

## Selective Testing As A Service for Cloud Computing

Garamu Tilahun Itichaa<sup>1\*</sup>, Tesfaye Fufa Gedefa<sup>2</sup>

<sup>1</sup>School of Computing, Debre Merkos Institute of Technology, Debre Merkos University Debre Merkos, Ethiopia,

<sup>2</sup>College of Computing, Institute of Technology, Debre Berhan Universit, Debre Berhan, Ethiopia,

Corresponding Author: Garamu Tilahun Itichaa

---

**Abstract:** Now a day's cloud computing has become an integral part of people's lives to acquire services such as software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). Users get services by deploying their data and application on remote servers. However, knowing the real information about a cloud performance before subscribing data and application is great challenge for the users. The objective of this study is to give selective testing and to investigate cloud performance status and capacity of servers. Finding the parameters helps us to measure the actual performance of a cloud computing on some specific data and application. These parameters will help users, enterprises, developers and IT specialists to measure cloud performance based on their requirements and needs. In order to fulfill the objective of this study, an experiment and implementation have been done in the lab test environment. Java programming language was used for changing the algorithm as one method. The study is unique from others we used Local cloud deployment method using open source cloud OpenStack to conduct real test and Applications Manager tool used for testing services of Amazon EC2. In the experiment, performance parameters (Throughput, response time and latency) including cost were tested and analyzed; and we found that throughput of local cloud is higher than EC2 and Windows Azure and Medium response time and lower cost as result tested in laboratory, and the future direction should be optimizing performance when you come with performance difference with cloud providers by assigning the same computing resource.

**Keywords:** Cloud Computing; Performance evaluation; Local cloud; Performance measurements; Performance parameters.

---

Date of Submission: 06-05-2019

Date of acceptance: 20-05-2019

---

### I. Introduction

Cloud computing is one of the greatest innovation in Information Technology in recent decades and it is a design of technology ever made that provides services, applications, and resources through a network <sup>[1]</sup>. Many several different companies and independent individuals have given definitions for cloud computing according to their strategies, visions, and products. A working definition of Cloud Computing goes as cloud computing is "a model for allowing suitable, on-demand network access to a shared pool of configurable computing resources (i.e. networks, servers, storage, and applications) that can be rapidly provisioned and released with minimal management effort or service provider contact." According to this definition, there are three types of service models, five major cloud characteristics and four classes of deployment models. Infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) are service models of cloud computing. On-demand self-service, resource pool, elasticity and scalability, pay-per-use and broad-network access are the main characteristics of cloud computing. Depending on the kind of cloud deployment model, cloud computing divided into the public cloud, private cloud, hybrid cloud, and community cloud delivery models <sup>[2]</sup>. There are various cloud-computing providers for various aspects. Different providers give different service delivery models. For instance, in 2006, Amazon was the first cloud provider that provides infrastructure as a service. OpenStack, Microsoft Azure, open nebula are also types of infrastructure service delivery models. Google mail and Salesforce are examples for SaaS cloud providers. Red Hat OpenShiftF, Amazon Simple DB/S3, and Google App Engine are examples for PaaS cloud vendors [3]. Cloud is beneficial for users, because cloud computing is virtualized technologies, has better service and collaboration, with the wide availability of high bandwidth networks and high capacity of storage. This makes a cloud to work in the web browser as a thin client that allows users to run any application. The resources might be computational power, has more storage, inter-networked, and run huge applications. Therefore, IT resources are elastic. They can be provisioned for the exact duration they are necessary and released when the consumer no longer needs them <sup>[4]</sup>. <sup>[5]</sup> Testing-as-a-Service delivers cloud servers quality management solutions in a flexible service model that accelerates the implementation of the quality center of excellence. IBM reported the experience of

cloud testing small business division, resilience and cost-efficient cloud-based development and testing environment is implemented [6]<sup>[7], [8]</sup>. The goal of testing is to determine how the system performs under different circumstances. These circumstances can be a great load, resource scarcity, or specific types of data. It is also performed to evaluate different performance enhancements<sup>[9]</sup>. The researcher motivated to study on testing of cloud performance because there are so many challenges and problems like throughput, response time, latency and cost of the resource. Most of the time users blindly host their data and application in the cloud. This faces users the risk of cloud computing performance.

The objectives of the study are selective testing for cloud performance to identify which and what factors affect the service quality of cloud providers. To hit the objective of the study, we addressed solutions for problems using different approaches and algorithms. Different proposed algorithms that had been taken as action were testing as service descriptions and algorithms, which were designed using an object-oriented programming language called class diagram. The problem raised was solved using Java programming language. In addition to this methodology, local cloud lab test environment was deployed for real test data and application. Since the study focused on selective testing (ST), it is indispensable to compare and evaluate several different cloud providers service.

## **II. Related works**

In recent years there has been a number of researches done in the area of cloud computing. However, a few types of research, which test and measure the performance of cloud computing service was conducted. Several papers also covered some concepts about testing as a service. The gaps of the thesis; the methodology used, their initiation; results got in the previous work was revealed in the next paragraphs. At last, the contribution of the study is added. [10], an overall perspective on cloud evaluation criteria and highlight with help of simulation were provided. The researcher's motivation is the higher performance of services and anything related to the cloud have to influence users and service providers. Due to the increase in the development of cloud computing, it is predicted that storage and computing on personal computers have been forgotten and all of these things will be transferred into the Clouds. Major factors of cloud computing performance are presented and analyzed. The authors use Cloud Analyst tool, which is a graphical design, based on Clouds for simulation determination. The paper took a different sample of users and requests per unit time. The paper concluded that if the data center is overrated capacity by the user, it will not only be unprofitable but it also reduces the efficiency of the center. The same number of user requests is connected to one data center. The result of the study indicates increment in a number of requests per unit time has little impact on response time, on processing time and data centers. It is effective on the cost of data transfer. Increasing the number of requests increases the cost of data transfer. Increasing power and speed of the data center is not always efficient. Sometimes it only has additional costs. For this case, the distribution of data centers and using the closest data center is better and more optimal. The factors that are not covered in this study were considered in the researcher work. Additionally, testing performance of cloud computing continues activity because of development of cloud computing and users request increased. Other measures will be conducted in this study since cloud computing still has shortages of performance; for instance, selecting an infrastructure configuration type and cost calculation system.

Service is the fact that most providers push a number of users to use their services but don't pay attention at the declining quality received by the average user. In this work, the researchers conducted experimental setup regarding service quality gained by the cloud users. To analyze the performance of the cloud, the criteria received was a number of users and number of requests per period of each user. The number of each user access single application service and all traffic directed to the same VM of cloud computing. The interval time used between requests is a normal distribution. This means a number of requests per period for each user are assumed to equal and the difference is the interval between requests is in a random process. The proposed tool used was eucalyptus as an open source cloud computing and they used HTTP protocol in case of evaluating how the quality service was provided by Eucalyptus.

