Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 12, December -2017

Assessment of Overall Response Time amongst major service broker policies: Closest Data Center First, Reconfigure Dynamically and Optimize Response Time

Neha Mathur

Department of Computer Science, M.B.M Engineering College, J.N.V.U Jodhpur

Abstract — *Cloud computing is the fastest growing internet-based technology which has influenced academia, research and global business industry. Users from all over the globe are demanding various services of the cloud at a rapid rate leading to busty workloads at data centers. Load balancing at data center level is the need of the hour. The service broker acts as intermediary between cloud users and cloud service providers and selects the most appropriate data center for each user request using underlying service broker policies. In this work, major service broker policies: Closest Data Center First, Reconfigure Dynamically and Optimize Response Time are compared and the response time delivered by them is analyzed. CloudAnalyst is used as simulator for implementing these policies.*

Keywords-*Load Balancing, Response Time, Cloud Analyst, Service Broker, Cloud*

I. INTRODUCTION

Cloud Computing is a computing paradigm that provides dynamically scalable and virtualized resource as a service over the Internet [1]. So, users are able to access the resources, such as applications and data, from the cloud anywhere and anytime on demand on a pay-as-you-use basis. With the advent of Internet technologies, users throughout the world demand services of the cloud at an accelerated rate leading to a data deluge. This increases load on cloud data centers and virtual machines. Efficient policies are needed to balance load for effective functioning of clouds. Load balancing enables enterprises to handle workload demands by allocating resources among multiple computers, networks or servers. To evaluate the performance of load balancing policies performance metrics need to be considered.

Response time is the time interval between sending a request and receiving its response. It should be minimized to boost the overall performance. In this work, assessment of response time for service broker policies is done.

II. LOAD BALANCING IN CLOUD

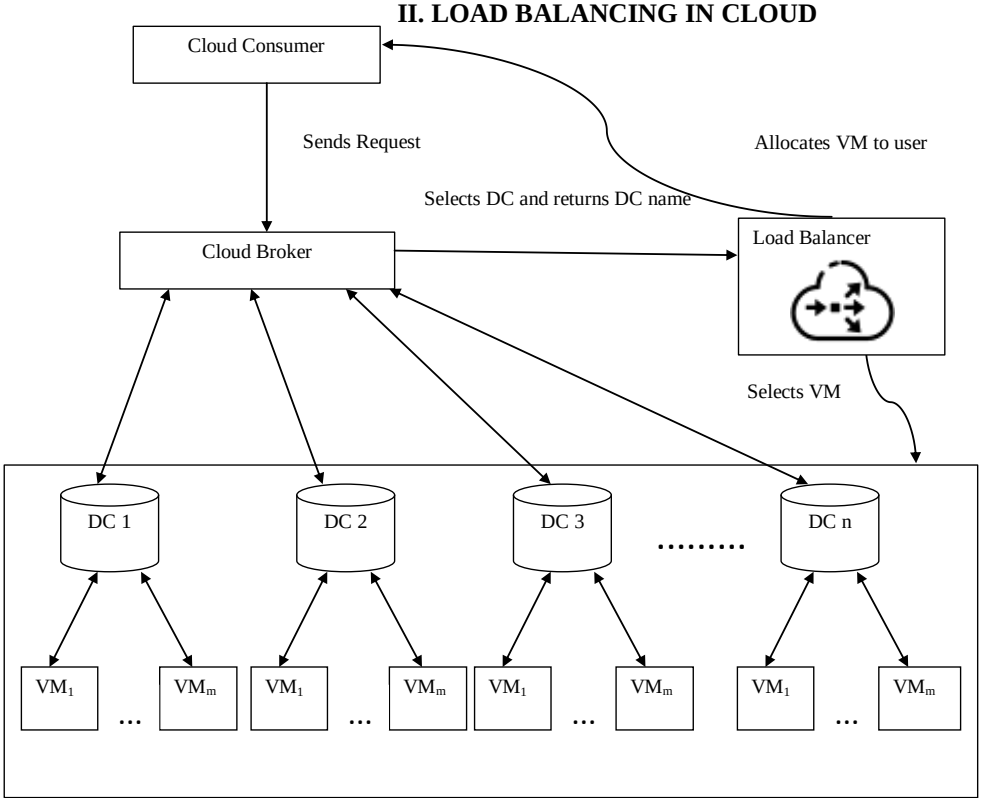

Figure 1. Architecture for Cloud Load Balancing

Load balancing refers to the act of distributing the amount of work that a node/server has to do between two or more nodes/servers in order to get more work done in the same amount of time so that all users get served faster. Cloud load balancing means load balancing process carried out in cloud computing environment. In Cloud load balancing the workloads are distributed across multiple cloud servers to enhance the overall system performance (increase throughput and reduce response time). Efficient load balancing techniques plays a major role in cloud computing by allocating requests to computing resources efficiently to prevent under/over-allocation of Virtual Machines (VMs) and improve the response time to clients.

