractive because it is simple and requires minimized runtime overhead. However, it has two major disadvantages. Firstly, the workload distribution of many applications cannot be predicted before program execution. Secondly, it assumes that the computing resources and communication network are all known in advance and remain constant. Such an assumption may not apply to a distributed environment. As static approach cannot respond to the dynamic runtime environment, it may lead to load imbalance on some resources and significantly increase the job response time.

### *Dynamic load balancing algorithm*

Dynamic load balancing algorithms attempt to use the runtime state information to make more informative decision in sharing the system load. However, dynamic scheme is used a lot in modern load balancing method due to their robustness and flexibility.

A list of common parameters that can be used to characterize most of dynamic load balancing algorithms are:

### *Adaptive vs. non-adaptive*

If the parameters of the algorithm can change when the algorithm is being run, the algorithm is said to adaptive (to the changes in the environment in which it is running). Otherwise, it is non-adaptive.

### *a) Sender-initiated vs. receiver-initiated*

In a source-initiated algorithm, an over-loaded node starts negotiations with the other nodes for a potential process-migration. If a negotiation is started by an under loaded node, the algorithm is said to be destination-initiated.

### *b) Preemptive vs. non-preemptive*

If a process that has started its execution can be transferred to some other node, then the algorithm is called a preemptive algorithm. If, on the other hand, only those processes that are in the ready queue but have not yet received CPU service could be considered for migration, the algorithm is called a non-preemptive algorithm. [2]

## LOAD BALANCING POLICIES

An algorithm for the load balancing problem can be broadly categorized in terms of four policies. They are:

### *A. Location policy*

It is the policy that affects the finding of a suitable node for migration. The common technique followed here is polling, on a broadcast, random, nearest-neighbor or roster basis.

### *B. Transfer policy*

It is that which determine whether a node is suitable for participating in a process migration. One common technique followed is the threshold policy, where a node participates in a negotiation only when its load is less than (in destination-initiated algorithm) or greater than (in sender-initiated algorithm) a threshold value.

### *C. Selection policy*

It is the policy that deals with the selection of the process to be migrated. The common factors which must be considered are the cost of migration (communication time, memory, computational requirement of the process, etc.) and the expected gain of migration (overall speedup of the system, etc.).

**D. Information policy**

It is that component of the algorithm that decides what, how and when the information regarding the state of the other nodes in the system is gathered and managed. They can be grouped under demand-driven, periodic, or state-change-driven policies.

**LOAD BALANCING ALGORITHMS****A. Round Robin**

It is a type of Static and Decentralized algorithm in nature. In the following algorithm, the processes are divided between all processors. Each process is assigned to the processor in a round robin order using a particular time value. The process allocation order is maintained locally independent of the allocations from remote processors. Though the work load distributions between processors are equal but the job processing time for different processes are not same. So at any point of time some nodes may be heavily loaded and others remain idle. This algorithm is mostly used in web servers. This algorithm is also implemented as Weighed Round Robin Algorithm. In this we assign the weights to the system in a manner for better resource allocation and utilization.

**B. Throttled Load Balancer**

This algorithm is a dynamic load balancing algorithm. It is used for load balancing in the case of the virtual machines to be used. Here we first check the index values of all the virtual machine in the system. The request is sent where load balancer parses a table for the allocation of the resources in the system. It assigns the request to a particular load balancer which passes or responds reverse the request to the requester and updates the allocation policy [9]. After the successful allocation of the system the whole process for the de-allocation of the system also starts. This mechanism provides a greater a higher amount of resource sharing and allocation on a whole in the system resulting in the higher performance and utilization. The throttling threshold maintained generally is 1. It could be modified easily to make the threshold a configurable value.