The hypothesis indicates that an increasing number of users will decrease the average performance received by users, which is a problem with the quality of service (QoS). The authors propose to use a specific model of cloud computing architecture with specific service running on it. As usual, the data was subscribed on the external cloud, and the users of the application were from the same enterprise. So, with this model, most of the users of the service application came from the same network or region and also members of the organization. Users' request repeatedly by the process called stochastic approach. To examine and have a comprehensive result, four setup experiments had been planned in different network regions. The main difference between the authors and our thesis is that the authors focus on performance testing at the eucalyptus level whereas our thesis focuses on the selective testing for CC as a general by deploying local cloud using OpenStack platform and Application Manager tool used to test Amazon via online testing.

As stated in<sup>[11]</sup>, the authors tried to identify factors that affect the performance of cloud computing. To

study the performance of cloud and find a solution, they used Eucalyptus by using KVM hypervisor. Different tools were used to demonstrate the performance of eucalyptus. Every experiment was performed with two running virtual machine instances. To analyze the capability of Eucalyptus, they took 15 GB disk, 512 MB RAM, and 1 CPU core using **Iperf** simulation tool. A simple program was created to check processor sharing between instances. This program consists of a while loop which calls function and saves the output in a text file. Flows of UDP packets, packet loss, RTT variations, and jitter results were obtained. The test involves incoming and outgoing packets and conducts each part of the experiment on both VMS.

In <sup>[12]</sup>, The authors suggested that different approaches to performance analysis of cloud computing platforms and proposes model. They proposed queuing model architecture. The model shows, the cloud computing platform is modeled as multiple queues and the VMs are designed as service centers. This means to test performance, VMs stores application which runs on servers. The analyzed result could be captured using JMT tool. The author used Virtualization software such as Xen, KVM to run an application through the cloud and to observe performance. The virtual machines of an application run on either one cloud node or multiple cloud nodes, and each of them has the same computing resource to see the difference, comparison, and evaluation of service. The author release tourism management system and product search engine which was run to measure parameters. The problem of the research was users submit their requests for computing resources like CPU, RAM, disk, application, infrastructure and software which are provisioned in the cloud without the users being aware of the details of the execution environment. The researcher saw the shortage of paper and brings proper awareness for users by the way of selective testing of cloud performance. This could be done through analyzing of computing resource request of a number of users by given of constant access time and create single CID(cloud identifier).

### III. Methodology

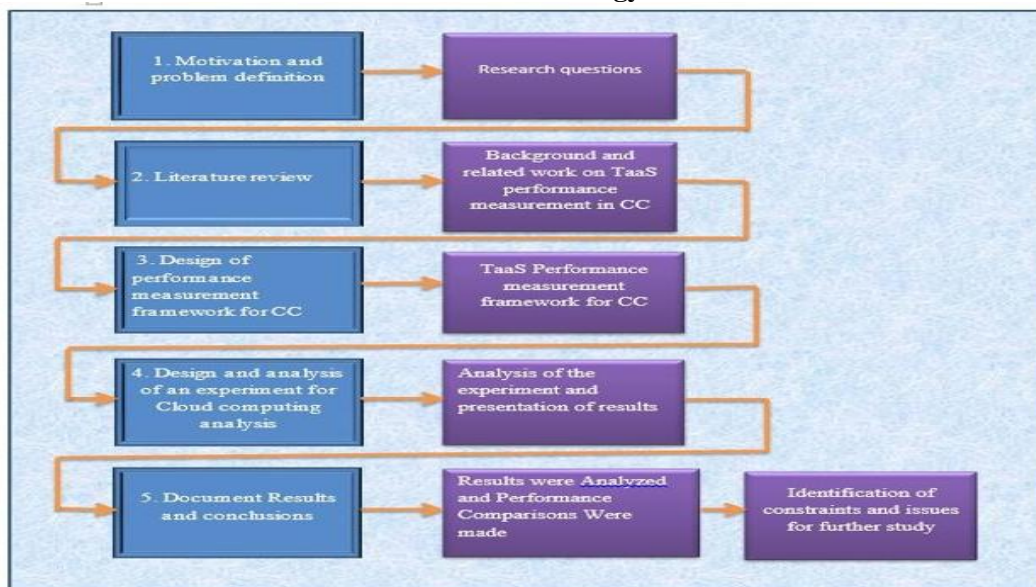


Figure 1: Graphical representation of research methodology

#### Algorithms Used

The researcher prepares to test as a service for cloud computing by means of different techniques and methodologies. Algorithms used describe the procedures applied. Interactive interface. This algorithm uses to design a user interface for interacting with cloud computing. Users are not familiar with cloud computing. Therefore, it is important to make the interaction with the cloud as easy as possible. It is very vital for the interest of the users as well as to have the confidence to upload their application in the cloud. Therefore, STFCC acts in the middle of the user and cloud system and reach cloud information to customers. The proposed algorithm brings all performance of cloud providers in a single database.

Performance of testing as a service for cloud computing pseudo code:

#### Algorithm 1: Interactive interface

**Input:** vCPU, RAM, Hard Disk; time; number of users no of Users; cloud providers cp; cloud provider's performance cPp;

**Output:** Performance P (throughput, response time, latency, description of cloud providers).

```

1. begin
2. For all cloud providers cp getcpPInfo.
   loop1: Choose cp
3. cloudcp1 ← Test
   loop2: if (VM= vCPU, RAM, Disk) && (tested based on time, number of Users, application)
   for user has App1, App2...Application
     VM ← put (Application)
     Number of Users ← send request
4. cp1 (Application) ← run
5. P ← respond then
6. if (user satisfied to providers performance)
7. Choose cp1
8. if (user not satisfied to cp1 performance)
9. go to loop1
10. cloudcp2 ← Test
11. go to loop2
12. cp2 (Application) ← run
13. P ← respond
14. If (cp1P> cp2P)
15. Choose cp2
16. Else
17. go to loop1

```

### User requirements

STFCC deals between user and cloud to give testing as a service. Users' requirements collected from users those who need to know cloud performance and subscribe their application. The users always require fast access in order to appropriately run business organizations. Cloud providers are business oriented, though to know which cloud provider give fast and best performance the users can use STFCC system. The system collects detail business requirement of users including amount of CPU, bandwidth in different speeds, memory RAM and storage of hard disk. Then, fill type of users' requirements in each cloud provider and display performance of each cloud providers' service for users. Decision of users depends upon cloud performance they have seen through interactive service. The following statement shows pseudo code for users' requirements:

### Algorithm 2: User requirements

**Input:** computing resource, no. of users, time, application

**Output:** SaaS, PaaS, IaaS provider performance

```

1. begin
2. For users need=Service type
3. If (users==IaaS)
4. display Providers ← IaaS (list) then
5. Provider ← choose
6. requirement list (computing resource ,number of users, time, application) ← executed list
7. performance ← read then
8. Else if (users==SaaS)
9. display Providers ← SaaS (list) then
10. Provider ← choose
11. requirement list (computing resource, no. of users, time, application) ← executed list
12. performance ← read
13. Else
14. (users==PaaS)
15. Providers ← PaaS (list) then
16. requirement list (computing resource, no. of users, time, application) ← executed list
17. performance ← read
18. End if

```

### Calculate cost and respond providers pricing policies

Cost saving is one of the main purpose that enterprises or users consider when migrating their application to the cloud. For this reason, cloud consumers able to make accurate estimation of cost through selective testing process. A question that comes from user is pay-per-use charging model. So the question is how much will cost to run one application service in the cloud for a given day (s), knowing that it may needs 90 GB of storage hard disk? When users bring their requirements such as memory capacity, bandwidth, and CPU speed and disk storage capacity, the system shows performance. Next to this, other link is displayed. It demonstrates total cost depending on the cloud vendor management and cost pricing system. Normally, cost covers usage duration, volume, read and write operation for storage on the disk, size of data incoming or outgoing. Mechanism of cloud providers' calculation depends on type of instance they give. Size of users' application decides instance type they are going to purchase. There are large, medium and small types of instances. These kinds of instances are associated with cost. When virtual machine was created for users to host

their application, cloud manager decides instance type as a result of information gained from subscribers. Different providers have their own pricing policy. The way of calculating cost algorithm is designed according to vendor's policy. For example, some cloud providers calculate cost based on type of virtual machine rented while other vendor is based on the pay-per-use, or access per hour, per month and per year. The next algorithm is applied for cost calculation based on computing resource and providers' price policy.

