# Performance Analysis of Web Application Deployed on Cloud using Cloud Analyst Simulator

**Mayanka Gaur, Manmohan Sharma**

*Department of Computer Science and Engineering, Mody University of Science and Technology, Lakshmangarh, Sikar (distt.), Rajasthan, India*

Email: gaur.mac1305@gmail.com, manmohan.manu@gmail.com

***Abstract---*** *Cloud computing is the field of computing that is growing rapidly day-by-day both in academic and industry in order to fulfill requirements of end-users. Cloud computing in real sense is performing any task by making use of services that are provided by cloud providers. Cloud computing enables a wide range of users to approach Distributed, Scalable, Virtualized assests over the net. The concept of cloud computing has changed the area of distributed and parallel computing. Cloud Computing is a part of Distributed Computing. Cloud Computing intended to influence next creation data centers and allows application service providers to hold data center capabilities for deploying applications depending on user's Quality of Service (QoS) requirements. One major issue that the web application developer or designer faces before deploying his or her application on cloud is meeting quality of service (QoS) with efficient performance based on the user needs. Above query can be dealed with analyzing the performance of application in a massively distributed environment through detailed comprehensive studies done through simulation techniques. CloudAnalyst is one of the simulation tools that extends GridSim and CloudSim techniques and is used by application developers or designers, to study the nature of large-scaled internet applications in cloud environment. In our paper, firstly we are presenting an overview of some of the cloud simulation tools along with the description of cloud simulator that we used (i.e. CloudAnalyst), then making some estimates or conclusions based on the simulation experiments we performed and their results using CloudAnalyst tool.*

***Keywords-*** *Cloud Computing, Information Technology, Modeling, Simulation, Cloud Simulators, Web Application, Virtual Machine, Performance Analysis, CloudSim, CloudAnalyst.*

## I. INTRODUCTION

Now-a-days progress in computer science and Internet technology made computing on real cloud a high demand. Cloud computing in real sense is performing any task by making use of services that are provided by cloud providers. Cloud computing is the field of computing that is growing rapidly day-by-day both in academic and industry in order to fulfill requirements of end-users. Cloud server is a combination of data storage server and computation server.

Cloud Computing is a part of distributed computing. Aim of cloud computing is to provide distributed, virtualized and flexible resources as services to users. Cloud computing includes services that are provided both by service provider and data centers. Cloud computing involves distributed and grid computing theories. Cloud computing has made the computing as a quality of practical use. Cloud Computing shares the server memory, data and applications simultaneously with multiple users. Cloud computing supports reliable, secure, fault tolerant, sustainable and scalable services. It not only provides physical hardware resources but also provides platform, data and applications to multiple end-users simultaneously. Cloud Computing provides on demand service model and pay-as-you-go service model to consumers.

Cloud Computing provides Infrastructure (IaaS), Platform (PaaS) and Software or Application (SaaS or AaaS) as utilities to cloud consumers or end-users. It not only supports storage services but also provides hosting of web applications on real cloud. Earlier, while designing a web application it's deployment and hosting was main concern or main issue. But with cloud infrastructure it is possible to solve above issue more economically and more responsively. Overall study of above dispute in a heavily distributed environment is very difficult. So, to study such a dynamic environment again and again in a controlled manner application developers or designers uses simulation tool. CloudAnalyst is one of the simulation tools that extends GridSim and CloudSim and is used by application developers, to study the nature of large-scaled internet applications in cloud environment.

## II. RELATED WORK

Simulating something requires that the model should be developed first. A model representing the system itself has some features or characteristics that it possess. These features are simulated by some simulation technique to check the behavior of the system for it's trait affirmation. Simulation is the process of finding out the behavior of system or application at some instance of time during it's decapitation. In case of cloud computing surroundings, applications are simulated using some simulation tool. But one question crop up why there is a need to simulate these cloud applications ?

Before deploying applications on real cloud, developers or designers need to perform timely repeatable and controllable methodologies in order to examine the application behavior for decreasing complexity of application and for quality assurance. This can only be done through cloud simulators. Cloud simulators were developed for analyzing performance of cloud computing environments. These simulators enable analysts to analyze system behavior by focusing on quality issues of specific component under different scenarios. With the use of simulation tool analyst can study and analyze the behavior of dynamic and massively distributed environment in a controlled and reproducible manner.

In previous years, hosting and deployment of cloud applications was main concern. But now with cloud infrastructure and services provided by cloud providers it is possible to solve above problem or issue more economically and flexibly. On one hand, cloud service providers provide large scaled computing infrastructure at cheaper price. On the other hand, cloud service providers provide large scaled software systems (like civil networking sites, e-commerce applications, etc) which can be deployed on infrastructure of cheaper price so as to minimize the cost and improve service quality to end-users. Bringing these two hands together means infrastructure of low cost and software systems above infrastructure of cheaper price will not only minimize cost and improve quality but also have an impact on net benefit.

A comprehensive study of such a dynamic and massively distributed environment in real world is very difficult. So best approach is to use some simulation tool for studying the performance of various dispersed arrangements. There are various cloud simulators that the application developers or designers can use according to their necessity. CloudAnalyst is one of the simulation tools to simulate large-scaled applications on cloud in order to study the behavior of such applications under various deployment configurations. CloudAnalyst is discussed later in detail.

Some of the various cloud simulation tools :

*A. GridSim [4]*

Peer-to-peer or Grid Computing solves many problems in science, engineering, and commerce. Grid Computing consists of millions of heterogeneous resources spread across various organizations or domain, etc. To simulate such a large scaled distributed systems of heterogeneous resources, thousands of users, and to study extensibility of arrangements, discoveries, effectiveness of resource allotment approaches, GridSim simulation tool is used. It was developed by R. Buyya. It evaluates performance of real large scaled distributed environments (i.e. Grid Systems and P2P networks) in a repeatable and controlled manner. It is a java based toolkit. It supports modeling and simulation of heterogeneous grid resources and users across multiple organizations. It supports modeling and simulation of multiple applications and creation of their tasks. It supports mapping and managing of these application tasks to resources.

Salient features of the GridSim :-

- It allows modeling of heterogeneous assests.
- Assests can be modeled operating under space-or time-shared mode.
- Assests can be placed in any time zone.
- Assests can be booked for advance reservation.
- Applications with contrast parallel application models can be simulated.
- Application assignment can be contrary and they can be CPU or I/O intensive.
- There is no limit on the number of application jobs that can be submitted to an assest.
- It enables the simulation of workload traces taken from real supercomputers.
- It allots incoming jobs based on space or time shared mode.
- It provides clear and precise interfaces for implementing different resource allocation algorithms.

One disadvantage of GridSim tool is that it does not explicitly define any specific application model.

*B. CloudSim [3,14, 15]*

CloudSim was developed in the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the Computer Science and Software Engineering Department of the University of Melbourne. It supports seamless modeling, simulation and experimentation on designing cloud computing infrastructures. It is a self-contained platform. It is used to model data centers, service brokers, scheduling and allocation policies of large-scaled cloud computing platforms. CloudSim uses Java language and is built on top of GridSim framework. It provides features for modeling and creation of virtual machines or engines in a data center.

The main components of CloudSim are Datacenters, Virtual Machines (VM) and Cloudlets. Datacenter has one or more than one VMs. Each VM accords with many cloudlets which are the units of cloud service.

