International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 04 | April-2016 www.irjet.net p-ISSN: 2395-0072

TESTING METHODS BASED ON MEASURING THE CLOUD IN THE CLOUD ENVIRONMENT

Dr.V.NETHAJI[1], Dr.M.DURAIRAJ[2]

[1]Senior Test Engineer, American Megatrends, Chennai, TamilNadu, India.
[2]Solution Architect, Anuta Networks, Bangalore, Karnataka, India.

---***---

Abstract- *Cloud consumers utilize the cloud because it saves money. They like that they can buy only what they need, and elastically expand and contract their purchase on demand. They can also deliver better quality of experience to their users by choosing a cloud provider that has data centers in close proximity to users. The cloud can provide higher availability due to the simple fact that cloud providers are experts in operating large data centers. But, there is no denying the fact that cloud consumers fear the loss of control inherent to using the cloud. To help consumers overcome this fear, the cloud carrier and cloud provider must provide information about the cloud, especially information on users' quality of experience. Measuring is dependent upon the cloud carrier working with the cloud provider to test the cloud. And together, they must make what they find available to their users. When it comes to user experience, the cloud must be transparent, not murky.*

Key Words: *Cloud consumers, IaaS, Paas, Saas, Carrier, Platform and Cloud Provider.*

1. INTRODUCTION

Broadband, backbone and mobile wireless service providers see the cloud as an opportunity for growth. New applications running in the cloud drive new traffic to their network. The cloud also offers new revenues from new services that can be sold to customers[7]. The business models for carriers and how they relate to the cloud are developing quickly, but one fact is attractive clear to achieve success in the cloud market, carriers must aggressively ensure that they offer a high quality of service to cloud consumers. The best way to assess cloud quality is to test the cloud.

A cloud is a shared computing platform available over the network used to run a variety of business or personal applications. The concept is hardly new; it has roots in service bureaus, outsourced data centers and utility computing. What makes the cloud work today is the rise of the web browser as a thin client that allows individual users to run any application, the wide availability of high bandwidth networks, and virtualization technology for computers, storage and networking. The cost savings, expanded reach, and improved quality gained by running an application in the cloud is proving to be a business success, as shown by the growth of cloud services into a market worth billions.

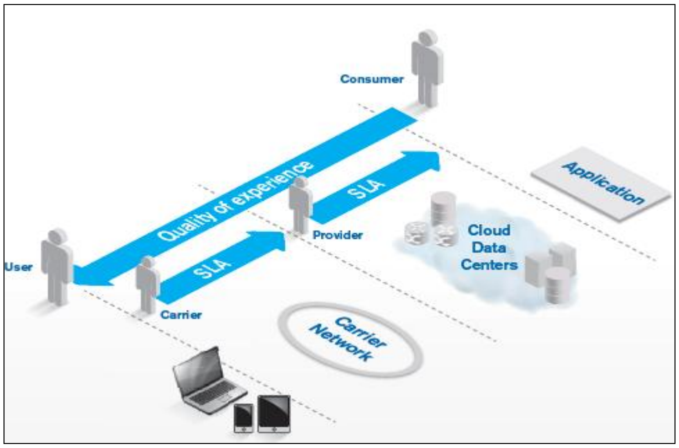
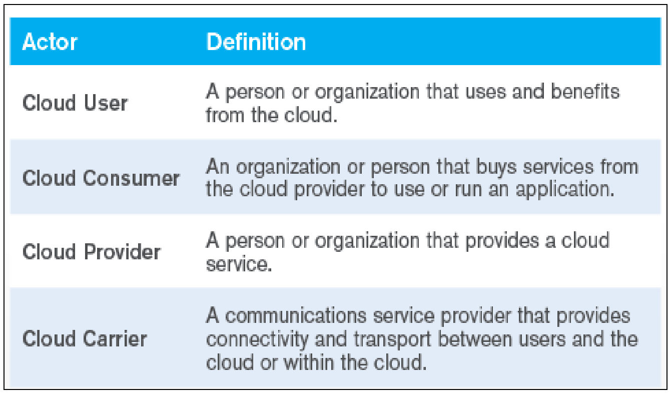


Figure-1: Cloud players: cloud users, cloud consumers, cloud providers and cloud carriers

Cloud services are sold to cloud consumers who have a business need. To meet that need, the cloud consumer deploys an application to be run in the cloud for a user community. The cloud itself is driven by cloud data centers that provide an environment for running the application [10]. The data centers provide high technology servers, vast storage space and latest networking. User access to the cloud data centers is provided by cloud carriers. The cloud provider manages the cloud data centers and their servers. The cloud carrier manages the interconnection between the user and the cloud data centers. The application may be owned and managed by the cloud consumer or the cloud provider.