**C. Active Monitoring Load Balancer**

This load balancing policy attempts to maintain equal workloads on all the available VMs. The algorithm used is quite similar to the throttled case as explained above but with faster and timely checking as well as the accessibility of the resources in the system. The ids for the allocation and de-allocation are specified. The value of the count changes with a new request. It gives the maximum utilization and performance of resources and machines respectively. It is a dynamic allocation algorithm. The important things to consider while developing such algorithm are: estimation and comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones. [4]

**PROBLEM FORMULATION**

Cloud Computing is a novel trend emerging in IT environment with enormous requirements of infrastructure and resources. Load Balancing is a central aspect of cloud computing environment. Competent load balancing design ensures efficient resource utilization by provisioning of resources to cloud user's on-demand basis in pay-as-you-say-manner. Load Balancing may even sustain prioritizing users by applying appropriate scheduling criteria. Cloud load balancing is the process of distributing workloads across numerous computing resources. Cloud load balancing lessen costs associated with document management systems and maximizes availability of resources.

Load balancing sometimes leads to data congestion when it is not done appropriately and it also limited the resources for other machines as well. Congestion may leads to loss of data packets. So, load balancing can be achieved on behalf of various methods according to the type of VM, location of VM, type of data etc. Many parameters should be considered before designing the actual cloud environment so that load balancing can be done appropriately and automatic.

So, there is a need of system which can balance the load among various machines and reduce congestion and response time by increasing an efficiency of a system. So the key target is to intend such a system on simulator will simulate the environment in which the entire globe can be connected together through various machines available at random location and the result formed from the experimental approach will produce better system, better environment for cloud federation.

## METHODOLOGY PROPOSED

Cloud Sim is to be used along with Cloud Analyst. Firstly, CloudSim is installed. For this, LINUX is required because Cloud Sim works on LINUX or the whole scenario can work on windows after generating an EXE file of whole experiment. To propose cloud computing environment by using multiple Data centres, User bases, Virtual machines by placing all on different regions. Load balancing can be implementing by using various techniques and results will be generated on behalf of various parameters like total load, response time, cost of machine etc. CloudSim is a toolkit (library) for simulation of Cloud computing scenarios. It provides basic classes for describing data centers, virtual machines, applications, users, computational resources, and policies for management of diverse parts of the system (e.g., scheduling and provisioning). MAC layer Protocol: TDMA, CDMA, IEEE Mac 802.x etc.

### Steps followed for methodology

In order to simulate the whole scenario multiple steps are followed. The set-up is to be done in obvious manner to achieve objectives of the research. The setup holds User bases, Data centres, Virtual machines, Memory, Load balancing policies etc. All these are implemented step after step and will exertion in a synchronized way to achieve the desired results.

### Steps of Simulation

#### Step 1: Configure Simulation

##### 1.1 Main configuration

**1.1.1 Simulation duration-** Simulation duration can be given in minutes, hours and days.

**1.1.2 User Bases-** This includes information like UserBase name, Region, Request per user per hour, Data Size per request (bytes), Peak hour start, Peak hour end, Avg. peak user, avg. off peak user.

**1.1.3** Application deployment configuration

**1.1.4** Service Broker Policy.

**1.1.4.1** Closest data centre.

**1.1.4.2** Optimize response time.

**1.1.4.3** Reconfigure dynamically with load balancing.

**1.1.5 Data Center-** This includes information like data Center ID, No. of Virtual machines belong to particular data center, Image size, Memory, Bandwidth.

**1.2 Data Center Configuration-** Under this step, information that plays a vital role is as follows: Name of Data center, Region of data center, Architecture, Operating System, Virtual Machine Manager/ Monitor, Cost of VM (\$/hr), Memory cost (\$/sec), Storage cost (\$/sec), Data transfer cost (\$/GB), Physical Hardware unit.

**1.2.1 Physical Hardware details of Data Center-** This will hold information like: Data Centre ID, Memory (in MB), Storage (in MB), Available Bandwidth (in KB), No. of Processors, Speed of processor (Clock cycles), VM policy.

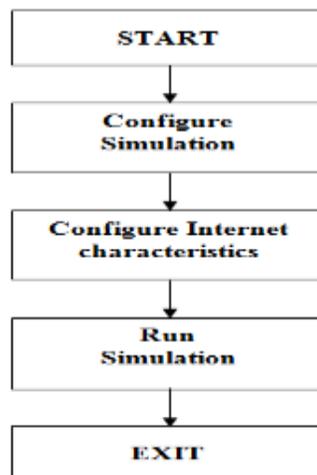


Figure 4- Simulation Process

**1.3** Advanced- This includes:

- 1.3.1** User grouping factor in User bases.
- 1.3.2** Request grouping factor in Data centers.
- 1.3.3** Executable instruction length per request.
- 1.3.4** Load balancing policy
  - 1.3.4.1** Round Robin
  - 1.3.4.2** Equal Spread current execution load.
  - 1.3.4.3** Throttled.