Main features of CloudSim toolkit :-

- backing for modeling and simulation of large scale Cloud computing data centers, virtualized server hosts, energy-aware computational assests, data center network topologies and message-passing applications, federated clouds.
- backing for dynamic insertion of simulation components, end and begin of simulation.
- backing for user-defined policies for allocation of hosts and resources to virtual machines.

New features of CloudSim :-

- Supplies modeling and simulation of large scale Cloud Computing surroundings, including Data centers on a single physical computing node.
- An autonomous platform for modeling Clouds, Service Brokers, provisioning, and allotment approaches.
- Supplies simulation of network connections among the simulated arrangement components.
- Facility for simulation of league cloud surroundings that inter-networks assests from both private and public domains, a feature critical for research studies related to Cloud-Bursts and automatic application scaling.

Advantages of using CloudSim for initial performance testing include :-

- Time effectiveness : It requires very less effort and time to implement Cloud based application provisioning test environment.
- Flexibility and applicability : Developers can model and test the performance of their application services in heterogeneous Cloud environments (Amazon EC2, Microsoft Azure) with little programming and deployment effort.

Disadvantages of CloudSim :-

- Any two CloudSim entities (hosts, data centers, cloud brokers, etc) cannot be mapped to the same network node.
- Current CloudSim cannot supply both the Power Model as well as Network Model at the same time.
- The network elements in current CloudSim do not supply power aware simulation.
- The simulation of moving to another place does not take into account the network expenses.
- CloudSim consists of no graphical user interface.

### *C. CloudAnalyst [1, 2, 3]*

Wickremasinghe and Buyya introduced a new tool "CloudAnalyst" at CLOUDS Laboratory, University of Melbourne with a new approach to simulate large-scaled applications on cloud in order to study the behavior of such applications under various deployment configurations. Application designers (or developers) would get benefit from such a study while identifying the optimal or best configuration for deploying their application on cloud. CloudAnalyst extends GridSim and CloudSim to study the behavior of large scaled Internet applications in cloud environment. CloudAnalyst seperates simulation experiment from programming task. Results of simulation are analyzed fastly, easily and more conveniently by generating graphical output of simulation experiment.

Features or advantages of CloudAnalyst are :-

- Ease of use.
- A high amount of Configurability and Flexibility.
- Graphical Output.
- Repeatability.
- Ease of extension.

Java, JavaSwing, CloudSim and SimJava technologies are used in CloudAnalyst. A detailed description of CloudAnalyst is given later in this paper.

### *D. Green Cloud [15, 16]*

GreenCloud is a packet-level simulator that uses the Network Simulator 2 (NS2) libraries for energy-aware data centers. It supplies modeling of servers, switches, links for communication and their energy consumption. In other words, we can say that GreenCloud models the energy consumed by data centers (such as Servers, Network switches and Communication links). GreenCloud can also be used to (1) monitor (2) allocate resources (3) schedule workload (4) optimize communication protocols and network infrastructures.

Features of GreenCloud :-

- Mainly focuses on cloud networking and energy awareness.
- Imitation of CPU, Memory, Storage and Networking assests.
- Independent energy models for each type of assest.
- Supply virtualization and VM movement to another place. Network-aware assest allotment.
- Full TCP/IP implementation.
- User friendly GUI.
- Open Source.

The only disadvantage of Green Cloud simulator is that it models and simulates only small data centers due to very large simulation time and high memory requirements.

*E. iCanCloud [14, 15,16]*

iCanCloud is a cloud simulator which is based on SIMCAN. iCanCloud is a software simulation framework for large storage networks. iCanCloud can forecast the accommodation between price paid and performance of a particular application in a specific hardware in order to inform the users about the costs involved. It focuses on policies which charge users in a pay-as-you-go manner. iCanCloud has a full graphical user interface from which experiments can be modeled and runned, but existing systems can only be modeled manually. It also allows parallel execution of one experiment over several machines.

Features of iCanCloud :-

- Conducts large experiments.
- Ability to integrate any cloud broker policy.
- Reproduction of instance types.
- User friendly GUI.

Disadvantages of iCanCloud are firstly, only Cost per Performance (C/P) modeling of cloud computing environments is simulated or validated and secondly, it models and simulates only EC2 (Elastic Compute Cloud 2) environments.

*F. MDCSim [5]*

MDCSim is an event based simulator developed at the Pennsylvania State University(University in State College, Pennsylvania). It models unique hardware characteristics of different elements of a data center such as Servers, Links and Switches which are collected from contrast dealers and allows estimation of power consumption. MDCSim is the most outstanding tool to be used as it has low simulation overhead and moreover its network package maintains a data center topology in the form of directed graph [Dr. Pawan Kumar & Gaganjot Kaur]. It uses Java language.

Features of MDCSim :-

- Flexibility
    - The number of tiers can be changed.
    - The scheduling methods can be optimized.
    - The type of interconnection used can be altered.
    - The communication mechanisms used can be altered.
- Easy to modify layers without affecting other layers.
- Scalability
    - Simulation tests are possible with large nodes across multiple clusters.

Disadvantage of MDCSim is that it does not support TCP/IP (or Network Protocols).

*G. NetworkCloudSim [7]*

NetworkCloudSim is an extension of CloudSim as a simulation framework which supports generalized applications such as high performance computing applications, workflows and e-commerce[Buyya , 2009]. NetworkCloudSim uses Network Topology class. NetworkCloudSim allows for modeling of Cloud data centers utilizing bandwidth sharing and latencies to enable scalable and fast simulations. NetworkCloudSim structure supports designing of the real cloud data centers and mapping contrast methods, Information of NetworkCloudSim is used to simulate latency in network traffic of CloudSim.

NetworkCloudSim does not support TCP/IP network protocols. It uses Java language.

In the next section, we are describing simulation tool that we used for performance analysis of web application deployed on cloud.

## III. CLOUD ANALYST

Cloud Analayst was developed by Wickremasinghe and R. Buyya at CLOUDS Laboratory, University of Melbourne, Australia with a new approach to simulate large-scaled internet applications on real cloud in order to study their behavior under various deployment senarios. Application designers would get benefit from such a study while identifying the optimal configuration for deploying their application on cloud.

In previous years, hosting and deployment of cloud applications was main concern. But now with cloud infrastructure and services provided by cloud providers it is possible to solve above problem or issue more economically and flexibly. On one hand, cloud service providers provide large scaled computing infrastructure at cheaper price. On the other hand, cloud service providers provide large scaled software systems (like civil networking sites, e-commerce applications, etc) which can be deployed on infrastructure of cheaper price so as to minimize the cost and improve service quality to end-users. Bringing these two hands together means infrastructure of low cost and software systems above infrastructure of cheaper price will not only minimize cost and improve quality but also have an impact on net benefit.

A comprehensive study of such a dynamic and massively distributed environment in real world is very hard to do. So best approach is to use some simulation tool for studying the performance of various dispersed arrangements. There are various cloud simulators that the application developers or designers can use according to their need. CloudAnalyst is a simulation tools to simulate large-scaled applications on cloud in order to study the behavior of such applications under various deployment configurations. CloudAnalyst extends GridSim and CloudSim to study the behavior of large scaled Internet applications in cloud environment. CloudAnalyst seperates simulation experiment from programming task. This feature of CloudAnalyst will enable user to quickly setup simulations and summarize results graphically using it's GUI component.