The quality of the user's experience depends on both the carrier and the cloud provider. The carrier manages latency, reachability, bandwidth, loss and other network key performance indicators that affect quality. The cloud provider manages processor utilization, switch utilization, storage and other resource KPIs that affect quality. But the KPIs that describe the quality of the service (QoS) provided to the consumer and the user are web download times, service availability, data delivery times, and other KPIs that are tied more directly to the service sold to cloud consumers. These serviceoriented KPIs cannot be measured by any one actor; together, they ether is determined by the business relationship between them. If they are independent, cloud carriers offer SLA guarantees to cloud providers. Cloud providers in turn offer SLAs to cloud consumers. If the carrier owns the cloud provider, then the single organization can offer the service-level agreements (SLA). These SLAs are one factor driving carriers and providers to test the cloud.

Table-1: Cloud actors



Actor	Definition
Cloud User	A person or organization that uses and benefits from the cloud.
Cloud Consumer	An organization or person that buys services from the cloud provider to use or run an application.
Cloud Provider	A person or organization that provides a cloud service.
Cloud Carrier	A communications service provider that provides connectivity and transport between users and the cloud or within the cloud.

In this complex business environment, carriers have strengths. First, they own the network that connects users to the application running in the cloud. Second, they provide the circuits that connect cloud data centers to the Internet and each other [5]. Third, they can provide the security and privacy customers want through dedicated circuits or virtual private networks that isolate one customer's traffic from another's. Finally, they know how to offer a high-quality service backed by the guarantees of an SLA.

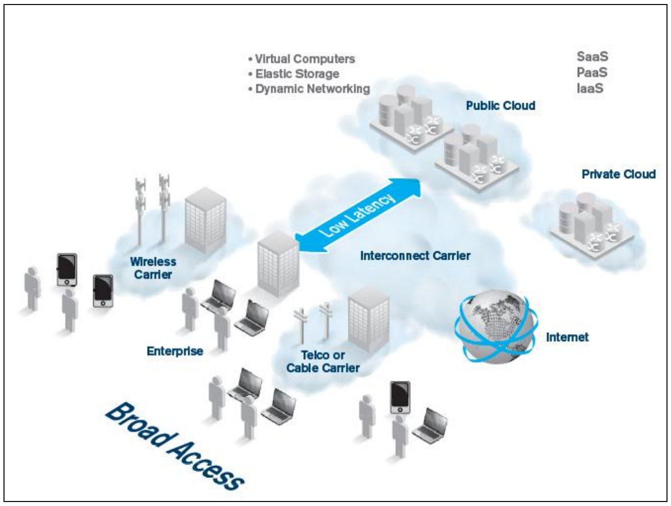


Figure-2: Broad network access in cloud computing

2. WHAT THE CLOUD OFFERS

Why the cloud consumers build applications in the cloud. The National Institute of Standards (NIST) has identified five essential characteristics of a cloud. These will provide a good summary of the benefits that cloud consumers.

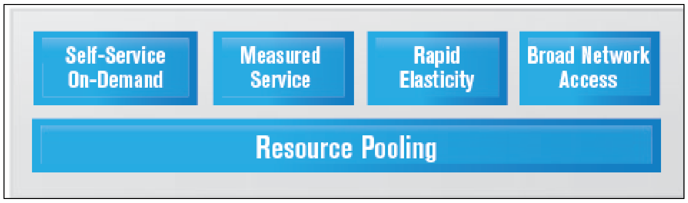


Figure-3: The essential characteristics of cloud computing

Pooling resources to be shared among many cloud consumers yields cost savings for all. Cloud consumers need to pay what are the resources they are going to use. Instead of buying a rack of servers that end up sitting idle most of the time, the cloud consumer can lease only what they need[6]. The resources available in the cloud data centers generally include racks of central processing units (CPUs) providing virtual servers on demand, racks of disks providing elastic storage on demand, and a data center network interconnection providing addresses and network capacity on demand. Cloud consumers control the quantity of resources they lease using a self-service administrative interface, and can expand or contract the resources as needed. Usage is measured to bill the cloud consumer for the resources used. This increase in flexibility enables cloud consumers to take control of their expenses.

