Cloud Performance and Load Balancing Algorithm
1 Cloud Performance and Load Balancing Algorithm

In cloud computing paradigm, application and data are stored in data center of cloud provider which is located in diverse geographical locations. Presently developed countries such as USA, UK and countries from European Union are selected as preferred locations for data center by the cloud providers. These locations have been selected considering the peace and stability existing in these countries, so that any loss due to terrorist activities can be avoided. Data center locations of some of the prominent cloud providers have been depicted in table 4.1.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Region</th>
<th>Data Center Location</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>USA</td>
</tr>
<tr>
<td>1.</td>
<td>Amazon Web Service</td>
<td>East(Northern Region), US West(Northern California), Brazil</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ireland, Netherlands, Germany</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Japan, China, Singapore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ireland, Netherlands</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Singapore, Hong Kong</td>
</tr>
<tr>
<td>3.</td>
<td>Google</td>
<td>USA (Quilicura, Oklahoma, Lenoir, The Dalles, Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Finland, Belgium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Singapore, Taiwan, Hong Kong</td>
</tr>
</tbody>
</table>

Each data center consists of a number of nodes; these nodes are in the order of hundreds. Data center offers basic on demand storage and computation over the internet. Provisions of these
computational resources are in the form of VM, which is an abstract unit of computational in cloud. These VM differs in configuration such as memory, CPU and bandwidth etc. Due to the dynamic nature of cloud environments, diversity of user’s requests and time dependency of load, cloud centers must provide expected quality of service at widely varying loads (Xiong & Perros, 2009; Baker et al., 2011).

When thinking about cloud computing, specifically cloud infrastructure, performance matters. Cloud performance is measured at the point of application and is the sum of network performance, application performance, geographic proximity and cloud infrastructure performance. Achieving high-speed delivery of applications in the cloud is a multifaceted challenge that requires a holistic approach and an end-to-end view of the application request-response path. In cloud computing network performance both within the cloud and out of the cloud is having the dependency on I/O access speed between the compute layer and the multiple tiers of data stores. Network and application performance is not in the control of cloud provider, while for the infrastructure performance it is vital to understand the factors which can affect the access time of user and can contribute to the overall performance. In cloud computing, application performance and network performance plays pivotal role in response time reduction. Response time will be more in cloud computing when comparing it to the traditional based system. Some of the important work which is already carried out in this area has been discussed in the upcoming section.

1.1 Performance Factors Determination

Monitoring the cloud performance is important for both the cloud user and the cloud provider in real time and analyzing the historical performance of the cloud. A large number of applications running in public cloud require huge and powerful resources. Applications known as “big data” consist of workload such as digital media collection, virtual worlds, simulation traces, data obtained from scientific instruments, and enterprise business databases are resource hungry application. Response time for these applications is very important, lacking which leads to unpleasant experience to the user. Applications hosted on public cloud should be checked for its performance i.e. response time and processing time so that these factors are within the tolerance limit. Before provisioning the more resources, it is important that resources available are used at its optimum level to avoid further investment in these resources.
Existing cloud monitoring tools are either provided by the cloud provider himself (such as cloudwatch by Amazon) or user can also use these tools provided by the third party such as revealuptime and cloudstatus.

Number of researcher has worked in this area, [2] carried out the extensive study related to cloud performance by conducting the test from multiple locations. Amazon, Google, Salesforce, Rackspace, and Terremark are some of the cloud where test were carried out. Author customized the requirement and conducted the test with the help of test agent and the result gathered. The report measures service response, network performance, CPU, and internal I/O. In addition CloudHarmony staff conducts test on the prescribed benchmarks for specific scenarios such as CPU performance, storage, I/O, memory I/O or video encoding[3]. Useful information that does not affect the safety of the company, is also published on their blog.

Factors before selection of the cloud provider or partner to be considered are highlighted by [4]. These factors are related to the performance of the virtual machine, bandwidth that is offered by the cloud provider so that the need of resource hungry application can be met. Other factors include I/O bottleneck and if the website is observing the heavy load, in that case whether the resources can be allocated and releases when this workload subsides.