CloudAnalyst is built directly on top of CloudSim Framework and CloudSim is built on top of GridSim Framework as shown in Fig. 1

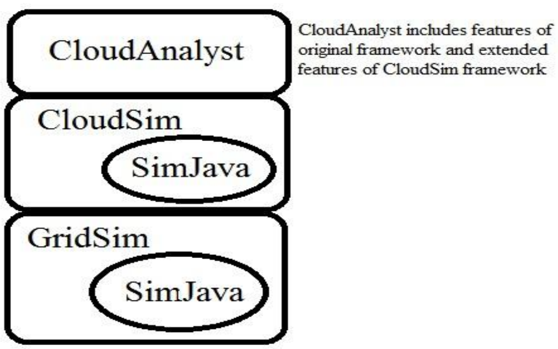


Fig. 1 : Pictorial view of CloudAnalyst Framework

*A. Architecture*

CloudAnalyst is built on top of CloudSim inheriting advantages of original framework, extending CloudSim functionalities or capabilities and introducing it's own concept or functionalities as shown in Fig. 2

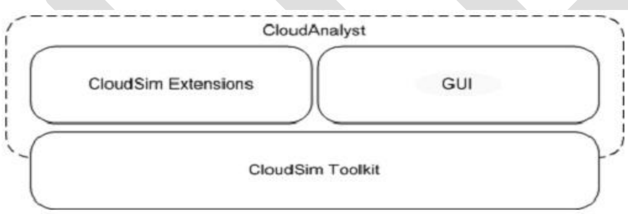


Fig. 2 : CloudAnalyst Architecture

*B. Main Components*

1) Region

The world is divided into 6 'Regions' coinciding within 6 main continents in world. User bases and data centers belong to one of these regions.

2) Internet

It is an abstraction of real world Internet. It implements only features that are required for simulation. It models Internet traffic, routing around the globe by defining transmission latency and data transfer delays.

3) Cloud Application Service Broker

Traffic routing between user bases and data centers is controlled by service broker. Service Broker decides which data center should service the requests from each user base. Current version of CloudAnalyst implements 3 types of service brokers :-

**Service proximity based routing :** In this, the service broker will route the user traffic to the nearest data center based on the transmission latency.

**Performance optimized routing :** In this, the service broker monitors the performance of all data centers and directs the traffic to that data center which it estimates to provide the best response time to the end user at the time it is queried.

**Dynamically reconfiguring router :** It is an extension to service proximity based routing with similar routing logic. Service broker has the responsibility of scaling the application deployment based on the load it faces. This is done by increasing or decreasing the number of VMs allocated in the data center dynamically.

4) User Base

It models a group of users that is considered as a single unit in simulation. User base's responsibility is to generate traffic for simulation. A single user base means thousands of users.

5) Internet Cloudlet

Internet Cloudlet is grouping of user requests. A single internet cloudlet means a bundle of user requests. Single Internet Cloudlet is configurable in CloudAnalyst.

6) Data Center Controller

It is an important entity in cloudAnalyst. A single data center controller is mapped to a single CloudSim. Data center objects and manages activities like VM creation, routing or allocating user request to VMs, VM destruction.

7) VmLoadBalancer

DataCenterController uses a VmLoadBalancer to determine which VM should be assigned the next Cloudlet for processing. Currently there are 3 VmLoadBalancers implementing 3 load balancing policies :

**Round-robin Load Balancer :** It uses simple round-robin algorithm to allocate VMs

**Active Monitoring Load Balancer :** It balances the tasks among available VM's.

**Throttled Load Balancer :** It ensures that only a pre-defined number of cloudlets are allocated to single VM at any given time. But if more requests are present and if these requests are more than the number of available VM's at a particular data center then some of the requests have to be queued until the next VM becomes available.

8) Graphical User Interface (GUI)

It is implemented as a set of screens that enable user to firstly define simulation parameters. Secondly, save and load simulation configurations. Thirdly, execute simulations with option of cancelling a simulation once started. Fourthly, view and save the results of simulation with graphical outputs.

*C. Functionalities*

Functionalities of CloudSim that are extended or inherited as it is in CloudAnalyst are as follows :

- Simulating Data Center hardware
- Simulating Virtual machine specification creation and destruction
- Management of virtual machine
- Allocation of hardware resources for operation of virtual machines based on different policies.
- Simulating execution of user requests on virtual machines.

Above functionalities of CloudSim are directly used in CloudAnalyst.

CloudAnalyst functionalities that are required to be built on top of CloudSim are :-

**1) Application users :-** Autonomous entities act as a traffic generators and their behavior need to be configurable.

**2) Internet :-** Data transmissions across the Internet need to be modelled with network delays and bandwidth restrictions.

**3) Simulation defined by time period :-** Converting simulation to a time-framed limited execution where user generates event continuously until a pre-defined time period expires.

**4) Service Brokers :-** CloudSim consist of one type of broker i.e. DataCenterBrokers which has two responsibilities firstly management of VM in multiple data centers and secondly routing traffic to appropriate data centers. Whereas in CloudAnalyst there are two types of service brokers one is DataCenterController that extends the DataCenterBrokers of CloudSim to perform VM management within a single data center and load balancing of VM's both and another one is CloudApp ServiceBroker of CloudAnalyst itself which manages routing of user requests between data centers based on different policies.

**5) GUI :** It setup and executes simulations easily and repeatedly to identify performance and accuracy of simulation.

**6) Ability to save simulations and results :** Able to save simulation experiments (input as well as output) as a XML or pdf file.

*D. Advantages of using CloudAnalyst*

- CloudAnalyst explores cloud applications in order to check it's behavior or nature for performance analysis for service quality assurance to end-users before deploying it on real cloud.
- CloudAnalyst has the ability to extend existing techniques and tools and defining new approach for effectively simulating cloud applications repeatedly in a controlled manner.
- CloudAnalyst has a flexible design so that it can be extended further in future.
- CloudAnalyst has the visualization capablity to generate visualized result report using efficient GUI.
- CloudAnayst seperates the programming task from the simulation experiment that enables the modeler to simulate experiments repeatedly in a controlled manner by focusing mainly on it with modifications to input parameters quickly and easily.
- CloudAnalyst enables a modeler or designer to repeatedly execute simulations quickly, easily and in a controllable manner.
- Results are analyzed quickly, easily and more efficiently by generating graphical output of the simulation.
- CloudAnalyst has GUI that helps in identifying any problems in performance and accuracy of simulation experiment.

*E. Features of CloudAnalyst*

**1) Ease of use**

- Setting and executing a simulation experiment is very easy using CloudAnalyst.
- CloudAnalyst provides easy to use GUI.

- Ability to define a simulation experiment with a high degree of configurability and flexibility
- Important feature is the level of configurability the tool can provide.
- Modeling an internet application depends on many parameters and most of the time value of these parameters need to be assumed.
- CloudAnalyst tool enables user or designers or developers to change these parameters quickly and easily and repeat the simulation experiments.

**2) Graphical User Interface (GUI)**

- Simulation generates a high amount of statistics that is highly desirable to understand through graphics in form of tables and charts.
- This helps in identifying important patterns of output parameters and also helps in comparison between related parameters.

**3) Repeatability**

- It is a very important requirement of a simulator.
- Same experiment with same parameters produce same results every time it is executed.
- Same experiment with different parameters produce different and more efficient results.
- CloudAnalyst tool is able to save experiment input parameters as an XML or pdf file.
- CloudAnalyst tool is able to save experiment output parameters as an XML or pdf file.

**4) Ease of extension**

- Simulator is expected to evolve continuously and must be used continuously.
- Simulator should support future extensions with minimum efforts.