Applications can support a broad user community over the Internet. To support Internet scale applications, cloud data centers need high bandwidth ranges and low-latency connections for Internet exchanges.

A high bandwidth connection to the data center helps applications scale to large numbers of users. Till now a fast-connection distant data center won't perform well. For most applications, latency has a bigger impact on user-perceived performance. Latency depends on distance due to the speed of light. It is therefore important to place applications in data centers located in close geographic proximity to users to ensure that data is efficiently routed between the data center and the user. Most enterprises would have a difficult time building data centers in the US, Europe, and Far East that would be capable of supporting a worldwide user base. But by hiring servers in the cloud, enterprises can afford to deploy applications in close proximity to their users, wherever they happen to be located. Cloud carriers are already facing the latency problem and have placed the data centers they operate close to users [8]. There are some applications for which throughput are more important than latency. Any user transfers large amounts of data between the application and the user will be sensitive to the available bandwidth. Streaming video and data backup services are examples of bandwidth-sensitive applications. A cloud data center with a high-bandwidth connection to the core of the Internet will be at an advantage in serving these types of applications.

Regardless of how good the cloud application is, it is of little use if unavailable to users. The cloud consumer can use the cloud to deploy the application at multiple sites that can be isolated to prevent any failure at one site from affecting the other sites. But, this introduces the technical problem of synchronizing the application's state and the user's state across the cloud. If a user stores a file in the cloud at one data center, it needs to be made readily available in the data centers serving that cloud-based storage application. Resolutions to this technical problem normally need high-performance and high-quality connections between the cloud data centers.

Cloud carriers can meet the unique needs of applications by mapping the applications' traffic into specific classes of service (CoS), each with its own bandwidth and traffic-handling policies. Pretty a bit determination has been made in the standards to define some typical CoS and the appropriate performance objectives that they should meet. In specific, the Metro Ethernet Forum (MEF) has developed MEF. Which includes markings and objectives for three CoS that apply to cloud carriers. Most haulers are aware with providing CoSaware services and are comfortable with providing CoS-aware SLAs.

While there are amply of advantages to installing applications in the cloud, most cloud consumers fear the loss of control, and the potential security and privacy breaches that moving to the cloud can bring[1].The fear of losing control can best be overcome by transparency, e.g., by providing the customer with the information they feel they need. Information about the number of users, how frequently they use the service, the resources consumed and the charges accrued are all examples of data that should be provided in any measured service. When combined with effective controls concerning who is allowed to use the service, the administration capabilities and the customer's ability to configure the services on demand will go a long way to addressing the perceived loss of control.

Privacy and security concerns can be partly met using virtualization techniques such as virtual private networks (VPNs) and virtual machines (VMs) to isolate one cloud consumer from another. Carriers, of course, are very familiar with providing VPNs and dedicated circuits as needed to ensure customer privacy. VMs are similar, and can be used to guard the integrity and privacy of servers and storage in the cloud. Security on the entire must be addressed as a full problem, and all components in the solution must be portion of that solution.

3. TYPES OF CLOUDS

The NIST Model of Cloud Computing describes three types of clouds: infrastructure as a service (IaaS), platform as a service(PaaS) and software as a service (SaaS). These three types of service offerings are a good way to layer the services, for which there are many existing examples.

In IaaS, the cloud provider sells simple servers with little or no software. The servers may come with a virtual machine and an operating system, network connections with attached storage ,IP addresses and DNS names. The cloud consumer provides all the software needed by the application. An example of an IaaS provider is Amazon's EC2 (Elastic Compute Cloud) service. Other IaaS vendors include Savvis, Rackspace, IBM, GoGrid and Terremark.

In PaaS, the cloud provider sells a software platform to make application development easy. Platforms can include services like a web server, a database, shopping carts and content management tools (like Wordpress or Drupal). The value provided by PaaS is that cloud consumers can develop their applications more quickly on these middleware services than on bare infrastructure. An example of a PaaS provider is Google's App Engine. App Engine provides a web server, No SQL and SQL databases, storage services and much more.