Some of the works which demonstrates the factors which significantly impact the performance has been carried out by the authors such as Vineetha (2012) who has suggested the importance of cloud performance monitoring tool considering cloud provider and cloud user perspective. Author has monitored the response time, CPU usage and storage usage etc. Reporting for these factors in peak and off peak hours was also suggested. Avram(2010) measured the performance of 05 dominant cloud providers (Google, Rackspace, Salesforce.com, Amazon and terremark) for the website and measured the response time and latency time of each of these cloud providers. In other work (Harzog, 2010) suggested a new approach for infrastructure management to determine the performance in virtualized environment by monitoring infrastructure response time. Bannerman (2010) suggested the performance factors i.e. latency, response time and execution time as one of the greatest risk in cloud computing. Virtual instrument (2012) emphasized to monitor the cloud performance by monitoring for its factor like response time and latency by use of automated tool to enable the cloud user to know about the exact status of the cloud.
1.2 Performance Monitoring Applications

To monitor the performance in the cloud computing, a number of automated performance monitoring (APM) tools from various vendors are available which provide the monitoring in real time as well as on historical data. Majority of them enable the user to monitor the details of their cloud using wide variety of devices such as smart phones to laptop devices. Some of these tools have been defined as in table 4.2.

*Table 1.2: Provider and factor for monitoring*

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Provider</th>
<th>APM Tool</th>
<th>Factors monitored</th>
</tr>
</thead>
</table>
| 1.    | Netinst  | Observe reporting system | • End user page response time  
• Transactions processed  
• Network error, latency & utilization |
|       | (http://www.netinst.com/-products/observer-reporting-server/index.php) | | |
| 2.    | Copper Egg | RevealUptime | • URL, port, & site latency, response time, uptime, health.  
• Worldwide real time data collection coverage, analytics, alarms.  
• Troubleshooting, multi-user access |
|       | (http://copperegg.com/revealuptimewebsite-monitoring/) | | |
| 3.    | Hyperic  | Cloudstatus | • Monitor service availability, response time, latency, and throughput  
• Provides real-time reports  
• Application availability and performance |
|       | (http://www.hyperic.com/products/cloud-status-monitoring) | | |
| 4.    | CA technologies | CA virtual assurance | • Infrastructure response time, latency time  
• Monitor the |
Hence, from the above discussion and table it can be inferred that response time, data processing time, throughput and uptime are important factors which plays pivotal role in cloud and hence, required to be monitored to determine the cloud performance.

### 1.3 Load Balancing in Cloud Computing

In a scenario of limited servers available at data center, if the request submitted are high than the capacity of the data center, its overall performance degrades. In such cases load balancer is used.
to improve the performance of data center. Load balancing is a technique to distribute load among multiple entities such as CPUs, disk drives, server or any other type of device. The goal of load balancing is primarily to obtain much greater utilization of resources. Load balancing can be provided either through hardware or software.

1.3.1 **Hardware Load Balancing**

Load balancing can be provided through the specialized devices such as a multilayer switch that can route the packets to the destination or the cluster. Hardware based load balancing is complex in configuration & maintenance, and not suitable for hosted environment.

1.3.2 **Software Based Load Balancing**

Load balancing can also be achieved through the software either using operating system or as an add-on application. Software based load balancing is simple to deploy and have the performance similar to that of hardware based load balancing. Some software based load balancing includes those bundles with Microsoft azure or Linux and add on such as PM proxy.

1.3.3 **Load Balancer**

Load balancer manages the traffic flow between various servers. Load Balancer is placed between the server and the client and distributes the load among the available servers depending upon the algorithm of the Load balancer. Load balancer is not only improves the response time of cloud applications but also ensures the optimum utilization of the resources.

1.4 **Major Load Balancer Algorithm**

To improve the performance in different types of cloud a number of Load balancing techniques are used. These Load balancing techniques are selected as per the load received. Depending on the load and the distance from the user, applications are directed to the data center to optimize the performance.

Considering the importance of load balancing, major operating system such as Microsoft windows and Linux are providing the load balancer as build-in capability or it can be implemented as add-on software which is having the comprehensive option. These load
balancing techniques can be implemented at the user end or at data center end. When implemented at user end at that time it is known as service broker policy.

1.4.1 **Service Broker Policy**

In cloud computing load management is required so that the request submitted should take minimum time and to be routed to the appropriate type of cloudlet. In case of one data center is overloaded at that point of time, provision should exist to divert the traffic to other data center. In brief management is required:

- For optimum use of VMs within the data center
- Routing the traffic among various different data center available as per Service level agreement.