*F. Output of Simulation Experiment*

1) Response time of simulated applications : Average, Minimum and maximum response time of all user requests is simulated.

2) Usage patterns of the applications : How many users use the application and at what time from different regions of the world and the overall effect of that usage on the data centers hosting the applications.

3) Time taken by data centers to service a user request : Request processing time of entire simulation. Average, minimum, maximum request processing time by each data centers.

4) Cost of operation

*G. Techniques used in CloudAnalyst*

- Java :- CloudAnalyst simulator is 100% built on Java platform using Java SE 1.6
- Java Swing :-Graphical User Interface of CloudAnalyst tool is built using Java swing components.
- CloudSim :- CloudSim features for modeling data centers is used in CloudAnalyst.
- SimJava :- SimJava is an event based simulation toolkit used in both CloudSim and GridSim. SimJava is an underlying simulation framework of CloudSim. Features of SimJava are directly used in CloudAnalyst.

*H. Installation Steps of CloudAnalyst*

Step 1 : Download Cloud Analyst using following URL

http://www.cloudbus.org/cloudsim/CloudAnalyst.zi

Step 2 : Copy zipped file in any location in your system

Step 3 : Extract the downloaded zip file to one particular drive (C: , D: or E: , etc.)

Step 4 : You will find following extracted folder format as shown in Fig. 3

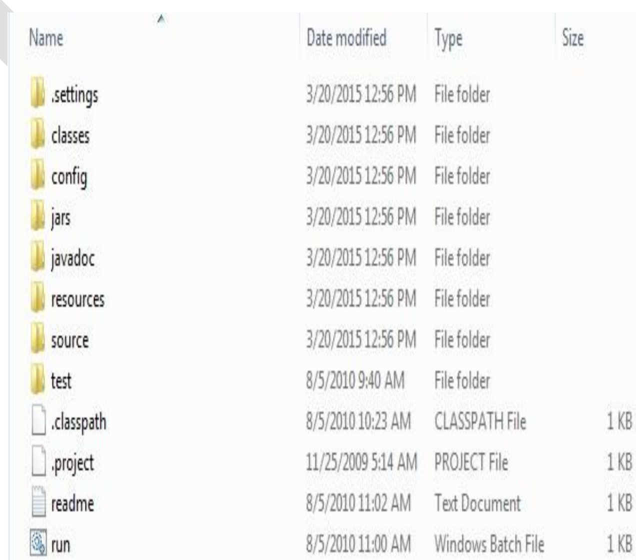


Fig. 3 : Extracted CloudAnalyst Folder Format

Step 5 : Either double click on Batch File to run CloudAnalyst or run it from command line using command (inside red color bordered rectangle) as shown in Fig. 4

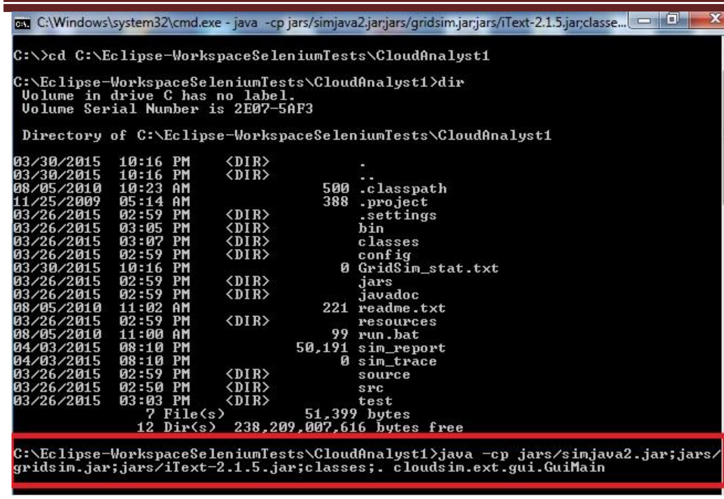


Fig. 4 : Running CloudAnalyst from command line

Step 6 : Either after clicking on Batch File or after running above command from command line, you will see following CloudAnalyst GUI as shown in Fig. 5

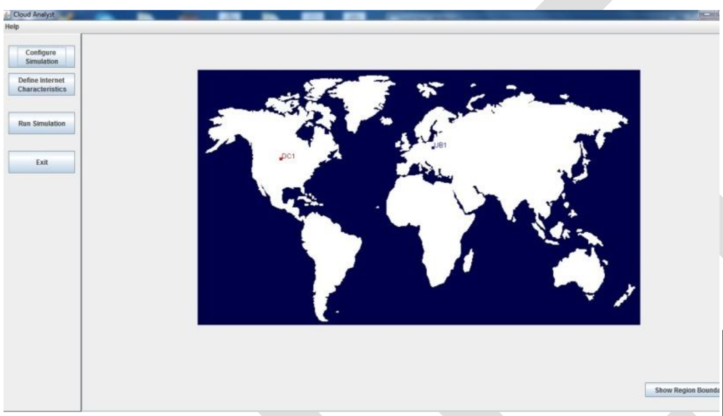

Fig. 5 : CloudAnalyst GUI

Step 7 : To see six different regions, click on "Show Region Boundaries" button on right side of GUI of CloudAnalyst, you will see following six regions as shown in Fig. 6

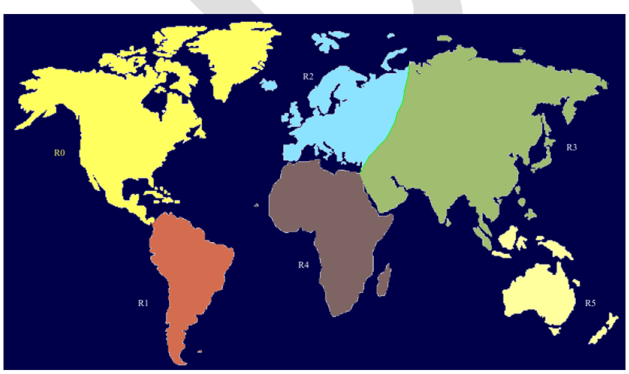


Fig. 6 : Regions

Step 8 : Configure simulation experiment and then run simulation.

## IV. TECHNIQUES

Cloud Analyst is a simulation tool developed at University of Melbourne, Australia whose purpose is to evaluate social networking sites or applications and to examine their behavior. CloudAnalyst is built on top of CloudSim framework. CloudSim models and simulates data centers and virtual machines. CloudSim is a network simulator tool based on NS2 (Network Simulator 2). CloudSim allocates hardware resourses for operating virtual machines based on different new policies. CloudAnalyst is used for simulating different web applications according to different configuration simulation parameters like user bases, data centers, application deployed, physical hardware details of data centers in the simulation experiment.

## V. SIMULATION EXPERIMENTS AND RESULTS

Before running simulation first configure or set simulation parameters like user bases, data centers, application deployment on data center, service broker policy, load balancing policy, VMs within single data center, etc. then run simulation and evaluate or analyze results based on overall response time.