**Algorithm 3:** Calculate cost and respond providers pricing policies

**Input:** cloud providers name cp; virtual central processing unit (cpu) vCPU; RAM; Hard Disk; API; region;

**Output:** Cost of cloud providers

```

1. begin
2. For cost ← price of cloud providers
3. if (price calculation policy based on provider)
4. Cost calculation = select policy
5. Configuration ← type(m1.small, t1.tiny, m1.medium, large)
6. VMid ← given ID
7. Size ← resource(vCPU, RAM, Hard Disk)
8. cloud ← API
9. Region ← datacenter(location)
10. cost ← rate
11. if (cost=per hour)
12. total cost=resource +contract period
13. cost ← rate
14. Else
15. if (cost=per day)
16. cost ← rate
17. go to 12
18. Else
19. go to step 3-to-step 10
20. End if
21. End for
    
```

With regard to solving user's problems, we intended to dig out cloud providers' pricing policies. The steps stated show that certain users may tend to pay per hour access while others may prefer to pay per year. Because of such variance of users' interest, system designed shows every provider's pricing facet according to users request and requirement.

**Algorithm 4: refreshing database**

**Input:** cloud provider cp; API; VMid; vCPU; RAM; HardDisk; instance type;

**Output:** Database Refresh

```

1. Begin
2. for new cloud provider
3. if (cloud is new cloud provider)
4. for API, VMid, vCPU, RAM, Hard Disk, instance type
5. cp ← VMM(register)
6. API(VMid) ← API
7. VM ← VMM(run)
8. Database ← update
9. for (cloud provider in CID)
10. if (cp_change = service) then
11. Data ← load
12. Rate="per day"
13. Database ← update
14. Else (cp = changed)
15. cp ← delete (from database)
16. End for
17. End if
    
```

#### IV. Design Architecture Of Selective Testing For Cloud Performance Testing

The proposed model retains the cloud application that assists on performance evaluation. The proposed cloud-computing model is comprised of a front end and back end architecture. These two elements are connected through a network. The front end is the vehicle by which the user interacts with the system and the back end represents cloud database, which contains cloud information. The front end is composed of a client computer, or computer network of an enterprise, and the applications used to access cloud. Back end will provide the required services in a required manner.

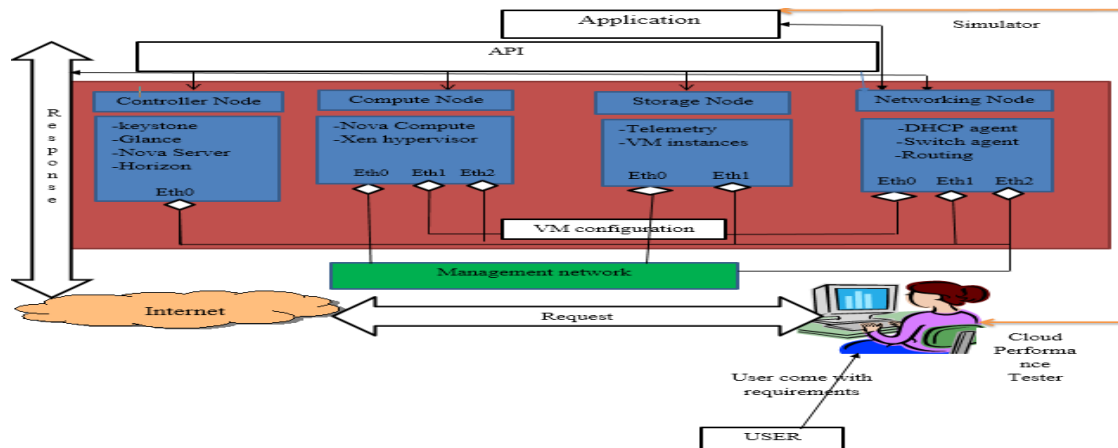


Figure 2: Proposed cloud performance testing architecture

A diagram with the whole system, which is drawn to develop thesis structure, is shown in the above figure. Once the VM image has been transparently deployed on a physical machine, the resource status is running as the instance is booted. At the end of the boot process, the resource status will be installed. The installed resource can be used as a regular computing node immediately after the booting process is completed through connection. The figure 3-7 refers skeleton of the thesis procedure, where request and response communication is performed. With this scenario, simulation tool has to run to look at the behavior of the packets going through the network. Afterwards, how the network performance was different when some factors could be changed (number of clients, type of instance, etc.) is to be checked. The traffic was recorded in the network tool for a further analysis and pattern evaluations. Finally, with the pattern obtained, the researcher comes to the last part of the traffic pattern gathering of key performance indicators. In this part, many data sources and many requests are set up to repeatedly recreate the traffic pattern towards the cloud. By this way, cloud performance is found when it is able to handle heavy traffic loads and requests come inside from different data sources as well as responding repetitive request of data.

### Application

Large data size can be stored in the cloud computing. Through the process of virtualization techniques, hardware of physical machine can be shared into several virtual machines. On the virtual machine, data has to be hosted and accessed by user. In this architecture, application sits on VM. When user access that application, cloud tester look at throughput, response time, latency and cost.

### Application Programming Interface (API)

Interface through which administrator and user get and access cloud is known as API.

### Network that capture KPIs

While the application is running and accessed by different group of users, cloud tester gather and write down what performance has been given by cloud through tool.

### VM configuration

Type of instance created in the cloud providers has different names. For example Amazon EC2 VM names are: small, large and extra-large<sup>[13]</sup>.

### Cloud performance tester

To analyze network performance and packet flow situation, the cloud tester has responsibility to capture and see providers' strength and weakness. Identifying and analyzing different cloud performance providers gives confidence for users in hosting application.

### Internet

Cloud computing is the delivery of computing services over the internet and it is metaphor of internet. Cloud computing is nothing without internet. The availability of internet determines existence of cloud. In the diagram above, we understand that internet link a user and administrator with cloud computing. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available[6]

### **Request**

Through http/s and URL of cloud providers, users send request to cloud. Generating more load to cloud shows about cloud providers' performance such as scalability. By stressing cloud servers, we can analyze cloud performance.

### **Response**

It is known that if there is request, there is also response. Users access different cloud like Amazon, Google, Facebook, Microsoft azure, etc. The researchers put application on the virtual machines in the case local cloud and test response time, throughput and latency which also applied in other cloud providers.