Traffic routing between user base and DC is performed by a service broker that decides which data center is to be used for a particular user base. Three types of service broker policies which are supporting three different routing policies are (Wickremasinghe, 2009):

1.4.1.1 **Closest Data Center Based Routing (CDC)**

CDC is based on the quickest path available from the user base to the data center with minimum latency. Service broker will transmit the traffic to DC with minimum transmission delay.

1.4.1.2 **Optimum Response Time (ORT)**

ORT Service broker policy actively monitor the performance of all the data center and directs the traffic to a datacenter which estimates to give the best response time to the end user at the time it queried.

1.4.1.3 **Reconfigure Dynamically with Load (RDL)**

This service policy is an extension to CDC and deals with the scaling of resources as per the load it receives. In case of overload it increases the number of VMs in the data center and reduces the VMs in case of less load.
1.4.2 **Data Center Controller**

DC controller is one of the most important entities in cloud. It manages the creation and destruction of VMs and performs the request routing from user base to the concerned VM. It consists of VM Load balancer to determine which VM should be assigned to the cloudlet for processing. Currently the three VM Load balancing techniques existing in cloud analyst are (Wickremasinghe, 2009):

1.4.3 **Round Robin**

Round robin performs the basic type of load balancing and functions simply by providing the list of IP address of cloudlet. It allocates first IP address to the first requester then second IP address to the second requestor for a fixed interval of time known as time slice. If the request is unable to finish within the given slice time, it will have to wait for the next cycle to get it turn for execution. This will continue till submitted tasks are not completed.

1.4.3.1 **Active Monitoring Load Balancer**

This load balancer find outs the active VM and also to event out the active task at any point of time.

1.4.3.2 **Throttled Load balancer**

This load balancing technique ensures that only a per-defined number of internet cloudlets are allocated to a single VM at any point of time. If more groups are presents in the data center than the number of available VM than some of the requests have to be queued until the next VM is available.

1.4.4 **Load Balancing Algorithm**

A number of load balancing algorithms existing which are distributing the load among the data center. Each of them has their own functionality. Some of the major load balancing algorithms has been discussed as follows:
1.4.5 **HAproxy**

HAproxy stands for High Availability proxy and used for loads balancing. It supports different operating system for instance Linux 2.4, 2.6, Solaris 8, 9, and Open BSD 3.1 to current etc. User can communicate at port 80 using application have TCP and HTTP support. HAproxy is a free fast and reliable solution and its new version 1.4 supports flexibility for mission-critical application. While the other version is 1.3 that offers content switching and extreme loads. HAproxy is used for very big sites supporting millions hits, 24x7 availability. HAproxy has become the defacto standard for the industry and coming as default for the cloud environment. Its single event driven model considerably reduces the context switch time and memory usage. Since, multi-process and multi thread are having the limitations associated with memory. To prevent the task from waiting indefinitely due to big task from waiting indefinitely due to big task it uses round robin method so that small task do not wait for indefinite time. HAproxy involves the several techniques found in OS architecture to achieve the maximal performance such as:

a) **Event Checker:** Used to check any event that is taking place.

b) **Single Buffering:** Supports single buffering that avoids copy for reading and writing in multiple devices.

c) **MRU:** To create new session it uses MRU for immediate memory allocation from the pool that favours hot cache regions over cold cache regions.

d) **Tree Based Storage:** To achieve the fast access of storage it uses elastic binary tree that has the cost of $O(\log (N))$.

e) **Usage of User space:** Reduces the usage of expensive system calls by using user space by default.

f) **Data Rate:** It provides the data rate in MB/s or sometimes in Mbps. To achieve the high data rate it uses large object for the session as they reduces the overhead. Data rate is reverse of the session created.

g) **Security:** Right now, it is being used in number of fortune 500 companies due to its...
reliability feature. These companies are using it for simpler to complex problems. HAproxy 1.3 and 1.4 is evolutionary and support mission-critical usage.

1.4.6  **Scheduling**

Allocating the processor to a task for execution is known as scheduling. The main aim of scheduling is to keep the CPU busy and increase the throughput and reduce the waiting time. To schedule the task in cloud computing a number of scheduling algorithm has been defined. Major scheduling algorithm has been discussed in upcoming subsection.