### *A. Scenario 1 : When web application is deployed on single data center DC1with 5VMs*

TABLE 1
CONFIGURATION PARAMETERS

| **Simulation experiment 1** | |
|---|---|
| **Parameters** | **Values** |
| User Base | UB1, UB2, UB3, UB4, UB5, UB6 |
| Data Center | DC1 |
| Data Center Region | R0 |
| Service Broker Policy | Closest data center |
| Application deployment | DC1 |
| Load Balancing Algorithm | Round-Robin |

TABLE 2
APPLICATION DEPLOYMENT CONFIGURATION PARAMETER

| Service Broker Policy | Data Center | #VMs | Image Size | Memory | BW |
|---|---|---|---|---|---|
| Closest Data Center | DC1 | 5 | 10000 | 512 | 1000 |

Issue 5, Vol. 3 (April.-May. 2015)
ISSN 2249-6149

TABLE 3
DATA CENTER CONFIGURATION PARAMETER

| Name | Region | Arch | OS | VMM | Cost per VM $/Hr | Memory Cost $/s | Storage Cost $/s | Data Transfer Cost $/Gb | Physical HW Units |
|---|---|---|---|---|---|---|---|---|---|
| DC1 | 0 | x86 | Linux | Xen | 0.1 | 0.05 | 0.1 | 0.1 | 2 |

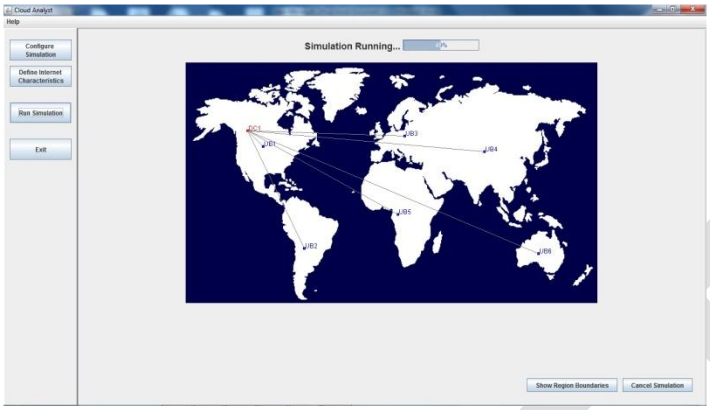


Fig. 7 : Data center DC1 allocated to different user bases according to geographical location of user bases


Fig. 8 : Showing response time to different user bases according to their geographical locations

Result of simulation :

| | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| Overall response time: | 292.12 | 37.61 | 607.62 |
| Data Center processing time: | 0.28 | 0.02 | 0.86 |

Fig. 9 : Overall Response Time Summary

| Userbase | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| UB1 | 50.17 | 37.61 | 60.11 |
| UB2 | 199.41 | 154.11 | 239.11 |
| UB3 | 299.27 | 232.62 | 369.12 |
| UB4 | 501.34 | 390.12 | 607.62 |
| UB5 | 500.33 | 375.12 | 602.62 |
| UB6 | 200.08 | 155.11 | 242.11 |

Fig. 10 : Response Time by Region

User Bases Hourly Response Times

UB1

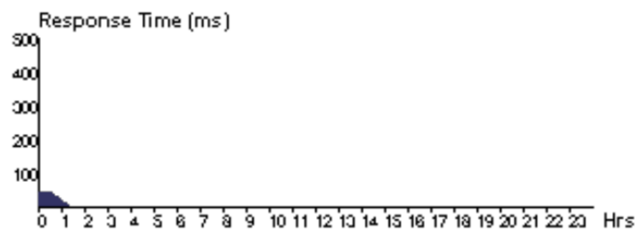


Fig. 11 : Hourly Response Time of UB1

UB2

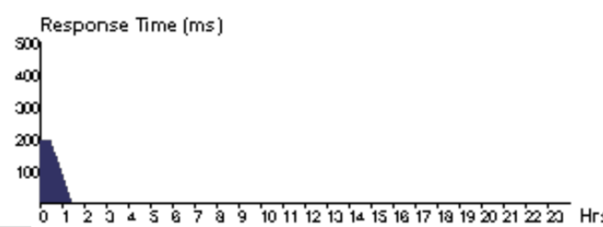


Fig. 12 : Hourly Response Time of UB2

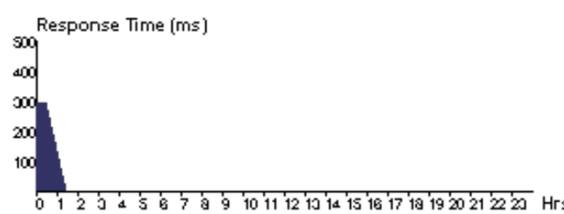


Fig. 13 : Hourly Response Time of UB3

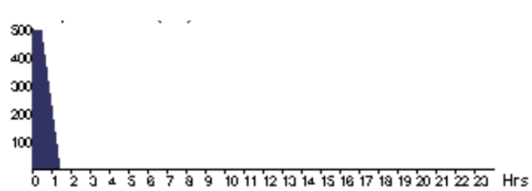


Fig. 14 : Hourly Response Time of UB4

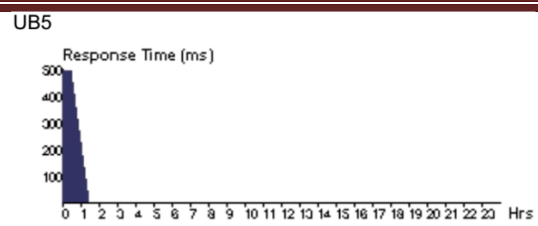


Fig. 15 : Hourly Response Time of UB5

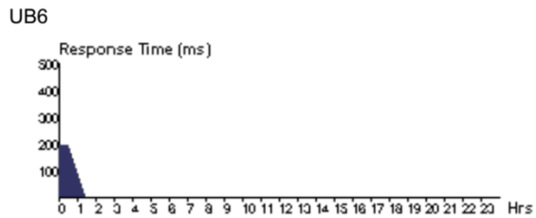


Fig. 16 : Hourly Response Time of UB6

Data Center Request Servicing Times

| Data Center | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| DC1 | 0.28 | 0.02 | 0.86 |

Fig. 17 : Data Center Request Servicing Times

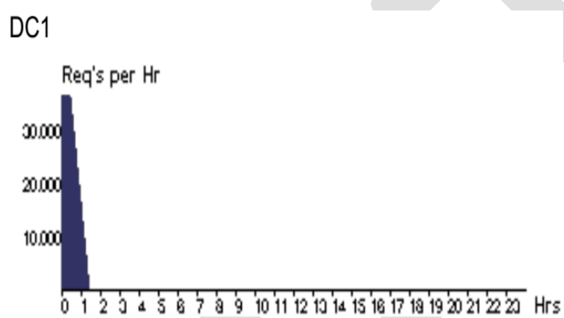


Fig. 18 : Data Center DC1 Hourly Loading

Cost

Total Virtual Machine Cost ($): 0.50

Total Data Transfer Cost ($): 0.38

Grand Total: ($) 0.89

| Data Center | VM Cost $ | Data Transfer Cost $ | Total $ |
|---|---|---|---|
| DC1 | 0.50 | 0.38 | 0.89 |

Fig. 19 : Cost of Operation

*B. Scenario 2 : When web application is deployed on two data centers DC1 and DC2 with 5 VMs in each*

Case 1 : Using Round-Robin Load Balancing Policy

TABLE 4
CONFIGURATION PARAMETERS

| Simulation Experiment 2 | |
|---|---|
| **Parameters** | **Values** |
| User Base | UB1, UB2, UB3, UB4, UB5, UB6 |
| Data Center | DC1, DC2 |
| Data Center Region | R1, R3 |
| Service Broker Policy | Closest data center |
| Application deployment | DC1, DC2 |
| Load Balancing Algorithm | Round-Robin |