A. Architecture of Cloud Load Balancing

The accumulation of requests at a particular interval of time leads to busty workloads. The main feature of cloud computing is the adoption of *virtualization*, in which virtual machines (VM) are running on top of the available hardware to satisfy the users need and demand. Therefore, managing VMs is an important aspect to be considered to keep the whole cloud running efficiently, which is carried out by the hypervisor. The selection of the VM for a particular workload is done by the load balancer. Figure1 depicts the architecture for load balancing in cloud.

The load balancer distributes the load in a way that ensures no VM is swamped with requests at one time. Above this level, another abstraction level called the service broker, acts as an intermediary between the users and the cloud service providers. The service broker utilizes existing service broker policies to route user requests to the most appropriate data center. Therefore, the optimal response time of a particular request and the efficient utilization of the datacenters are governed through a proper data center selection policy.

III. SERVICE BROKER POLICIES IN CLOUD

The service broker uses a policy to select appropriate data center to execute a cloudlet/task. Each policy has some advantages and some drawbacks.

3.1 Closest Datacenter First (CDF)

In this policy, the broker chooses the shortest path from the user to the data center (DC) based on the network latency only. The service broker maintains an index table of all data centers indexed by their regions. A proximity list of regions is maintained which orders the remaining regions in the order of lowest network latency first when calculated from the given region. The Service Broker picks the first data center located at the earliest/highest region in the proximity list. If more than one data center is located in a region, one is selected randomly.

Drawback: This may result in overloading the closest data center and its communication channel because it does not consider the channel bandwidth.

3.2 Optimized Response Time (ORT)

The service broker maintains an index of all data centers available. The closest data center (in terms of latency) is identified. The broker iterates through the list of all data centers and estimates the current response time at each data center by querying the last recorded processing time. If this time is recorded before a predefined threshold, the processing time for that data center is reset to 0. This means the data center has been idle for duration of at least the threshold time. The network delay is added to the value arrived at by above steps. If the least estimated response time is for the closest data center, the service broker selects the closest data center. Else, it picks either the closest data center or the data center with the least response time with a 50:50 chance (i.e. load balanced 50:50).

Drawback: The optimized response time routing policy, the broker chooses the best path based on network latency and DC workloads, to achieve the best response time based on the last job response time. This will be generalized as the status of any other DC. If any DC with a current load of zero, it will not be selected unless a certain amount of time is waited (i.e. Cool-Off-Time). This could leave the DC idle with no jobs assigned even if it was on the closest (i.e. least latency) and highest available bandwidth network path.

3.3 Reconfigure Dynamically (RD)

The dynamically reconfigure routing is similar to the closest datacenter first routing, but the broker scales the application deployment based on the load it is facing. The service broker maintains a list of all data centers and another list with the best response time recorded so far for each data center. The closest datacenter first policy is used to identify the destination datacenter and updates the best response time records if the current response time is better than previous. This routing algorithm is the same as one of the other policies but in addition to the above; service broker should run a separate thread to monitor the current response times of all the data centers. If the current response time is increasing and is greater than the best response time for the data center plus some pre-defined threshold, the dynamic service broker notifies the data center to increase the VM count by creating more VMs. If the current response time is decreasing steadily for a pre-defined threshold of time the dynamic service broker notifies the data center to reduce the VM count by releasing VMs.

Drawback: The dynamically reconfigure routing is not efficient if the number of regions and the number of data centers are limited; because it scales the applications deployment based on the current load .