### **Management network**

Management network is a separate network for use by cloud operators. Naturally it consists of a separate switch and NICs, and is a recommended decision. The purpose of management network is for internal communication between cloud components. The IP addresses on this network should be reachable only within the component nodes to facilitate communication among them. This segregation prevents system administration and the monitoring of system access from being disrupted by traffic generated by guest computers.

## **V. Experimentation And Laboratory Test Beds Setup**

Our study was started with identification of cloud computing that provide an infrastructure as a service. It is the most appropriate layer to deploy reference model and cloud framework. The experiment was prepared using open source cloud service provider. Open stack platform is one of them which have to be evaluated through the study [14].

**Controller node:** The controller act as cloud service manager that has the capability of managing cloud service applications. Identity service, image service as well as management portions of compute and networking, networking plug-in, and the dashboard were run on the controller node. Controller node was configured with one NIC, four RAM GB, 3.3 CPU, 1.7GHz and 500GB storage.

**Network node:** This could be run net working plug-in and several agents that deliver tenant net works and provide switching, routing, NAT, and DHCP services. Network node also handles external (Internet) connectivity for tenant virtual machine instances. The main task of Network node is tunneling purpose among the nodes. It deal with providing internet connection among nodes and pour the communication easily when internet connection is available. The machine has three interfaces, two of them connect internal devices, and the third interface part connected to external network, the node is configured with three NICs, two RAM GB, 3.3 CPU, 1.7 GHz and 500GB storage.

**Compute node:** compute nodes run different components. Hypervisor which operates tenant VMs or instances. Xen hypervisor used to configure, manage and monitor instances. Networking plug-in and an agent that connect tenant networks to instances. Telemetry agent to collect meters also run on this node. Compute node act as a user and on the machine supports to create virtual machines. The node is configured with three NICs, two RAM GB, 3.3 CPU, 1.7 GHz and 500GB storage.

**Storage node:** The Storage node contains disks that the Storage service provisions for tenant virtual machine instances. Telemetry agent runs on storage to collect meters. Network interface on storage node added to provide communication performance of storage services. The storage nodes contain the disks that the object storage service uses for storing accounts, containers, and objects. The function of storage node is storing data or application of users, because when cloud developed and deployed several tenants and volumes has been created. Virtual machine has the capability of storing data and application of users depending on the business requirements. The node is configured with two NICs, two RAM GB, 3.3 CPU, 1.7 GHz and 500GB storage.

### **Hardware requirements used in four nodes for local cloud configuration**

The test bed was composed of four hosts. Table below shows in detail hardware requirements that were used in deployment laboratory test environment.

**Table 1:** Hardware Configuration of the system

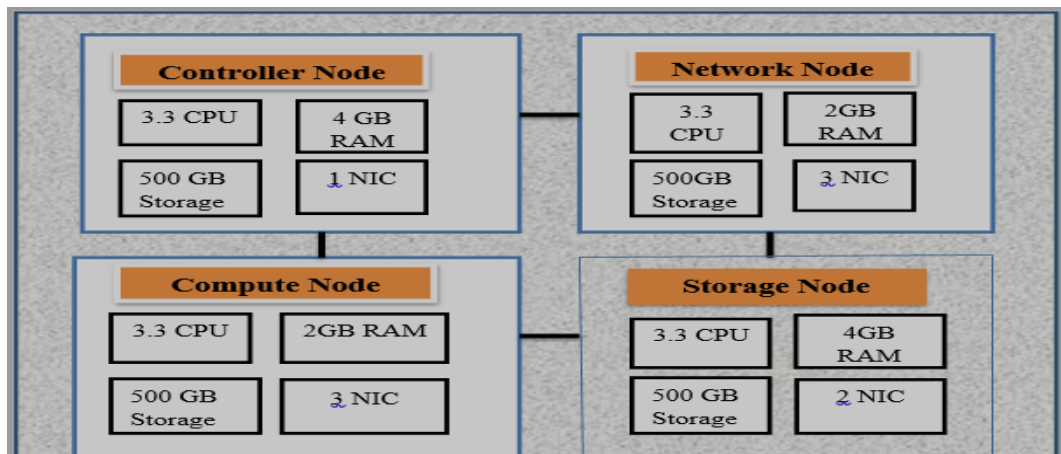
Hosts	CPU	RAM	Storage	NIC	System Type
1.Controller	Intel(R) Core(TM) i3- 3317U CPU@1.70Gh	4GB	500GB	1	64-Bit OS
2.NetworkNode	Intel(R) Core(TM) i3- 3317U CPU@1.70Gh	2GB	500GB	3	64-Bit OS
3.Compute Node	Intel(R) Core(TM) i3- 3317U CPU@1.70Gh	2GB	500GB	3	64-Bit OS
4.Storage Node	Intel(R) Core(TM) i3- 3317U CPU@1.70Gh	2GB	500GB	2	64-Bit OS

**Stating virtual machines in cloud computing**

Virtual machines have been created for testing purpose on the local cloud. Cloud state is what the instance requests are being matched against to find optimal placements. The resource management system keeps track of the current cloud state and feeds it to the placement controller. Information about capacities of physical machines used to host VM loads and the network that connects them is used together with current VM allocations to find machines with extra capacity where new VMs could run.

**Architecture of laboratory test**

Before testing performance of cloud computing, we have to logically design architectural theory. This is very important decision to get perfect answer for our research questions which are stated so far. Nodes were configured with different packages installed on them which were connected with each other and with a device called switch which distributes and provides internet access. Mainly the figure below shows more detail.



**Figure 3:** Design architecture of lab setup: source (own)

**Configuration steps on laboratory test environment**

In above diagram lab installation, setup structured and different nodes with different tasks could be designed in laboratory test environment for the purpose of testing. Cloud prototype must be deployed to have deep test. Local cloud is highly configurable to meet different needs with controller nodes, various computes, networking, and storage options. Network interfaces should be configured. The DEVSTACK Kilo shell script was used to install Local cloud on the target server node. Local cloud installation process is as follow:

1. Install Ubuntu 14.04 LTS version on the target server node
2. Install each packages on each different nodes
3. Clone the DEVSTACK installation software from Github source
4. Deploy the Local cloud by executing the DEVSTACK shell script on the laboratory environment.

**Local cloud networking architecture layout.**

The architecture layout with Local cloud networking (neutron) requires one controller node, one network node, and at least one compute node. The controller node contains one network interface on the management network. The network node contains one network interface on the management network, one on the instance tunnels network, and one on the external network. The compute node contains one network interface on the management network and one on the instance tunnels network.