TABLE 5
APPLICATION DEPLOYMENT CONFIGURATION PARAMETER

| Service Broker Policy | Data Center | #VMs | Image Size | Memory | BW |
|---|---|---|---|---|---|
| Closest Data Center | DC1 | 5 | 10000 | 512 | 1000 |
| Closest Data Center | DC2 | 5 | 10000 | 512 | 1000 |

TABLE 6
DATA CENTER CONFIGURATION PARAMETER

| Name | Region | Arch | OS | VMM | Cost per VM $/Hr | Memory Cost $/s | Storage Cost $/s | Data Transfer Cost $/Gb | Physical HW Units |
|---|---|---|---|---|---|---|---|---|---|
| DC1 | 1 | x86 | Linux | Xen | 0.1 | 0.05 | 0.1 | 0.1 | 2 |
| DC2 | 3 | x86 | Linux | Xen | 0.1 | 0.05 | 0.1 | 0.1 | 1 |


Fig. 20 : Data Center DC1 and DC2 allocated to different user bases according to their geographical locations

 Issue 5, Vol. 3 (April.-May. 2015)
ISSN 2249-6149

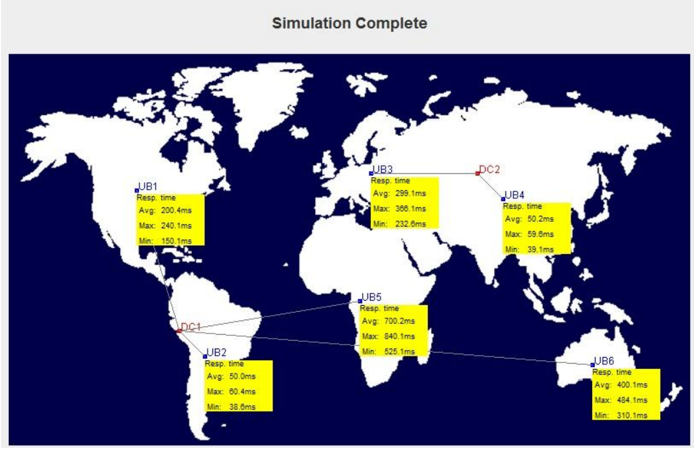


Fig. 21 : Response time to different user bases according to their geographical locations

Result of simulation :

Overall Response Time Summary

| | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| Overall response time: | 284.95 | 38.62 | 840.12 |
| Data Center processing time: | 0.32 | 0.02 | 0.88 |

Fig. 22 : Overall Response Time Summary

Response Time by Region

| Userbase | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| UB1 | 200.43 | 150.11 | 240.11 |
| UB2 | 49.97 | 38.62 | 60.37 |
| UB3 | 299.13 | 232.64 | 366.07 |
| UB4 | 50.24 | 39.14 | 59.64 |
| UB5 | 700.19 | 525.12 | 840.12 |
| UB6 | 400.10 | 310.11 | 484.11 |

Fig. 23 : Response Time by Region

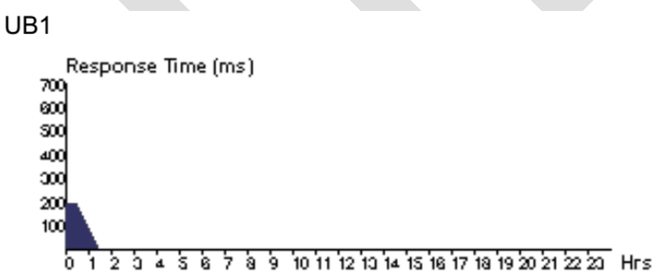


Fig. 24 : Hourly Response Time of UB1

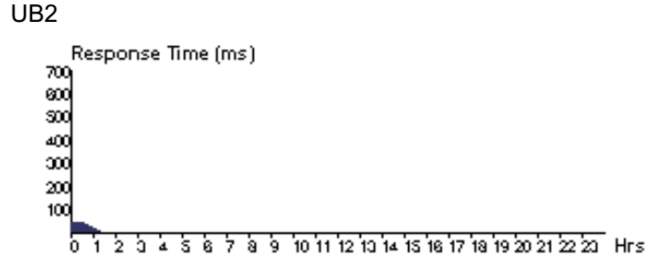


Fig. 25 : Hourly Response Time of UB2

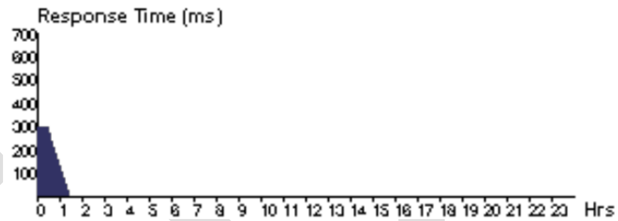


Fig. 26 : Hourly Response Time of UB3

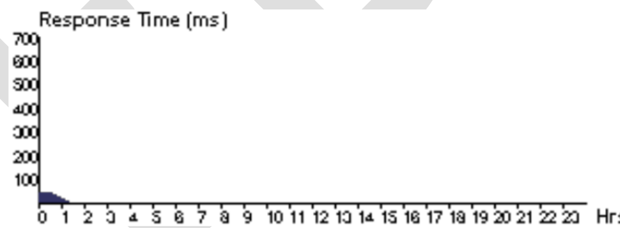


Fig. 27 : Hourly Response Time of UB4

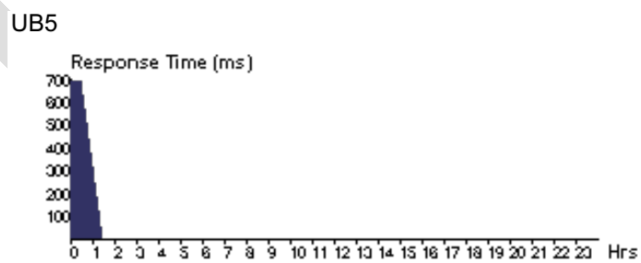


Fig. 28 : Hourly Response Time of UB5

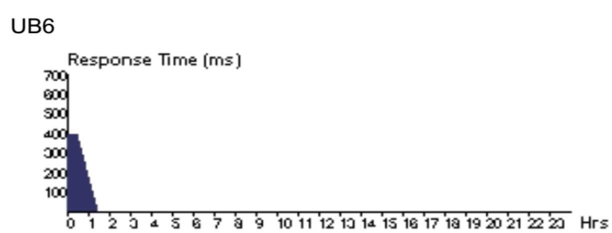


Fig. 29 : Hourly Response Time of UB6

Data Center Request Servicing Times

| Data Center | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| DC1 | 0.26 | 0.02 | 0.87 |
| DC2 | 0.43 | 0.02 | 0.88 |