Key PaaS providers are cloud storage services like Amazon's S3(storage service). Cloud storage provides a simple web based application programmable interface (API) with replication [4],which makes it easy to develop sophisticated device backup and file sharing services. Apps that "sync" mobile devices to cloud-based storage or that share photos via the cloud are enabled by cloud storage services.

In SaaS, the cloud provider sells the complete application. The cloud consumer outsources the entire application to the cloud provider, and grants users access to their specific instance of the application. Almost any application can be delivered as SaaS. An example is Salesforce.com, which provides a popular Customer Relationship Management (CRM) application. Other examples include hosted e-mail applications from Gmail, Yahoo and Microsoft, ERP applications from SAP, and hosted private branch exchange(PBX) from XO Communications. Just about any application can be provided as a SaaS.

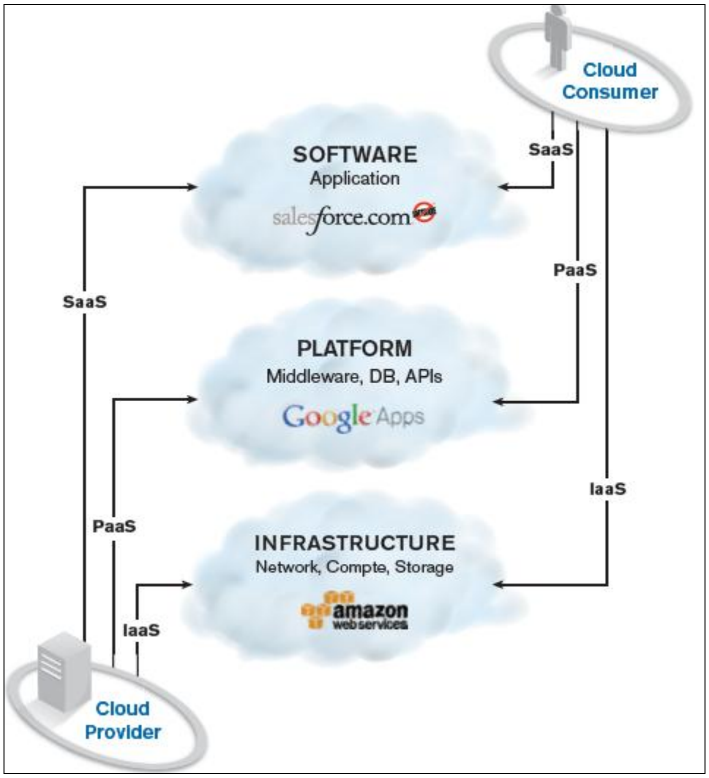


Figure-4: The three main types of cloud services: IaaS, PaaS and SaaS

4. DEPLOYMENT AND OWNERSHIP MODELS

The NIST model of cloud computing describes three models for cloud deployment and cloud ownership, and in part, some basic business models designate how cloud carriers, cloud providers and cloud consumers relate to one another. The three deployment models are private clouds, public clouds and hybrid clouds.

In private clouds, an enterprise buys its own equipment, and builds and operates its own data centers using the same server, networking and storage technologies as the public cloud. The users are mainly employees of the enterprises, connecting to the private cloud via a VPN for private and secure access.

In public clouds, the cloud provider is a separate entity selling services to many cloud consumers. The cloud provider is connected to the public Internet, and hosts applications are accessible to anyone on the Internet. Public cloud providers offer access to the broadcast to the possible community of users.

In between these extremes lie hybrid clouds, in which some components of this model are public, and others are private. For example, a cloud-based banking application may have a public webbased interface to Internet users that is hosted on a public PaaS cloud, with a private connection to back-end servers running on a private infrastructure cloud that is located entirely within the bank's private network [9].