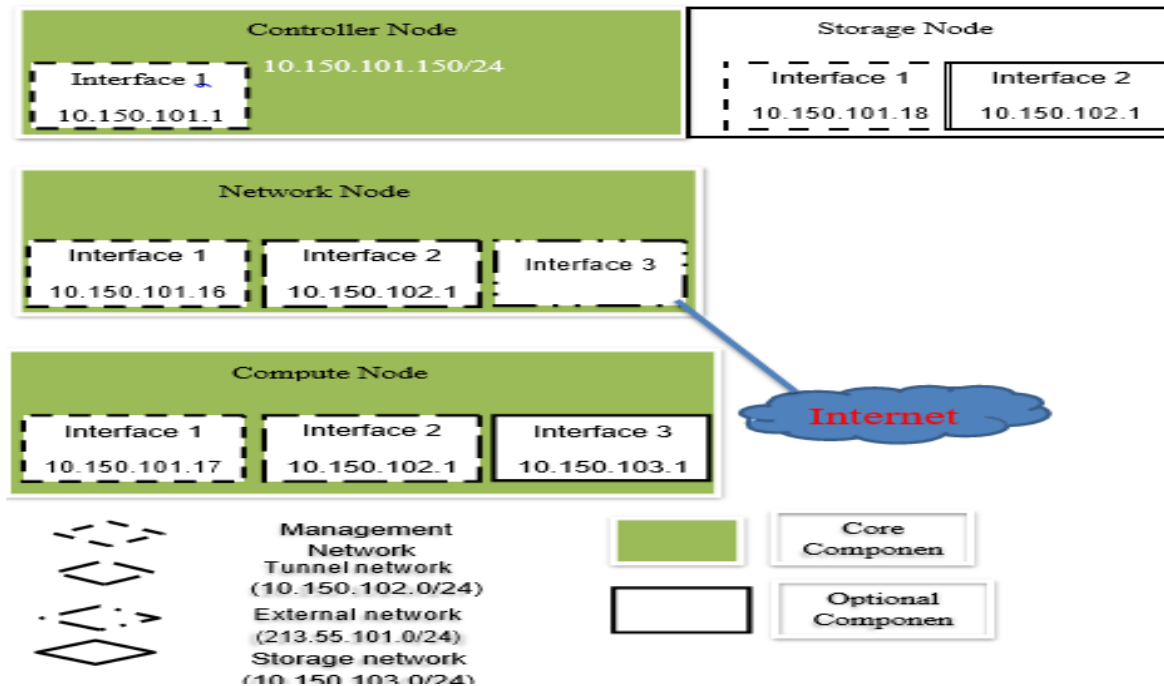


Figure 4: Neutron local cloud architecture: source (own)

**Network Configuration**

After neutron layout and IP address is decided, the next step is configuring and editing of network, changing the hostname of the computer node. This step is strictly recommended to configure because without applying this step it is impossible to have intercommunication of each node. In the first stage we assign IP address manually then we change the name of the computer node using terminal session. The detail IP address assigned to each node is described in the next table.

**Table 2: IP address assigned for each computer**

Name	Interface (NIC)	IP address	Gateway	Subnet-mask
Controller node	1	10.150.101.150	10.150.101.1	255.255.255.0
	1	10.150.101.160		
Network node	2	10.150.102.160	10.150.101.1	255.255.255.0
	3	Not assigned		
Compute node	1	10.150.101.170	10.150.101.1	255.255.255.0
	2	10.150.102.170		
	3	10.150.103.170		
Storage node	1	10.150.101.180		255.255.255.0
	2	10.150.102.180		

**Experimental setup details**

The experiment was conducted in Ubuntu operating system in which the evaluation was run number of times both in virtual machine of cloud environment. The experiment was executed in number of virtual machines environment. While request and response time was performed, the key performance indicators are gathered for the next evaluations of the result. Hardware virtualization enables a single physical platform to run multiple operating systems and software stacks. Virtualization creates an abstraction layer between user and physical resource but at the same time it provides the user the illusion of direct interaction with the physical resource. The Virtual Machine Monitor (VMM), also known as the hypervisor, establishes the abstraction layer that encapsulates and isolates each Virtual Machine (VM). The VMM runs on the actual machine and maps physical resources (processing power, memory, storage, and network) to VMs. The Operating System (OS) running inside a VM make use of the virtual resources mapped to the confining VM. Consequently, the physical resources of a machine can be partitioned between multiple VMs.

On virtual machines, one type of application is installed. When local cloud was installed and displayed horizon dashboard, instances run through mechanism of virtualization. The middleware called Xen hypervisor which was installed on hosts of computer and manages VMs. The other virtual machines request application service from the cloud which was installed on the other VMs. Following are the steps that we took to perform our test in local cloud test environment. After local cloud was deployed, it requests username and

password for authentication and authorization. We got two way of login page. One is demo page and the other one is admin page which is required for administrator only.

**Local cloud interface**

When the administrator login to page or cloud, there are different resource types assigned for the next creation of virtual machines. From the figure below we can understand resource type like instances used, vCPUs given, RAM used, floating Ips, security groups, volumes used and volumes storage size have been given for creating instances and deploying applications.

**Local cloud interface overview**

Local cloud deployed supports creating instances and booting instances using ISO images. The instance should be functional if the image boot. Ubuntu iso image was used. Instance name is the name of the new instance created in the cloud. After the instance is successfully launched, it is connected to the instance using remote console to boot the system and run.

**Creating an instance on local cloud**

We could list with all volumes in the local cloud system. In this list, we could find the volume that is attached to ISO image created instance. Volume is uploaded to glance. The volume has ID of the volumethatattachedtoISOcreatedinstance. Whentheimageissuccessfullyuploaded, itispossible to use the new image to bootinstances.

**Creating volume on local cloud**

To run instances and tenants, configuration of networking has important role. It used for communication of each virtual machine when we access instance image.

**Creating network on local cloud**

After the network is created, we get network topology which shows the private network created and connected to public network. Router is also created to facilitate flow of data to external network.

**VI. Implementation and Analysis**

To do this research, it is necessary to identify and choose parameters during experimentation and implementation phase. Requirements have been chosen, in order to measure the performance of cloud providers. For instance, specification of instance of virtual machines for selected cloud providers. The resource characteristics for the instance types offered to select clouds have been identified. We specify computing resource and requirements for testing all clouds with the same application. Table 5.1 indicates the information which is briefly discussed. The price calculations are based on the region in which cloud instance is running. Because, when the distance of data center increase there is an increase of cost on the user since bandwidth has been added to the dedicated virtual machines.

**Table 3: Computing resource and performance link**

Provider name	Instance name	CPU core/s	RAM size (GB)	Harddisk (GB)	Performance
Local cloud	M1.medium	3	5	250	<a href="#">View</a>
Amazon EC2	M1.medium	3	5	250	<a href="#">View</a>
Windows Azure	M1.medium	3	5	250	<a href="#">View</a>

Table 3 shows local cloud and other two cloud providers which have been tested using the same computing resource. Performance column has been classified into cost and performance metrics which were discussed in the table 3 and table 4

**Table 4: Price of instance for each provider**

Provider name	Instance name	Price Per Hour
Local cloud	M1.medium	0.062\$
Amazon EC2	M1.medium	0.0712\$
Windows Azure	M1.medium	0.0701\$

**Local cloud Performance analysis**

Depending on the resource given, we gather different performance. The performance got from several different cloud providers by the same computing resource except cost is displayed in the following table, which is substituted in the implementation by a name called view under the column of performance in the table5.

**Table 5:** Performance description of local cloud

Provider name	Throughput	Response time	Latency
Local cloud	0.0041	0.05	0.00114
Amazon EC2	0.0034	0.032	0.0032
Windows Azure	0.0023	0.021	0.0026

**Requirements used for testing local cloud, Amazon EC2 and Windows Azure**

To come over the research work and convert the approach taken, various requirements are decided. CPU core, RAM size and hard disk are computing resources studied. Instance type, number of users, time to run and type of application are factors used to identify difference providers' performance. This methodology is very important especially for those users who need to know detail performance of cloud providers' service.