Fig. 30 : Data Center Request Servicing Times

Data Center Hourly Loading

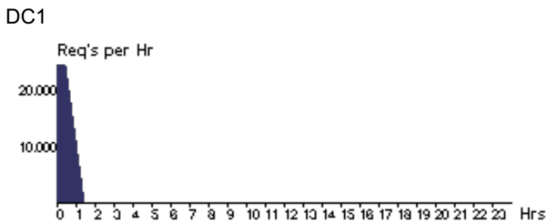


Fig. 31 : Hourly Loading of DC1

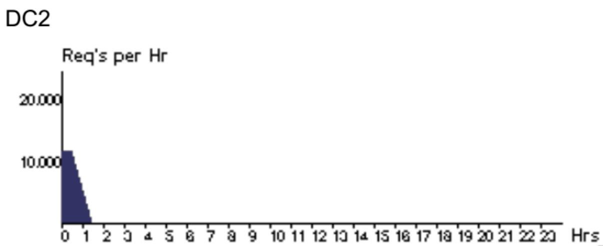


Fig. 32 : Hourly Loading of DC2

Cost

| | |
|---|---|
| Total Virtual Machine Cost ($): | 1.00 |
| Total Data Transfer Cost ($): | 0.38 |
| Grand Total: ($) | 1.39 |

| Data Center | VM Cost $ | Data Transfer Cost $ | Total $ |
|---|---|---|---|
| DC2 | 0.50 | 0.13 | 0.63 |
| DC1 | 0.50 | 0.26 | 0.76 |

Fig. 33 : Cost of operation

Case 2 : Using Equally Spread Current Execution Load Balancing Policy

TABLE 7
CONFIGURATION PARAMETERS

| Simulation Experiment 2 | |
|---|---|
| **Parameters** | **Values** |
| User Base | UB1, UB2, UB3, UB4, UB5, UB6 |
| Data Center | DC1, DC2 |
| Data Center Region | R1, R3 |
| Service Broker Policy | Closest data center |
| Application deployment | DC1, DC2 |
| Load Balancing Algorithm | Equally Spread Current Execution Load |

TABLE 8
APPLICATION DEPLOYMENT CONFIGURATION PARAMETER

| Service Broker Policy | Data Center | #VMs | Image Size | Memory | BW |
|---|---|---|---|---|---|
| Closest Data Center | DC1 | 5 | 10000 | 512 | 1000 |
| Closest Data Center | DC2 | 5 | 10000 | 512 | 1000 |

TABLE 9
DATA CENTER CONFIGURATION PARAMETER

| Name | Region | Arch | OS | VMM | Cost per VM $/Hr | Memory Cost $/s | Storage Cost $/s | Data Transfer Cost $/Gb | Physical HW Units |
|---|---|---|---|---|---|---|---|---|---|
| DC 1 | 1 | x86 | Linux | Xen | 0.1 | 0.05 | 0.1 | 0.1 | 2 |
| DC 2 | 3 | x86 | Linux | Xen | 0.1 | 0.05 | 0.1 | 0.1 | 1 |

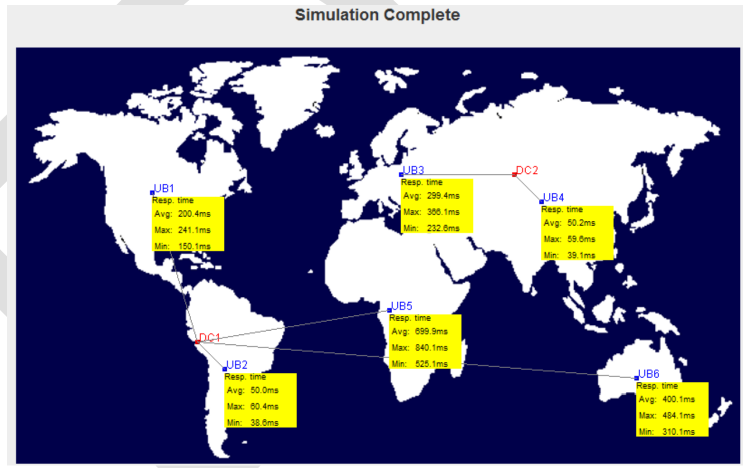


Fig. 34 : Response time to different user bases according to their geographical location

Result of simulation :

Overall Response Time Summary

| | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| Overall response time: | 284.93 | 38.62 | 840.12 |
| Data Center processing time: | 0.32 | 0.01 | 0.88 |

Fig. 35 : Overall Response Time Summary

Response Time by Region

| Userbase | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| UB1 | 200.42 | 150.11 | 241.11 |
| UB2 | 49.95 | 38.62 | 60.37 |
| UB3 | 299.41 | 232.64 | 366.07 |
| UB4 | 50.24 | 39.14 | 59.64 |
| UB5 | 699.93 | 525.12 | 840.12 |
| UB6 | 400.06 | 310.11 | 484.11 |

Fig. 36 : Response Time by Region

Issue 5, Vol. 3 (April.-May. 2015)
ISSN 2249-6149

UB1

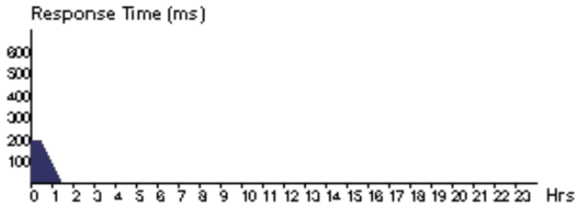


Fig. 37 : Hourly Response Time of UB1

UB2

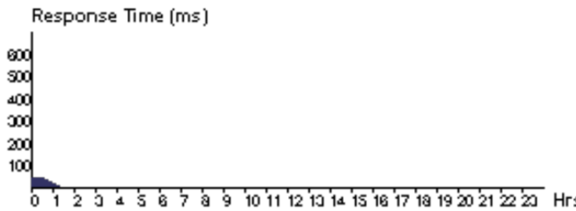


Fig. 38 : Hourly Response Time of UB2

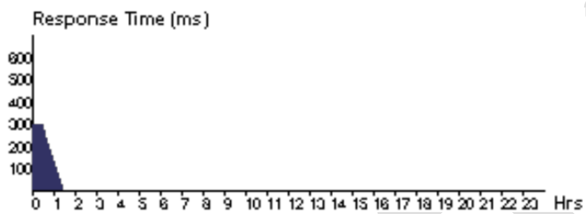


Fig. 39 : Hourly Response Time of UB3

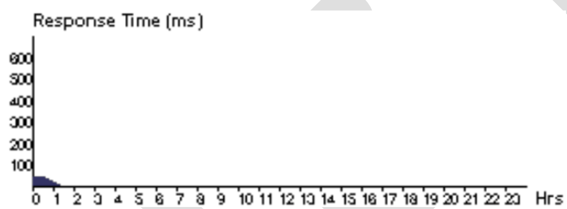


Fig. 40 : Hourly Response Time of UB4

UB5

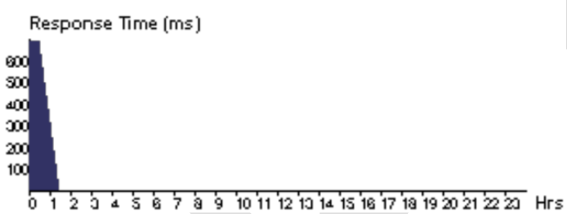


Fig. 41 : Hourly Response Time of UB5

UB6

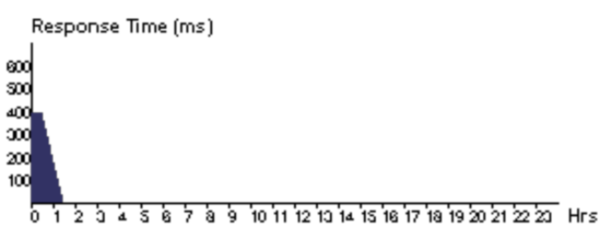


Fig. 42 : Hourly Response Time of UB6

Data Center Request Servicing Times

| Data Center | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| DC1 | 0.26 | 0.01 | 0.87 |
| DC2 | 0.43 | 0.02 | 0.88 |