**Step 2: Configure Internet Characteristics**

**2.1 Delay Matrix-** Delay between regions in milliseconds.

**2.2 Bandwidth matrix-** Available bandwidth between regions in Mbps.

**Step 3: Run simulation**

This will generate the overall response time summary as output which includes multiple tables and graphs.

**Step 4: Exit**

**SIMULATION**

Figure 5 is related to the region boundaries. In this figure various regions are shown which split the entire globe. Here, six regions are shown as R0, R1, R2, R3, R4 and R5. Also, three data centers are shown as DC1, DC2 and DC3 and five User Bases are defined as UB1, UB2, UB3, UB4 and UB5. User bases UB1, UB2, UB3, UB4 and UB5 belongs to region R0, R1, R2, R4 and R5 respectively. The Data centre DC1, DC2 and DC3 belongs to regions R0, R2 and R4 respectively. On the left hand side of the screen all steps discussed in the methodology are shown i.e. Configure simulation, Define Internet characteristics, Run Simulation and Exit.



*Figure 5- Regions boundaries*

*A. Main configuration*

In this part, two major terms are considered, the first term is user bases and second one is application deployment configuration. Before all, simulation time is to be mentioned in minutes. This time is the actual duration on which simulation will work.



*Figure 6- Configure Simulation*

Data center configuration is shown in figure 7, it is an arrangement that deals with the details of each data center which include its name, region etc.

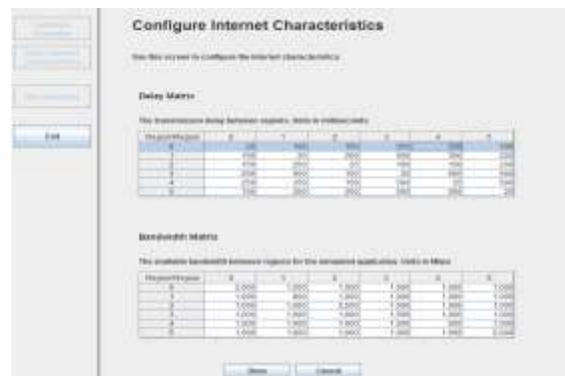


**Figure 7- Data Centre Configuration**

Figure 8 shows advanced configuration from where scheduling algorithm can be selected to perform various jobs on cloud accordingly



**Figure- 8 Advanced Configuration**



**Figure 9 – Configuring Internet characteristics**

Delay matrix and bandwidth matrix both comes under internet characteristics. Both can be configured in the step shown in figure 9.



**Figure 10- Running simulation**

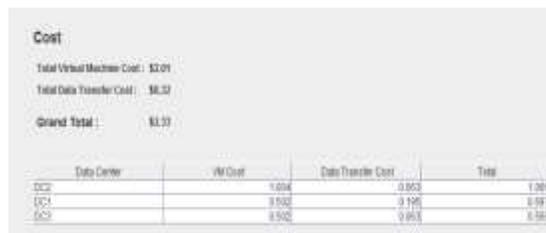
When user click on the left button labeled with “Run Simulation”, the actual simulation performed. This will show the maximum, minimum and the average time of each data center under which various jobs runs. Figure 11 shows the overall summary report, in this figure response time is shown. This time is the used base hourly average response time. This output shows both graphs and tables for Max, Min and Avg.



**Figure 11- Overall response time summary**



**Figure 12- User base hourly avg. Response time**



**Figure 13- Cost**

Hourly average response time of user bases is shown in figure 12. This time is based on the hardware configuration of data centre. This also involves three variations i.e. max, Min and average. This is actually data centre request servicing time. Figure 13 shows the cost of computing. This included three data centres as DC1, DC2 and DC3. There in each data centre cost of Virtual machine and cost of data transfer is considered. On behalf of both these parameters, total cost is calculated.