1.4.6.1  **Join the Shortest Queue (JSQ)**

In the JSQ the delivery controller dispatches the jobs to the processor with the least number of jobs. It follows the greedy algorithm to achieve the optimum result. Even though JSQ is not optimal, yet it provides better performance in comparison to those having higher complexity. It causes less overhead because only processors with their allocation are required to be maintained so that the processor with shortest queue can be selected. The entire request for processor is directed through the centralized load balancer while all the response also directed from the centralized processor. Therefore, load balancers can easily track the status of current request and response to each processor.

1.4.6.2  **Task Duplication Based Scheduling for Heterogeneous Network**

This is designed to avoid transfer cost inflicted by transfer of result between interdependent tasks. This feature is useful in reducing estimated completion time, comes under the cluster and duplication based algorithm. First refers that it grows the task and assigned to the cluster, while the latter refers to the action of duplicating some parent task on other resources.

1.4.6.3  **Opportunistic Load Balancing**

Tasks are allocated to the next machine which is available in arbitrary order. It is quite simple and does not maintain EET. Since, it is not considering EET therefore, quite fast. But the main drawback is that it may lead to high makespan.
1.4.6.4 Minimum Execution Time

This algorithm first finds out the best resources considering the execution time. On the arrival of the task, it is allocated to the best resource whether resource is available or not at that time. The main disadvantage is that it causes large imbalances across the resources.

1.4.6.5 Min/EETK-Percent Best (KPB)

It considers mapping of task to the machine by calculating the K-percent, which is calculated using $Km/100 \[100/m \leq k \leq 100\]$. The scheduling heuristics assigns the task to the resources with the best MinECT within a subset of superior resources.

1.5 Cloud Simulation Tools

Simulation software is based on process of real phenomenon with a set of mathematical formulae’s. This software provides the simulated environment that is similar to real world environment. Simulation software is designed in a manner so that result should be close to real world. Discrete event and continuous simulation are the two categories of simulation package. Discrete simulations are used to model statistical events such as customer arriving in bank; while continuous simulation is used in wide variety of physical phenomena like ballistic trajectory, human respiration, and radio frequency data communication etc. Cloud environment can also be simulated with the help of simulation software. Major cloud simulators that can be used for cloud simulations are as follows:

- iCanCloud
- OPNET
- CloudAnalyst

1.5.1 iCanCloud

iCanCloud is a simulation platform aimed to model and simulate cloud computing system. The main objective of iCanCloud is to report the trade-off between performance and cost on a particular hardware. This tool is highly suitable from basic users to developers of large distributed applications. The main features of iCanCloud include the following.
a) Existing and non-existing cloud computing architectures can be modeled and simulated.

b) Cloud broker policies can be tested due to flexible hypervisor.

c) VM can simulate uni-core/multi-core systems.

d) A wide variety of storage options can be selected ranging from local storage systems, remote storage systems like Network file system (NFS), parallel file systems and redundant array of independent disk (RAID) systems.

e) Provides easy, user friendly graphical user interface (GUI) to ease the generation and customization of large distributed models.

f) Provides POSIX-based API and an adapted MPI, Library.

g) Provides feature to add new component in repository of iCanCloud to achieve more functionality.

1.5.2 OPNET

OPNET is a simulator build on NS2. OPNET also released OPNET IT Guru Academic edition, where limited cloud simulation can be performed (http://www.opnet.com). The other tool is Automated Performance Monitoring (APM) which is comprehensive simulator for cloud and highly suitable from migration to monitoring of the cloud (http://www.opnet.com/solutions/application_performance/). OPNET modeler can simulate wide varieties of network which are linked to each other. OPNET provides the following features:

- Allows building and analyzing models.

- Provide option to select wide varieties of components.

- Permit network entity to run on individual workstations.

- Routers, Switches and web servers can be used in modeling.

OPNET consists of the following modules:

**Flow Analysis:** Provides the capacity failure analysis and extensive reporting.
**NetDoctor:** Validates the configuration based on policy.

**Discrete Event Simulation (DES):** Provides graph, web reports, charts for all time-varying network performance metrics.

**Application Characterization Environment (ACE):** Provides application visualization, Diagnosis, analysis.

OPNET supports the following simulation technologies.