Fig. 43 : Data Center Request Servicing Times

DC1

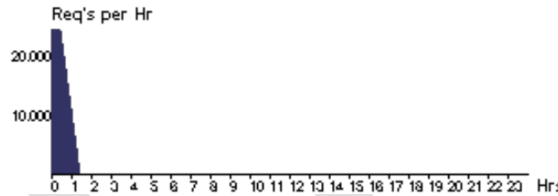


Fig. 44 : Hourly Loading of DC1

DC2

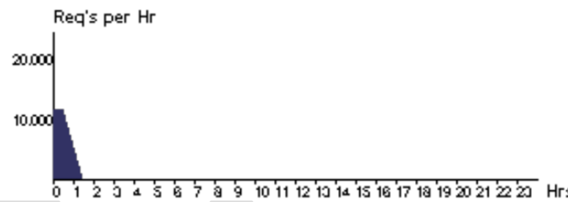


Fig. 45 : Hourly Loading of DC2

Cost

| | |
|---|---|
| Total Virtual Machine Cost ($): | 1.00 |
| Total Data Transfer Cost ($): | 0.38 |
| Grand Total: ($) | 1.39 |

| Data Center | VM Cost $ | Data Transfer Cost $ | Total $ |
|---|---|---|---|
| DC2 | 0.50 | 0.13 | 0.63 |
| DC1 | 0.50 | 0.26 | 0.76 |

Fig. 46 : Cost of operation

## VI. CONCLUSION

From the simulation experiment that we performed in various scenarios and from their results we analyzed and concluded that the overall response time (more or less) depends on the geographical location of data centers and user bases within regions. If user base is nearer or closer to one particular data center then user requests from that user base will be processed fast. Processing speed of VMs is fast or slow depends on the load balancing policy used. If processing speed of particular VM is fast then the time required to respond to user request will be less thus improving performance. So, we can say that, overall performance of cloud application depends on the processing speed of virtual machine (i.e. VM load balancing policy used), geographical location and number of data centers and user bases present within regions. In our simulation experiment, we tested the web application in two different scenarios as shown in Table 10 and got different values for overall average response time, data center processing time and data center request servicing times.

TABLE 10
SIMULATION SCENARIOS SUMMARY

<table>
<tr><th rowspan="2">Parameters</th><th colspan="5">Values</th></tr>
<tr><th>Scenario 1</th><th colspan="4">Scenario 2</th></tr>
<tr><td>User Bases</td><td>UB1, UB2, UB3, UB4, UB5, UB6</td><td colspan="2">UB1,UB2,UB3,UB4, UB5, UB6</td><td colspan="2">UB1,UB2, UB3,UB4,UB5, UB6</td></tr>
<tr><td>Data center(s)</td><td>DC1</td><td colspan="2">DC1, DC2</td><td colspan="2">DC1, DC2</td></tr>
<tr><td>Region(s)</td><td>R0</td><td colspan="2">R1, R3</td><td colspan="2">R1, R3</td></tr>
<tr><td>Virtual Machine(s)</td><td>5 VMs</td><td colspan="2">5 VMs in each data centers</td><td colspan="2">5 VMs in each data centers</td></tr>
<tr><td>VM Load Balancing Policy</td><td>Round-Robin</td><td colspan="2">Round-Robin</td><td colspan="2">Equally Spread Current Execution Load</td></tr>
<tr><td>Overall Avg. Response Time (ms)</td><td>292.12</td><td colspan="2">284.95</td><td colspan="2">284.93</td></tr>
<tr><td>Data Center Avg. Processing Time (ms)</td><td>0.28</td><td colspan="2">0.32</td><td colspan="2">0.32</td></tr>
<tr><td rowspan="2">Data Center Avg. Request Servicing Times (ms)</td><td rowspan="2">0.28</td><td>DC 1</td><td>0.26</td><td>DC1</td><td>0.26</td></tr>
<tr><td>DC 2</td><td>0.43</td><td>DC2</td><td>0.43</td></tr>
</table>

## REFERENCES

[1] Bhathiya Wickremasinghe, Rodrigo N. Calheiros and Rajkumar Buyya, "CloudAnalyst: A CloudSim-based Visual Modeler for Analysing Cloud Computing Environments and Applications," 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.

[2] B. Wickremasinghe, "CloudAnalyst: a cloudSim-based tool for modeling and analysis of large scale cloud computing environmens," MEDC Project Report, 2009.

[3] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities," The International Conference on Hign Performance Computing and Simulation, pp.1-11, 2009.

[4] R. Buyya and M. Murshed, "GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," The Journal of Concurrency and Computation: Practice and Experience, Vol.14, No.13-15, pp.1175-1220, 2002.

[5] S. H. Lim, B. Sharma, G. Nam, E. K. Kim, and C. R. Das, "MDCSim: a multi-tier data center simulation, platform," IEEE International Conference on Cluster Computing and Workshops, pp.1-9, 2009.

[6] R. N. Calheiros, R. Ranjan, C. A. F. De Rose, and R. Buyya, "CloudSim: a novel framework for modeling and simulation of cloud computing infrastructure and services," Technical Report, GRIDS-TR-2009-1, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 2009.

[7] S. K. Garg and R. Buyya, "NetworkCloudSim: modeling parallel applications in cloud simulations," 4th IEEE International Conference on Utility and Cloud Computing, pp.105-113, 2011.

[8] Rakesh Kumar Mishra , Sreenu Naik Bhukya, "Service Broker Algorithm for Cloud-Analyst," (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014.

[9] Dr. Neeraj Bhargava, Dr. Ritu Bhargava, Manish Mathuria, Ravi Daima, "Performance Analysis of Cloud Computing for Distributed Client, " IJCSMC, Vol. 2, Issue. 6, June 2013, pg.97 –104.

[10] Wei Zhao, Yong Peng, Feng Xie, Zhonghua Dai, "Modeling and Simulation of Cloud Computing: A Review," IEEE Asia Pacific Cloud Computing Congress (APCloudCC), 2012.

[11] Komal Mahajan, Deepak Dahiya, "A cloud based deployment framework for load balancing policies," IEEE, 2014.

[12] Niloofar Khanghahi and Reza Ravanmehr, "Cloud Computing Performance Evaluation," International Journal on Cloud Computing: Services and Architecture (IJCCSA) ,Vol.3, No.5, October, 2013.

[13] Tanveer Ahmed, Yogendra Singh, "Analytic Study Of Load Balancing Techniques Using Tool Cloud Analyst," International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 2, Mar-Apr 2012, pp.1027-1030.

[14] Dr. Rahul Malhotra, Prince Jain, "Study and Comparison of CloudSim Simulators in the Cloud Computing," SIJ Transactions on Computer Science Engineering & its Applications (CSEA), Vol.1, No. 4, September-October 2013.

[15] Ranu pandey, Sandeep Gonnade, "Comparative Study of Simulation Tools in Cloud Computing Environment," International Journal of Scientific & Engineering Research, Volume 5, Issue 5, May-2014.

[16] Parveen Kumar, Anjandeep Kaur Rai, "An Overview and Survey of Various Cloud Simulation Tools," Journal of Global research in Computer Science, Volume 5, No. 1, January 2014.

[17] Mahdi Mollamo tale bi, Rahe le h Maghami, Abdul Samad Is mail, "Grid and Cloud Computing Simulation Tools," International Journal of Networks and Communications, 2013.