## CONCLUSION

The cloud makes it possible for a user to access information from anywhere at any time. The cloud removes the need for user to be in the same physical location as the hardware that stores data. Small companies can store their information in the cloud, removing the cost of purchasing and storing memory devices. Cloud computing provides huge benefits. This research is based upon the simulation through which some data centers and user bases can be deployed on different regions and after setting some parameters on behalf of memory, machine, storage, bandwidth etc the simulation is performed.

Any number of user bases and data centers can be deployed. The broker policy is also involved in the research as the service of cloud is also dependent on SLA (service Level Agreement). So, closest data center, Optimize response time and dynamically load balancing broker policies are involved. For each simulation only one type can be considered at a time. For load balancing round robin, throttled and equally spreading execution load are the three algorithms which are involved. Only one algorithm can take charge at a particular time depending on the situation. The simulation runs on behalf of algorithms and parameters under which some statistics are entered. On behalf of the algorithm's performance and metrics, the simulation is performed.

The result of simulation will produce graphs and tables which show the overall response time of data centres, user bases on behalf of regions. The response time of all the entities is purely based on the distribution method and algorithm used.

## REFERENCES

- [1] Byunseok Kang, Hyunseung Choo, "cluster-based decentralized job dispatching for the large-scale cloud", Kang and Choo EURASIP Journal on Wireless Communications and Networking, Springer, 2016.
- [2] Chandrasekaran K., "Load Balancing of Virtual Machine Resources in Cloud Using Genetic Algorithm", Elsevier Publications, pp (156-168), 2013.
- [3] Walter Cerroni, "Network performance of multiple virtual machine live migration in cloud federations", Cerroni Journal of Internet Services and Applications, Springer, 2015.
- [4] B.Poornima, Dr.T.Rajendran, "Improving Cloud Security by Enhanced Has be Using Hybrid Encryption Scheme", International Journal of Advanced Research in Computer Science and Software Engineering, 2014.
- [5] Soumya Ray, Ajanta De Sarkar, "Execution analysis of load balancing algorithms in cloud computing environment", International Journal on Cloud Computing: Services and Architecture, 2012.
- [6] [https://www.google.co.in/search?q=cloud+computing&newwindow=1&biw=1366&bih=598&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjv6d7TscrLAhUJUZQKHxW9C30Q\\_AUICCG#imgrc=3SRVYhIjSc\\_d6M%3A](https://www.google.co.in/search?q=cloud+computing&newwindow=1&biw=1366&bih=598&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjv6d7TscrLAhUJUZQKHxW9C30Q_AUICCG#imgrc=3SRVYhIjSc_d6M%3A)
- [7] <http://www.thbs.com/thbs-insights/cloud-computing-overview>.
- [8] Pankaj Sareen, "Cloud Computing: Types, Architecture, Applications, Concerns, Virtualization and Role of IT Governance in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, 2014.
- [9] Qian Zhang, Hong Liang and Yongshan Xing, "A Parallel Task scheduling Algorithm Based on Fuzzy Clustering in Cloud Computing Environment", International Journal of Machine Learning and Computing, 2014.
- [10] Nusrat Pasha, Dr. Amit Agarwa, "Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment", International Journal of Advanced Research in Computer Science and Software Engineering, 2014.
- [11] Reena Panwar, Bhawna Mallick, "A Comparative Study of Load Balancing Algorithms in Cloud Computing", International Journal of Computer Applications, 2015.

- [12] Rajkumar Somani, Jyotsana Ojha, "A Hybrid Approach for VM Load Balancing in Cloud Using CloudSim", International Journal of Science, Engineering and Technology Research (IJSETR), 2014.
- [13] Bhupendra Panchal and R.K.Kapoor, "Performance Enhancement of Cloud Computing using Clustering", IJCSNS International Journal of Computer Science and Network Security, 2014.
- [14] Volker Fusenig and Ajush Sharma, "Security Architecture for Cloud Networking", Fujitsu 2012.
- [15] Mayanka Katyayal, Atul Mishra, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment", International Journal of Distributed and Cloud Computing, 2013.