IV. METHODOLOGY AND EXPERIMENTAL SETUP

4.1 Methodology

Social networking applications are one of the complex large-scale applications on the internet that can benefit from cloud technology. A popular social networking site, Facebook has over 189 million registered users worldwide. In June 2017, the approximate distribution of its user base across the globe was the following: North America: 263 million users; South America: 370 million users; Europe: 343 million users; Asia: 736 million users; Africa: 160 million users; and Oceania: 20 million users. [4]

In this work, the behavior of social networking application is modeled and CloudAnalyst is used to evaluate performance of various service broker policies. [3].

4.2 Experimental Setup

To evaluate the various service broker policies and compare it with the proposed service broker policy, CloudAnalyst is used as the simulator. A detailed description of the CloudAnalyst Simulator is presented in Annexure -B. NetBeans IDE 8.0.2 is used as development environment. In the development environment, the programming language Java is used for coding. The IDE and simulator are setup on a machine which is configured with Intel (R) Core™ i3 CPU M 330 @ 2.16 GHz processor and 2 GBDDR3 RAM, and installed with Windows 7 operating system.

The six main regions of the world are represented by six user bases [3]. The parameters of six user bases are defined in Table 1. a similar hypothetical application at 1/100th of the scale of Facebook is used for the simulation. Each user base is contained within a single time zone for the sake of simplicity. The experiment resulted in to an assumption that most users use the application in the evenings after work for about 2 hours. So, peak hours for each region are 7-9 pm local time. It is also assumed that during the peak time 5% of the registered users are online simultaneously and during the off-peak hours only one tenth of that number of users is on line.

In terms of the cost of hosting applications in a Cloud, a pricing plan which closely follows the actual pricing plan of Amazon EC2 is assumed. [5]

Table 1: Uses Bases used in the experiment

User Base	Region	Region #	Time Zone	Peak Hours (GMT)	Simultaneous Online Users during Peak Hours	Simultaneous Online Users during Off-Peak Hours
UB0	N. America	0	GMT -6.00	13:00-15:00	130,000	13,000
UB1	S. America	1	GMT -4.00	15:00-17:00	115,000	11,500
UB2	Europe	2	GMT +1.00	18:00-20:00	172,000	17,200
UB3	Asia	3	GMT +6.00	01:00-03:00	368,000	36,800
UB4	Africa	4	GMT +2.00	21:00-23:00	80,000	8,000
UB5	Ocenia	5	GMT +10.00	09:00-11:00	9,750	975

All simulation parameters used in the experiment are showed in Table 2. VMs used to host applications in the experiment have a size of 100MB. Available bandwidth and RAM memory of VMs is 10MB and 1GB respectively. Simulated hosts are characterized by Linux operating system, x86 architecture and VM monitor. Physical Machines (Data Centers) have 2 GB of RAM and 100GB of storage. There is a capacity power of 11000 MIPS in each CPU. For scheduling resources to VMs a time-shared policy is used. User grouping factor is 1000, and request grouping factor is 10. 250 instructions are required to execute each user's request. User bases used in the experiments are displayed in Table 1.

Table 2: Simulation Parameters

Parameter	**Value**
Datacenter Architecture	x86
OS	Linux
Virtual Machine Manager VMM	Xen
Cost per VM/Hr $	$0.10
Data Transfer Cost $/GB	$0.10
Physical HW Units (Machines) per Datacenter	2
No. Of Processors Per Machine	4
Processor Speed	10000 MIPS
VM Policy	TIME SHARED
VM Level Load Balancing Policy	Throttled

There are three major VM load balancing polices: round –robin, throttled and active monitoring. The throttled load balancing policy yields better results (determined experimentally).

4.3 Simulation Scenarios:

Three scenarios are considered in this work. All the three existing service broker policies are applied in all three scenarios for evaluation and performance comparison of existing and proposed policy. Parameters used for comparative analysis of the policies are data center request processing time and response time

4.3.1. Classic Scenario

This is simplest one which consists of modeling the case where a single Cloud Data Center is used to host the social network application in each region.100 virtual machines are allocated in the application at the data center.

4.3.2 Homogeneous Scenario

This configuration consists of modeling the case where two cloud data centers are used to host the social network application in each region. Each of the two data centers has 50 VMs dedicated to the application are used.

4.3.3 Heterogeneous Scenario

In this three data centers have different amount of virtual machines, each with 20, 30 and 50 VMs.

V. OBSERVATIONS AND RESULT ANALYSIS

5.1 Classic Scenario:

In classic scenario, there is one data center in each region. Table 1 shows the results of the experiment for classic scenario.