So, for different cloud providers, the researcher took the same resource type, similar requirement factors and one type of application. When testing has been processed, we took number of users based on user base. Since different users of cloud computing come from different locations, our system gives distributive testing. For this reason, we categorize our selective testing by location of users.

**Table 6:** Requirements to measure cloud provider performance

Provider name	Instance name	No. of users	Time to run	Application type
Local cloud	M1.medium	7	5 minutes	Network bound
Amazon EC2	M1.medium	7	5 minutes	Network bound
Windows Azure	M1.medium	7	5 minutes	Network bound

**User interface**

According to the proposed algorithm, we implemented STFCC interface. The web page designed solves problems of users by providing interactive interface with any cloud computing. Therefore, anybody can test cloud computing based on testing requirements. The system gives different information about any kinds of cloud computing. For example, any cloud providers which gives infrastructure as a service has been explained with their properties like hypervisor they support, types of operating system, location of data center and deployment model. Our first interface provides interactive system between user and CC. We can get cost estimation for type of virtual machine selected.

When new cloud provider developed, the system accepts and evaluates performance of virtual machines. After provider performance tested new provider information added to database. Since database refreshed according to set time, any user those come to test cloud will get performance of newly added cloud provider. The following interface requests new provider to full the requirement before join to CID.



**Figure 5:** Register new cloud providers

**Testing requirements**

For users of cloud, the system gives service of selective testing based on the requirements of the users. The system requests different requirements for the purpose of testing different cloud providers. After all actions are taken, the performance of cloud is displayed by the graph or table form which shows throughput, response time, latency and cost. Every cloud providers' performance data should be collected and stored into CID. To get this, STFCC should be registered for any new cloud provider as discussed in the figure 5-2. Their performance and every requirement and computing resource is measured and analyzed by using VMM. The interface below shows what types of requirement should be included at the time of testing and it is must to add to CID.

**Medium sized instances result**

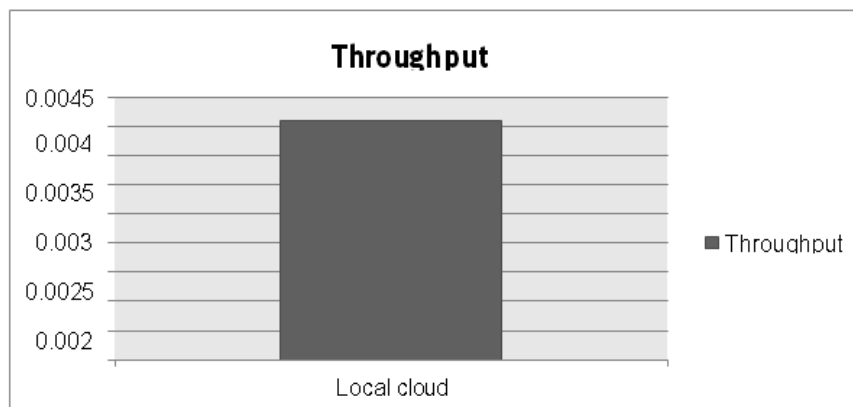
In this experiment, m1.medium instances were tested at the same time, having the same compute resource and similar requirements. This helps to investigate the network performance when various providers use similar VMs and also used resource at the same time. We have taken medium

**Figure 6:** Requirements to test cloud provider

sized VMs for testing different provider service performance. Based on the medium sized instances, we has seen performance and cost difference in the table 5.2 and in the table 5.3.

**Throughput analysis and results of local cloud**

We have local cloud, which was deployed in lab test environment. In this case, different test has been taken and got performance metrics. From those all, the analysis shows that throughput is one of them. Number of users sent request to the virtual machines through interface to analyze throughput. Therefore, we run seven numbers of users to upload and download data. Type of application processed were network traffic to see flow of traffic and how the cloud has been stressed and transfer throughput. The figure 5.3 depicts throughput generated by local cloud based on the requirements given.



**Figure 7:** Throughput analysis of Local cloud

**Response time analysis and results of local cloud**

The result of response time that was tested in laboratory environment from local cloud shows that 0.05ms, which did seven numbers of users, test for 5 minutes. Application was tested using uploads and downloads of data from virtual machines. The application type was network traffic and up on a test, it indicates there is slow packet flow and the result shows that there is problem on response time when it was compared with Amazon EC2.

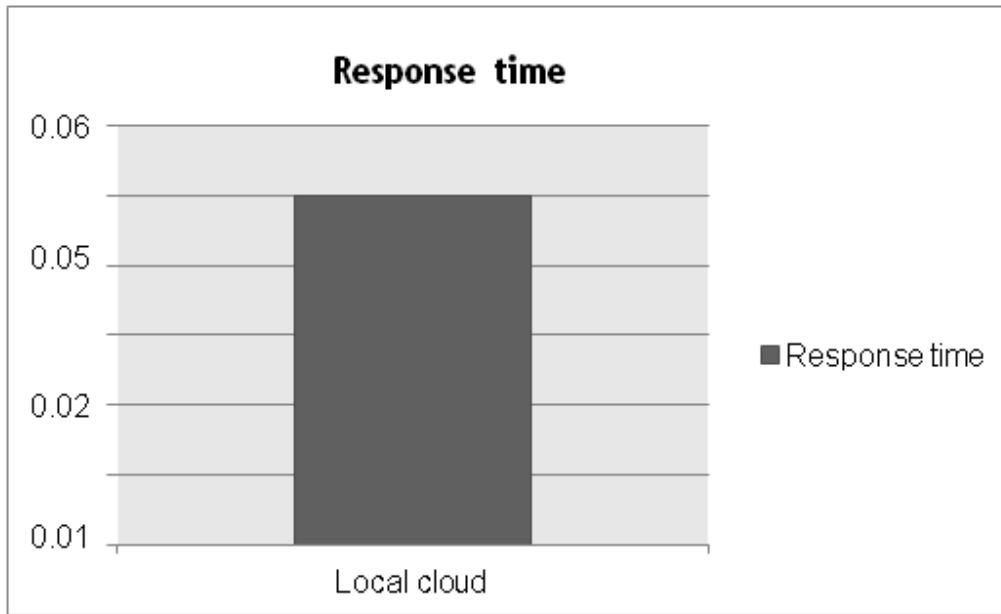


Figure 8: Response time of Local cloud

**Throughput comparison of local cloud, Amazon and Windows Azure**

Throughput has been measured by placing an application on virtual machines. In this study throughput analyzed by deploying local cloud and generating data from database for the purpose of showing sample difference particularly for other cloud. As the figure 5.5 indicates throughput of LocalcloudishighwhencomparedtoAmazonEC2andWindowsAzure.Inthecaseoflocalcloud, the performance generated is high. The number of request per time determine throughput of cloud. So, the time against each activity or process were taken and written down to measure and analyze request of workload per given time to conclude throughput of eachcloud. On the same computing resource and requirements, workloads were parallel processed. The times againsteachprocesswasdeterminedandmeasuredbysingandrequestingapplication, whichstored on virtual machines. Finally, throughput has been generated. Figure 5-depicts the throughput measured every 5 minutes for 7 clients or users. We analyzed and concluded that the throughput of Local cloud is better than the other cloud environment as shown in Figure 9.