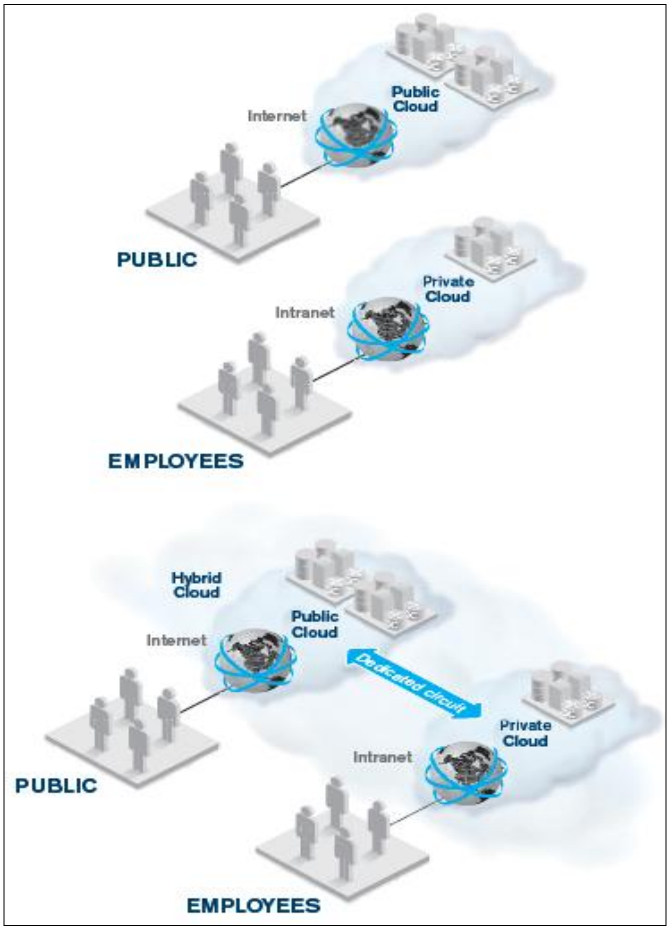


Figure-5: Cloud deployment models: public, private and hybrid

5.MEASURING THE CLOUD

Ultimately, the cloud consumer owns the application and therefore cares about users' quality of experience. Together, the cloud provider and carrier determine how well those expectations are met. If the network is slow, data centers are offline or the servers are overloaded, access to the application will be hampered and users will be affected. To guard against poor quality, providers must monitor the cloud's performance through measurement of availability, correctness and responsiveness. When providers have the appropriate measurements on hand, they can use this information to detect problems, predict when they will need to expand the capacity, and perhaps most importantly, prove to cloud consumers that their service quality expectations and guarantees have been met.

The specific measurements needed depend on the service being offered to the cloud consumers, and the business relationship between the cloud carrier and the cloud provider. The sections below describe the measurements of quality applicable to carriers, IaaS ,PaaS and SaaS providers.

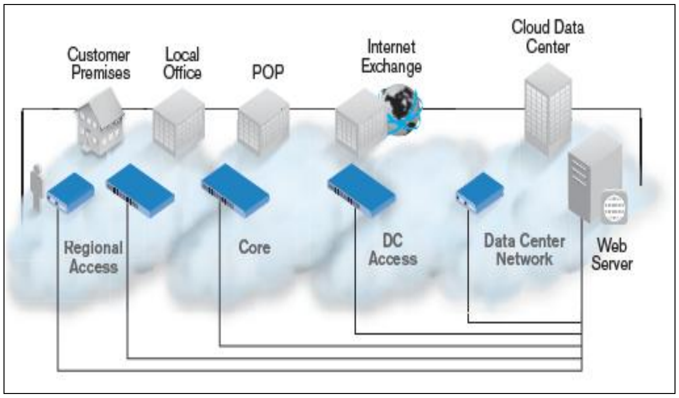


Figure-6: Testing cloud services from the carrier's network

5.1Measuring Carrier Services

Cloud carriers have been selling data connectivity services for decades, and as such, the familiar measurements provided by testing the availability, latency, loss, delay variation and throughput of the connections all apply. The location from which the tests are conducted depends largely on how the cloud is deployed and who its users are.

For a public cloud providing services to the general public, wireless provider might enter into a contract with the cloud provider to offer its subscribers fast access to the nearest cloud data center. The wireless provider could conduct tests from each 3G or 4G mobile switching center (MSC) to the nearest data center [2]. The tests serve to verify the availability, packet loss, network latency and throughput between the testing MSC and the data center. For cable and wireline carriers, the same kinds of tests would be conducted from headend sand central offices.