Table 1: Results of Classic Scenario

Service Broker Policies	Overall Response Time (ms)
Closest Data Center First	421.16
Optimized Response Time	307.39
Reconfigure Dynamically	423.07

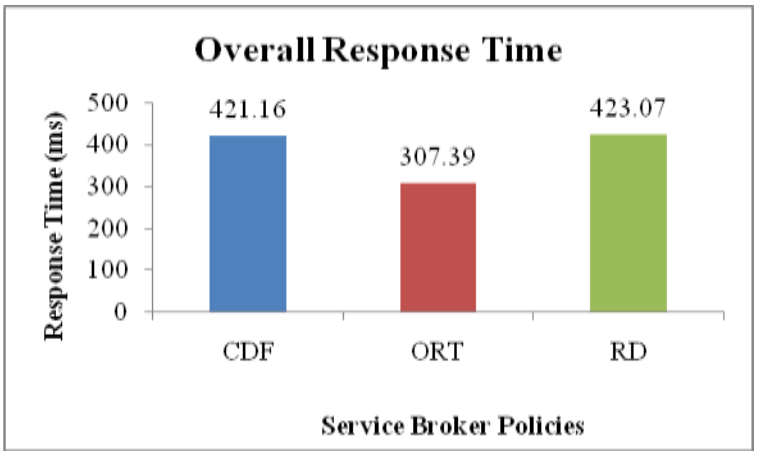

Figure 2: Results of Classic Scenario

Figure 2 displays the trend of overall response time of all policies. The overall response time of CDF policy is 421.16 ms, of ORT is 307.39 ms and of RD policy is 423.07 ms. The ORT policy results in least response time. In CDF and RD policies, the response time is higher than ORT because requests are distributed among nearest/regional data centers which leads to queuing in times of heavy load. In ORT, the requests are sent to the data center which is available no request is directly migrated and no time is spent waiting in any queue.

5.2 Homogeneous Scenario:

In homogeneous scenario, there are two data centers in each region. Table 2 shows the results of the experiment for homogeneous scenario.

Table 2: Results of Homogeneous Scenario

Service Broker Policies	Overall Response Time (ms)
Closest Data Center First	234.84
Optimized Response Time	225.51
Reconfigure Dynamically	420.23

The overall RT of CDF policy is 234.84ms, of ORT policy is 225.51 ms and of RD policy is 420.23 ms.

Figure 3 displays the overall response time of all policies. The CDF policy sends all requests of a region to either of its 2 DCs randomly. The RD policy sends requests to DC with optimum response time. But when load increases, the DC with optimum expected response time gets fully loaded leading to DC contention. So, RT increases, and DC

processing time increases during heavy traffic. The ORT policy efficiently sends requests from requesting region to an available DC in a region with minimum latency and least expected response time. So, ORT policy provides least response time.

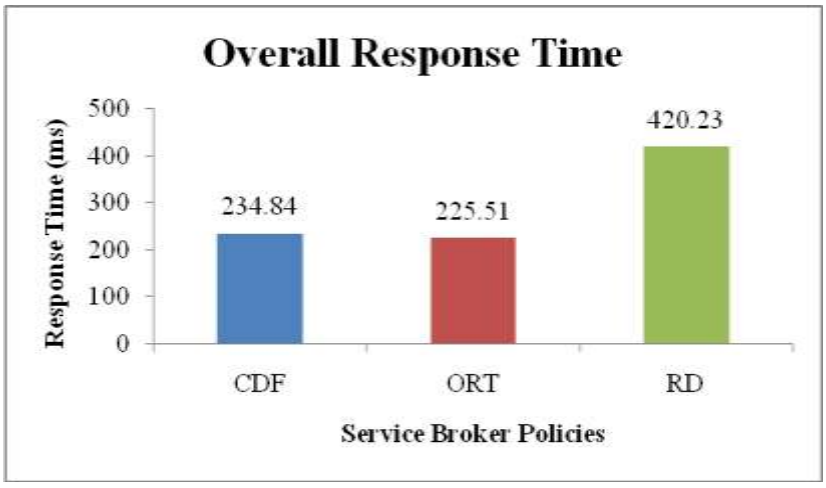

Figure 3: Results of Response Time in Homogeneous Scenario

5.3 Heterogeneous Scenario:

In heterogeneous scenario, there are three data centers in each region. Table 3 shows the results of the experiment for classic scenario.