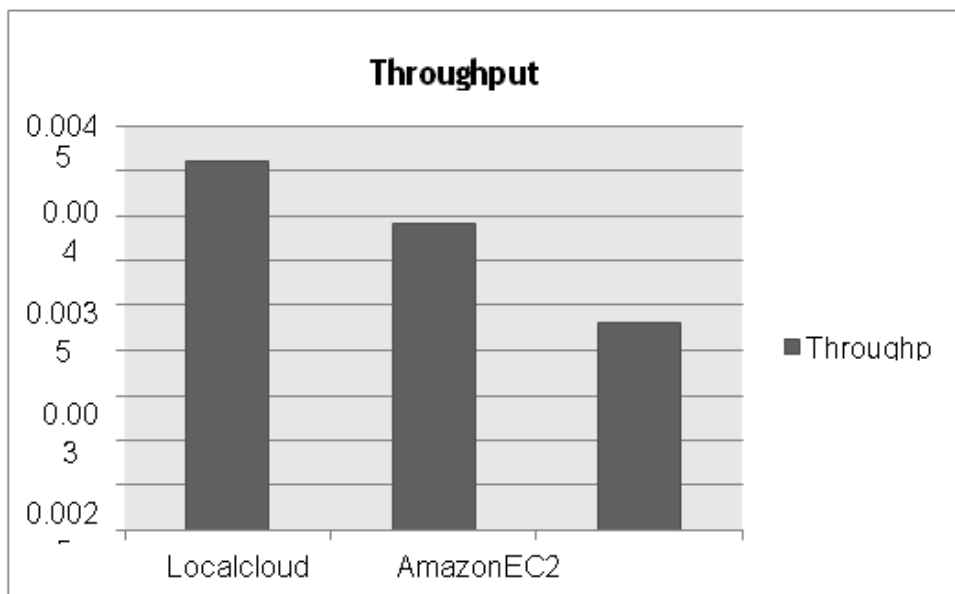
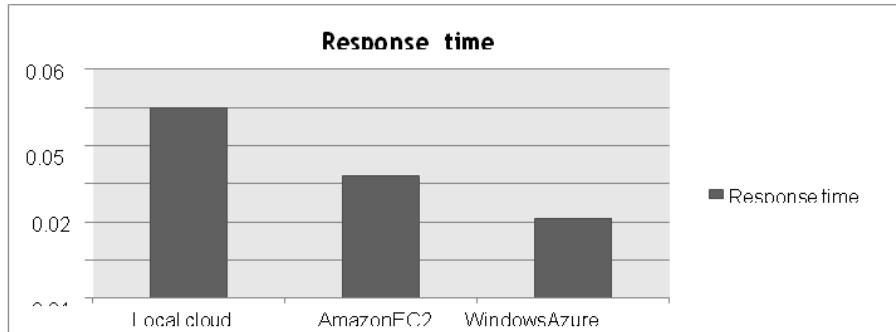


Figure 9: Comparison of cloud provider throughput

**Response time comparison of local cloud, Amazon and Windows Azure**

To see the performance of different clouds, we have local cloud, which has been deployed in lab test environment. Windows Azure has been taken for comparison as a sample from database. This comparison indicates that windows Azure takes less time to response for the requested data. The application of network bound was made to run. This means in windows Azure cloud provider, server is not bottlenecked compared to the other two clouds for quick response. Amazon EC2 is fast in response time than local cloud. Therefore, the user can decide which cloud computing is best for them according to their requirements and depending on the given testing as a service

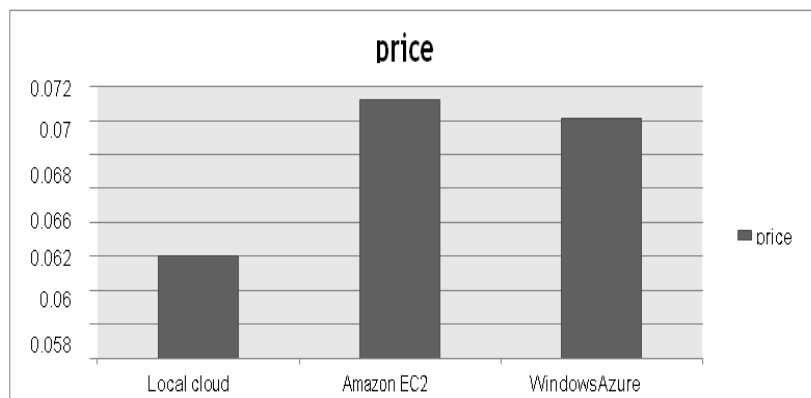


**Figure 10:** Comparison of cloud provider response time

If we compare and contrast the execution times in Tables 5.1 for workload 1-7, the execution time shows difference for the same computing resources for the local cloud, Amazon EC2 and Windows Azure environment. However, the response time was twice as high for the Windows Azure, i.e. it takes 0.021 seconds to process 7 parallel requests in 5 minutes test in the cloud environment. For Amazon EC2, it took 0.032 and 0.05 for Local cloud to do the same thing.

**Price comparison of local cloud, Amazon EC2 and Windows Azure**

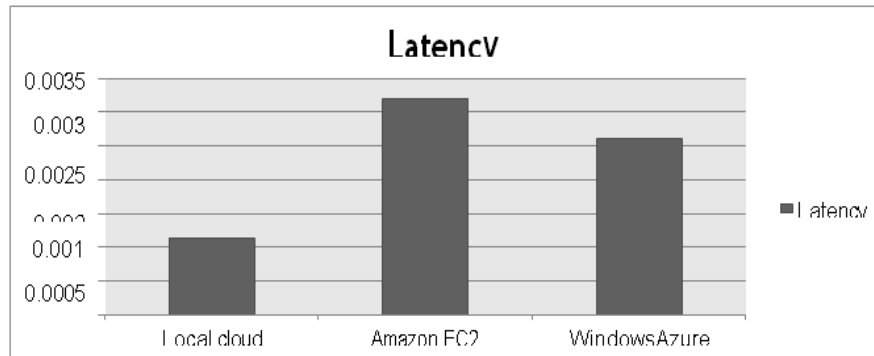
Any users of cloud computing contracts with cloud provider for cost pricing. Providers of cloud disallow accessing the data for corporate gain if there is no agreement and no payment is made for the resources. In our work, local cloud includes dedicated CPU cores, dedicated RAM, hard disk and others. This could be known only by testing the service of cloud providers. The VM configuration tested was m1.medium instance. All details of providers' name, instance name and cost per hour have been discussed in the table 5.3. From the scenario, we conclude Local cloud price is lower than Amazon EC2 and Windows Azure.



**Figure 11:** Comparison of cloud provider cost price

**Latency comparison of Local cloud, Amazon EC2 and Windows Azure**

The figure 5-9 illustrates packet delay for VMs on different cloud providers when they were configured with the same computing resource (i.e. CPU core, ram and hard disk). Lower latency shows packets arrived at faster speed to their endpoint. This result is confirmed by the previous throughput discussion which showed local cloud has higher throughput implies lower packet delay or latency.