- Discrete event simulation (DES).
- Flow analysis.
- ACE Quickstart.
- Hybrid simulation/ Environment within the DES

### 1.5.3 Cloud Analyst

Cloud analyst is graphical simulator built on cloudSim. It can simulate the behavior of large scaled application on the internet. Cloud Analyst is GUI based and frees the modeler from programming complexity and to concentrate on modeling. It also enables the modeler to execute the simulations repeatedly by modifying the parameters. Graphical output enables to read and analyze the output quickly and easily (Wickremasinghe, Calheiros & Buyya, 2010).

Main features of CloudAnalyst are as follows:

**Ease of use:** Graphical method of simulation provides ease of use.

**Configuration:** A number of configurations can be set considering applications, data center and broker service policies etc.

**Graphical output:** Output provided in the graphical form makes the output easier to analyze.

**Repeatability:** Same simulation can be repeated by modifying the parameters used previously.

**Ease of Extension:** It features can be extended to accommodate more features. This feature will enable to keep this simulator up-to-date that can cater the latest demand.
**Technology:** Java, JavaSwing, CloudSim, SimJava are some of the underlying framework that has been used by the simulator.

### 1.6 Simulation Environment

To carry out this experiment, cloud analyzer simulator has been used that is built on CloudSim. Results obtained from CloudSim environment are very close to real environment. It is not only used by researcher but also by corporate to perform simulation and to obtain the result for analysis. This Experiment has been carried out considering the following parameters.

#### 1.6.1 Region Division

We have divided the whole earth into five regions to match with the classification of cloud analyzer. Region divisions and the countries within these regions are as follows:

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Region</th>
<th>Countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$R_0$</td>
<td>USA</td>
</tr>
<tr>
<td>2.</td>
<td>$R_1$</td>
<td>Countries of North America</td>
</tr>
<tr>
<td>3.</td>
<td>$R_2$</td>
<td>Countries of European union</td>
</tr>
<tr>
<td>4.</td>
<td>$R_3$</td>
<td>Countries of Asia like china, India</td>
</tr>
<tr>
<td>5.</td>
<td>$R_4$</td>
<td>Country from Africa</td>
</tr>
<tr>
<td>6.</td>
<td>$R_5$</td>
<td>Australia</td>
</tr>
</tbody>
</table>

#### 1.6.2 Data Center Configuration

Total 05 data center have been considered for the simulation environment. Architecture for each data center is given in table 4.4.

*Table 1.4: Architecture of Data Center*
In our experiment, we have considered the constant pricing for storage and memory for all the data centers. Each Data center will have three hardware units as per the details shown in Table 4.5.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Memory size(Mb)</th>
<th>Storage Details(Mb)</th>
<th>Available BW</th>
<th>No. of Processor</th>
<th>Processor Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2048000</td>
<td>10000000000</td>
<td>10000000</td>
<td>20</td>
<td>10000</td>
</tr>
<tr>
<td>2.</td>
<td>2048000</td>
<td>10000000000</td>
<td>10000000</td>
<td>4</td>
<td>10000</td>
</tr>
<tr>
<td>3.</td>
<td>2048000</td>
<td>10000000000</td>
<td>10000000</td>
<td>4</td>
<td>10000</td>
</tr>
</tbody>
</table>

1.6.3 **User Base**

Simulation is carried out considering 30 user base (UB) located in different regions to access a particular data center. Each user base has the following configuration (Table 4.6):

<table>
<thead>
<tr>
<th>Configurations Factors</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>User grouping factor in each user base</td>
<td>1000</td>
</tr>
<tr>
<td>(Equivalent to number of simultaneous users from a single UB)</td>
<td></td>
</tr>
</tbody>
</table>
Request grouping factor in Data Center
(Equivalent to number of simultaneous request a single application server instance can support) 100

Executable Instruction length per request
(These user bases are not homogeneously distributed to evaluate the effect of response time of data center as per the user base) 1000

1.6.4 **Application Deployment**

Each server has the architecture as shown in Table 4.7:

<table>
<thead>
<tr>
<th>VM</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory (each VM)</td>
<td>512 MB</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>1000</td>
</tr>
</tbody>
</table>

1.6.5 **Service Broker Policy**

Three service policies which have been evaluated are:

- Closest Data Center
- Optimize response time
- Reconfigure dynamically with load

1.6.6 **Load Balancing Policy**

Load balancing policy on which experiment is carried out.

- Round Robin
- Equally spread current execution
- Throttled