Table 2: Results of Heterogeneous Scenario

Service Broker Policies	Overall Response Time (ms)
Closest Data Center First	173.13
Optimized Response Time	184.75
Reconfigure Dynamically	467.25

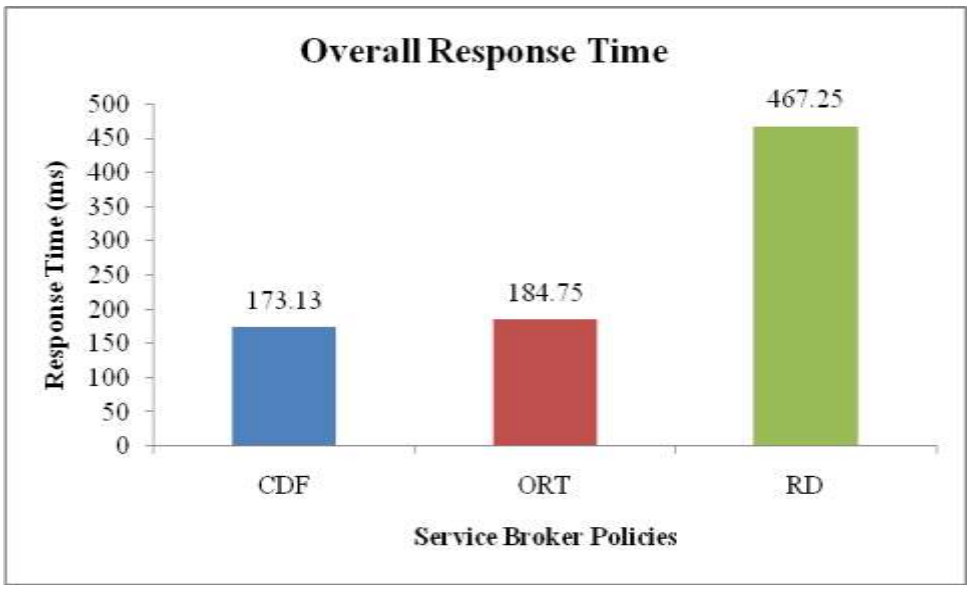

Figure 4: Results of Response Time of Heterogeneous Scenario

Figure 4 displays the overall response time of all policies.

The observation of overall RT in heterogeneous scenario of CDF, ORT and RD service broker policies is 173.13 ms, 184.65 and 467.25 ms respectively.

In RD policy, DC is selected based on previously recorded best response time. The selected one might be fully loaded currently. So, it results in highest response time. ORT policy displays least response time due to selection of DC based on minimal latency and least expected response time.

5.4 Overall Response-Time Comparison Scenario-wise

Table 4: Overall Response Time Scenario-wise

Scenario	Response Time (ms)		
	CDF	ORT	RD
Classic	421.16	307.39	423.07
Homogeneous	234.84	225.51	420.23
Heterogeneous	173.13	184.75	467.25

The overall RT of CDF policy in classic scenario is 421.16 ms, in homogeneous scenario is 234.84 ms and in heterogeneous scenarios is 173.13 ms
The overall RTs of RD policy in classic, homogenous and heterogeneous scenario are 307.39 ms, 225.51 ms and 184.75 ms respectively.
The overall RTs of RD policy in classic, homogenous and heterogeneous scenario are 423.07 ms, 420.23 ms and 467.25 ms respectively.