The idea is to test access to a public cloud from as near as possible to the user as is practical. Certainly, testing can be conducted from fixed locations like the home or small business using dedicated test devices. It is even possible to run tests from mobile test units or the user'sphone or tablet. These examples have the advantages of being able to test the last mile, but also have the disadvantages of having to test potentially huge number of devices. For cost reasons, most carriers would opt to test from a subset of the users, but this introduces its own set of problems involving signing up, selecting and managing representative population of users. In any case, carriers who own the network that is used by public users to access the cloud are in the best position to measure service quality from the perspective of many users.

Carriers also sell dedicated L2 or L3 circuits to cloud providers. Dedicated circuits between distributed data centers owned by a public cloud provider are used to synchronize the application state between the data centers. Circuits from private or public data centers to Internet peering locations are used to connect the data centers to the Internet. Dedicated circuits from private cloud data center to the public cloud data center can be used to build a hybrid cloud. VPNs that run over the Internet can be used to provide a virtual private cloud that actually runs on a public or hybrid cloud.

These circuits carry multiple CoS, each with its own QoS markings, provisioned CIR and EIR, and its own service-quality objectives. Testing and monitoring the circuit implies testing and monitoring each CoS. Regardless of the types of circuits provided by the carrier, the same basic measurements are needed. The measurements maybe use the different testing protocols or operate at different layers, but the metrics are similar. Carriers conduct tests to measure availability, loss, latency and delay variation (or jitter) on a 24/7 basis. They monitor utilization of the circuit and each CoS around the clock, and may also monitor which applications are using the circuit [4]. They also perform tests to confirm that the provisioned bandwidth is actually available whenever the circuit bandwidth. Please note that on-demand elastic bandwidth provisioning on these circuits implies that these "turn-up" bandwidth tests need to be run on a frequent and automatic basis. The days of T1 circuits being set up and left untouched for years are long past. Certainly, changes are now so frequent that carriers may wish to periodically retest the circuit's throughput.

While the carrier's network performance is vital, it does not represent the whole picture. The user's awareness of the service's quality is dependent upon both the carrier's network and the cloud provider's data centers. To see the cloud services, we need to test further upthe stack, as described in the following sections.

5.2 Measuring Infrastructure Services (IaaS)

A cloud IaaS provider sells basic infrastructure components of a VM on which software can be run, in addition to storage for data and a local network enabling communications within the cloud data center. Each of these infrastructure resources must be managed and monitored.

There are existing tools in the marketplace that can be used to manage VM. These tools monitor the number of VMs being run, and the CPU, memory, I/O and network resources they are using. They also measure which operating systems are being run on the

VMs, in addition to the middleware and applications running on those virtual machines. Similar tools are also available to manage storage, and to monitor how much storage is used and which users are consuming the disk space. There are also tools available to manage and monitor the data center network, set up VLANs and routing domains, and measure the usage of ports, VLANs, routers and switches.

These infrastructure management and monitoring systems are absolutely critical to managing the infrastructure. However, they only address part of the problem. To measure service quality, the performance of the service needs to be tested from the user's perspective. This means testing from the locations within the carrier's network as close as practically possible to the users. The infrastructure services that should be tested include the DNS,data center network, server, application and web. Each of these services should be tested for availability and performance.

DNS provides a basic network name to address translation. As new VMs are brought online, they must be assigned network addresses. These addresses must be set to the DNS name used in the application. Users access VM by name. Routing users to the nearest cloud data center is accomplished through DNS.A problem in DNS (either a slow response or a wrong translation)can have significant impacts on the application. A very simple test to ensure that the DNS name can be looked up quickly and that it resolves to the right address will detect these problems. Large cloud data centers are deploying new routing technologies like TRILL and IEEE 802.1aq (shortest path bridging), in addition to new Software Defined Networks (SDN) like Open Flow in order to deploy a more scalable, dynamic local switching and routing network within the local network. A carrier's duty often ends at the switch, where a circuit plugs into the local data center network. Measuring the local switched network in the data center or the end-to-end route from the user to the server can be done with simple reachability tests. More sophisticated network tests, like Two-Way Active Measurement Protocol (TWAMP), can measure packet loss,delay variation, latency and other important measures of IP network performance and quality.

Cloud data centers are protected by firewalls and VPN servers, and use load balancers in the network. Firewalls protect the servers from attack. VPNs are used when the consumer desires private communications between the user and the cloud. Load balancers in the network front end serve a cluster of servers and balance the workload among them. Each of the network components affects the service offered in an IaaS cloud. Measuring their impact on performance can be done with differential testing, running one test of the service through the firewall, VPN or load balancer, and another directly to bypass the device.