**Figure 12:** Latency comparison of Local cloud Amazon EC2 and Windows Azure

Amazon EC2 and Windows Azure show great packet loss when compared with local cloud. This indicates that if there is high throughput, there is slow response time. Comparing latency has to be related with response time. High response time indicates minimized delay and packet loss.

### VII. Discussions

Some research has been done to test performance of cloud computing. The study brings different approaches and algorithms. Most of approaches are linked to deploying cloud computing on local environment to know and evaluate performance. They took different parameters and requirements. In other research, selective testing for SaaS has been done using different tools. All of the research done focused on testing performance of cloud computing. Our study differentiate from all of research done by providing testing as a service for cloud computing. As explained throughout this paper, our study based on giving testing services for cloud providers in different aspects such as types of cloud deployment model, hypervisor, service type, location of data enter and kind operating system supported. Additionally, we test throughput, response time, latency and cost. Every cloud providers tested before included to CID. According to our approaches and algorithms, new cloud providers tested and evaluated by registering to VVM. Then VMM measures their performance and saves to database. The algorithm developed performs on the same computing resource, the same number of users and requirements. Users are located in different locations. For this reason, our step of testing considered distributive testing. The previous testing system is not distributive testing.

### VIII. Conclusions

Owing to its importance and impact of selective testing and performance evaluations, there is an ongoing research that is in advance in cloud computing. As new innovations arrive, it is equivalent to evaluate new technologies applied on cloud computing environment to enhance the correct understanding about new technology and giving testing of cloud performance status.

This study investigated the possible testing as a service for cloud computing. Several steps and procedures are applied to test cloud computing. In the beginning local cloud deployed for prototype and to analyze its performance based on our requirements. On the virtual machines of local cloud we put one file application. After that users permitted to access the application that stored on virtual machines. This is done by the process of request and response. When this activities happen we able to evaluate performance of local cloud. This experiment was carried out to show a performance of cloud, highlighting steps involved in setting up a cloud laboratory environment for testing and evaluations. It gives value to demonstrate the efficiency, benefits and result obtained from cloud technology.

The evaluation was done using instance of local cloud to test cloud performance by deploying application on local cloud environment. For this purpose, we have conducted seven clients that request and get response. When this type of experiment was applied, we observed PKI (i.e. throughput, response time and latency). Results gathered from the experiment were analyzed and represented in table and graph form. The algorithm was implemented and tested using designing selective testing interface with single cloud database and deploying local cloud environment by using open source cloud provider. From local cloud lab test, we get real PKI and compare with Amazon EC2 and Window Azure cloud providers. The providers PKI got from gathered information on line, specially computing resource and converting them into algorithm. Using AppManager, Amazon EC2 was tested and response time was measured. In the case of Amazon performance, we uploaded one file on instance of it, observed what have been going on, analyzed, and captured how much time the response time would take to process a response. Therefore, based on the response time, we concluded that response time of Amazon EC2 was higher than local cloud and slower than Windows Azure.

We found that throughput of local cloud and Amazon EC2 by testing on laboratory environment. Window Azure performance was attained by changing the approaches to implementations. All clouds was tested by workload of one up to seven users. This test was done by the interval of five minutes based on constant computing resources and requirements. The analyzed result got from both experimentation and implementation. So, data obtained from algorithm and experiment show that throughout of local cloud is higher than Amazon EC2 and Windows Azure. Throughput of Amazon EC2 was slower than local cloud and higher than Windows Azure. It is possible to conclude that local cloud performs and give best performance for users when we test using similar applications and requirements. Response time of cloud was measured with unique computing resources and requirements. From the results displayed, the response time of Windows Azure is higher than Local cloud and Amazon EC2. On top of this, the study adds testing cloud providers pricing techniques. The study was designed and reached algorithmic approaches based on providers' pricing policy and we got that by the same computing resource, it was good to display and measure cloud infrastructure as well as other cost assigning system. Cost data gathered from different sources of cloud providers. The data helps us to compare and contrast Local cloud, Amazon and Window Azure price. Since providers have their own way of cost calculation methods, selective testing gives system of selecting provider payment based on the users that match with their interest. In this study, local cloud cost is lower than Amazon EC2 and Windows Azure. This difference is based on the policy of cloud providers' amendment. Generally, selective testing for cloud performance in cloud computing is big, bold, interesting and attractive research study based on the result and advantages that give for the cloud users. It is prerequisite for those users, enterprises and organizations going to subscribe their applications in cloud computing. The prerequisite testing clouds may save users from the risk of prospective performance Degradation

### **IX. Future work**

In order to generalize the results, we need to perform the same experiment on some cloud providers. Since the data obtained from the experiment presented only three cloud providers, it is difficult to generalize the results to other available cloud computing environments; for example, the performances of other clouds like Rackspace and Open nebula. Suggested further research work on the following areas: Test performance of cloud computing by deploying other cloud provider on physical environment, Test all types of instance then measure and analyze under congested environment. Work on testing as a service; generate other PKI like time to live, jitter, etc. Study how to optimize performance come with performance difference with cloud providers by assigning the same computing resource.

### **References**

- [1]. P. Mell, T. Grance and others, "The NIST definition of cloud computing," National institute of standards and technology, vol. 53, p. 50, 2009.
- [2]. H. Yang and M. Tate, "A descriptive literature review and classification of cloud computing research.," CAIS, vol. 31, p. 2, 2012.
- [3]. M. Khan and M. F. Amin, Web Server Performance Evaluation in Cloud Computing and Local Environment, 2012.
- [4]. R. Buyya, C. S. Yeo and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on, 2008.
- [5]. M. S. A. Kolte and P. E. Ajmire, "A Survey-Cloud Computing".
- [6]. K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman and N. J. Wright, "Performance analysis of high performance computing applications on the amazon web services cloud," in 2nd IEEE international conference on cloud computing technology and science, 2010.
- [7]. B. Kepes, "Understanding The Cloud Computing Stack SaaS, Paas, IaaS, 2011," URL: <http://www.rackspace.com/knowledgecenter/whitepaper/understanding-thecloud-computing-stack-saas-paas-iaas>.
- [8]. S. Jagannathan, Comparison and Evaluation of Open-Source Cloud Management Software, 2012.
- [9]. M. O'Neill, Connecting to the cloud, Part 1: Leverage the cloud in applications. Retrieved Nov. 9, 2010, 2009.
- [10]. T. Merrill and T. Kang, "Cloud Computing: Is Your Company Weighing Both Benefits & Risks?," Ace Group, 2014.
- [11]. M. D. Tarapra, M. K. Jadeja, J. Jeba Praba and R. Sridaran, "Classification of Performance Degradation Issues In Virtualized Cloud Environments".
- [12]. A. Heads, "Subject Page No.," 2014.
- [13]. S. Addamani and A. Basu, "Performance analysis of cloud computing platform," International Journal of Applied Information Systems ISSN, pp. 2249-868, 2012.
- [14]. S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, 2010.

Garamu Tilahun Itichaa. " Selective Testing As A Service for Cloud Computing" IOSR Journal of Computer Engineering (IOSR-JCE) 21.3 (2019): 70-85.