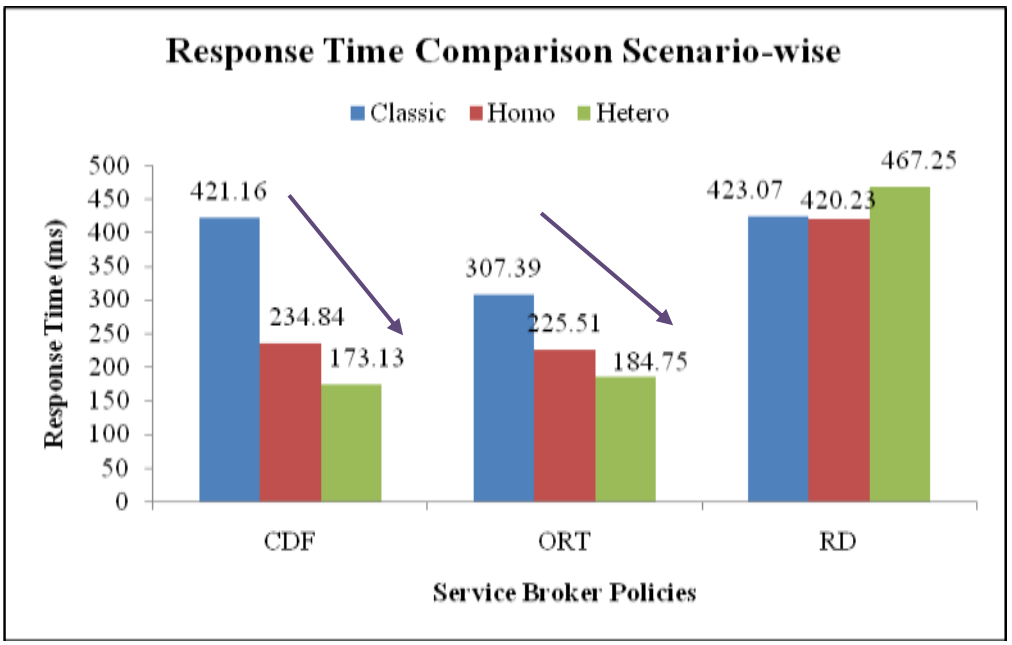

Figure 5: Scenario-wise Response Time Comparison

In CDF Policy, the overall RT decreases as the number of DC's increases. The overall RT in homogeneous scenario is 44.23% lower than RT in classic scenario. The overall RT in heterogeneous scenario is 58.89% lower than classic scenario and 26.27% lower homogenous scenario.
As the number of DCs increase, the capacity of system increases .So, the time required to process request decreases.
In ORT Policy, the overall RT decreases as the number of DC's increases. The overall RT in homogeneous scenario is 26.61% lower than RT in classic scenario. The overall RT in heterogeneous scenario is 39.89% lower than classic scenario and 18.07% lower homogenous scenario.
As the number of DCs increase, the capacity of system increases .So, the time required to process request decreases.

In RD Policy, the overall RT in homogenous scenario is 0.67 % lower than in classic scenario.The overall RT in heterogeneous scenario is 11.18 % higher than homogenous scenario and 10.44% higher than classic scenario.
The reconfigure dynamically policy chooses the DC based on proximity. During heavy traffic, the regional DC's get fully loaded leading to slower response times and turnaround times.
The reason for higher response time in heterogeneous scenario is the heterogeneous distribution of VMs. In classic scenario, each DC has 100 VMs. In homogeneous scenario, each region has 2 DCs and each DC has 50 VMs. In heterogeneous scenario, each region has 3 DCs with 20, 30 and 50 VMs respectively. Out of the three DCs, a DC is randomly selected based on proximity. Since the no. of VMs is not same, it leads to higher response time.

VI. CONCLUSION

In classic scenario, each region has one datacenter with 100VMs. The ORT service broker policy performs **37.01%** better than reconfigure dynamically policy and **37.63%** better than closest datacenter first policy. In homogeneous scenario, each region has two datacenters with 50 VMs in each. The ORT service broker policy performs **46.33%** better than reconfigure dynamically policy and **4.13%** better than closest datacenter first policy. In heterogeneous scenario, each region has three datacenters with 20, 30, 50 VMs respectively. The ORT service broker policy performs **152.9%** better than reconfigure dynamically policy and **6.28%** better than closest datacenter first policy. It can be concluded that the ORT policy delivers least response time in all scenarios. Therefore, applications that are response time sensitive such as interactive applications must use optimum response time service broker policy for load balancing at data center level.

REFERENCES

[1] R. Buyya et al, "Cloud computing and emerging IT platforms: Vision, hype, and reality for Delivering computing as the 5th utility", *Future Generation Computer Systems*, 2009.
[2] F. Liu et al, "NIST Cloud Computing Reference Architecture", *Special Publication (NIST SP) - 500-292*, September 2011.
[3] B. Wickremasinghe, R.N. Calheiros and R. Buyya, "CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications", *International Conference on Advanced Information Networking and Applications (AINA), IEEE*, April 2010.
[4] www.internetworldstats.com/facebook.htm [Last Accessed On: 12-12-17]
[5] https://aws.amazon.com/ [Last Accessed On: 02-11-17]