To ensure that the virtual machines in the data centers are actually up and running an operating system, you can use simple tests like UDP Echo. Further tests can be run to verify that an application is running and listening on specific TCP or UDP ports within the OS, and that those ports are reachable by users across the network. The measurements provided include availability, and simple response and connection times.. To test those web servers, simple web requests must be made to the web servers running over HTTP/ HTTPS.

5.3 Measuring Platform Services (PaaS)

PaaS cloud providers build on the same kind of infrastructures as IaaS providers, adding the middleware that makes it easier to develop applications. Almost any middleware could be presented as a platform, but the most common are those services that are designed to make building websites or mobile apps easier [7]. All the testing and monitoring capabilities described for IaaS are also needed for PaaS providers. However, the PaaS provider needs to go one level up in the stack to fully test the service. Three examples of PaaS services are website development services, cloud storage services and mobile application development services.

Website development services provide a basic web server such as Apache, content management systems such as WordPress, and APIs such as Google's App Engine. These tools enable the cloud consumer to develop a better website, faster. The website test is performed by downloading the complete web page and measuring its availability, download time and response time and the throughput of the page and all of its content.

Mobile application development services simplify the process of building applications that run in mobile devices (e.g., phones or tablets), or in browser environments like Google's Chrome. While these services vary, many are REST APIs built on a web protocol for communications, and a web server running in the cloud. These applications issues HTTP GET and PUT operations to the right URL, transmitting data encoded in language like XML or JSON. Tests that measure the availability of the service, detect failures, and measure the service's response time and throughput simply issue the same API calls that an application would.

Cloud storage services allow applications to use persistent storage in the cloud. Cloud storage can be used for the applications such as data backup on a cell phone or synchronization of files between tablets and PCs, or secure file sharing applications. These services run over HTTP/S, and generally provide the ability to get and put files to the cloud. Because it can be time consuming to replicate the files to other sites, a key performance metric consists of measuring how long it takes to distribute updated files to all the servers. To test these services, get and put operations are issued to the cloud storage service. Testing replication involves writing a new file to one site and then getting the file from a different site, while measuring how long it takes the new file to propagate.

5.4 Measuring Application Services (SaaS)

An SaaS provider builds on the infrastructure and platform services described above, but delivers the complete application as a service. Measuring the service means measuring the application from the users' perspective. Many applications in the management tools measure the application server in traditional client/server architecture. This works fine when the users are close to the server or within the enterprise, but for the broad user community served by the public cloud, these traditional application management tools break down. Simply put, you have to manage the service, not the server. The specific measurements needed depend entirely on the application. The following examples illustrate this point.

For web-based software, such as a CRM application in the cloudlike Salesforce.com, the user interface is entirely within the web, and therefore the web-based testing described above works just fine. In some cases, testing the application requires more sophisticated application scripting, where a series of web pages must be navigated to perform an action that should be measured. But in all of these cases, the availability and response time of the application are the primary measurements to be taken. The nature of the cloud means that most applications provide a web front end. These applications can all be tested from sites in close proximity to users via the previously described types of web-testing methods.

Over-the-top video services such as YouTube, Hulu and Netflix are all delivered in the cloud. Although these applications are webbased, they use specialized video delivery protocols to download the video to the user's device. Tests on these services are performed by downloading a video using the appropriate delivery protocol. The tests measure the availability of the service, the response time of the application, the download time and the throughput experienced by the user. They also detect any episodes of rebuffering in which the delivered video data falls behind the playout, and other errors users would perceive as failures. In SaaS testing, it is all about the applications.

5.5. What's Common?

Across these four service layers (Carrier, IaaS, PaaS and SaaS),there are a few common trends.

First, testing from a location close to the user and located within the carrier's network provides the best measurement of the user's experience, which is perfectly logical. The user's QoE depends on the full network path from the user through to the access network, through the backbone to the nearest cloud data center, and all the way to the server hosting the application. Carriers own the network, and therefore also own the test locations. This means that they are equipped to perform the best testing.

Due to the variety of services offered through the cloud, a variety of tests need to be performed. Any testing solution offered must be a multiply solution—in other words, one capable of testing at all layers, from L2 to L7.

```
Step 1: Cloud-Testing is performed using the framework
Step 2: Given the Services_ID and Inimitable_ID, the first
field is obtained as input
Step 3: If the exchange of Testing(T) to be performed is get()
Step 4: Evaluate the frequencies of the Inimitable_ID
Step 5: Determine the number of time with which each
Inimitable_ID appears in the respective Service_ID
Step 6: Sum the Inimitable_IDs by summing the
frequencies appeared in the respective Service_ID
Step 7: If the exchange of validation to be performed
Step 8: Evaluate the frequencies of the Unique_ID
Step 9: Determine the number of time with which each
the Inimitable_ID appears in the respective Service_ID
Step 10: Sum the Inimitable_IDs by summing the
frequencies appeared in the respective Service_ID and
then sort them in that order.
```

The algorithm given above provides the detailed steps involved in the designing of advanced hash function to ensure cloud testing with the help of the framework. The input obtained is the Service_ID and User_ID. Thenext step consists of the exchange of testing performed with get() or get(). If the exchange of testing to be performed is get(), then the frequencies of the Inimitable id(Inimitable_ID) is evaluated. Then the number of time the Inimitable id appears in the respective segment is determined. Finally, a summation is performed and sort in the ascending order.

6. LIFECYCLE

The testing defined above must be performed throughout the lifecycle of the service, as shown in the figure below.

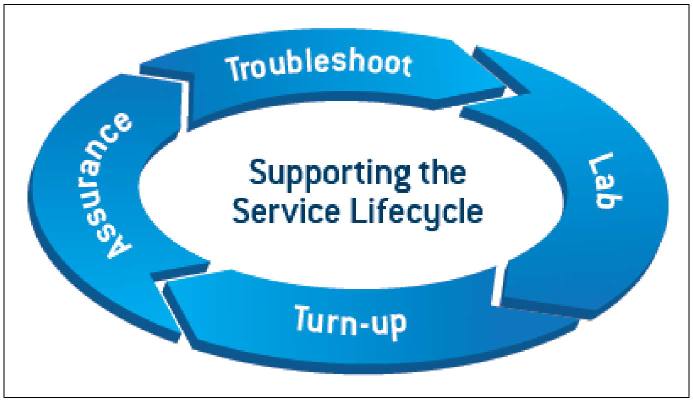


Figure-7: Service Testing Lifecycle

During development of the service, testing is conducted in a development laboratory environment in order to measure the capacity of the system under load. Cloud services, especially public cloud services, are meant to be scaled to very large numbers of users. As such, load testing the service before deployment is critical.
When a new service or a new customer is first deployed, carriers and cloud providers should test to ensure that the service will actually function for the customer. The test should validate that the capacity measures up to the actual capacity purchased by the customer, and that the promised SLA is achievable with the resources provided. Some of the challenges in performing effective turn-up testing result from the elastic capacity, and on-demand customer provisioning of capacity that is a key feature of the cloud. Cloud consumers expect to be able to turn up new VMs, new storage, new servers with new addresses, and new DNS names backed up by new network bandwidth within minutes [5]. To ensure that those resources are available, carriers and the cloud providers need to carry out tests to confirm that they are present within that time frame.

Once a service is up and operating, 24/7 assurance testing of the service should be set in motion. These tests measure availability and performance to ensure that SLAs are met, and proactively detect developing problems.
When problems are found, the operations staff and development teams need to be able to troubleshoot the issue. On-demand testing of the service can help shorten the amount of time required to isolate, diagnose and fix the problem. These troubleshooting tests often dig deeper into the service than routine testing.

7. Results and Discussion:

Fault Localization based on Band (FLB) mechanism is compared against the existing Genetic Programming (GenProg) using the JAVA programming. Time overhead is the processing time required by a device (i.e.,) FLB mechanism prior to the execution of a program elements in instruction phase is measured in terms of percentage (%). CPU utilization refers to a usage of processing resources for fault localization. Actual CPU utilization in FLB varies depending on the amount and type of managed computing tasks. Certain tasks require heavy CPU time, while others require less because of non-CPU resource requirements, measured in terms of Mega Bytes (MB).
Amazon EC2 provides a broad collection of instance types optimized on top form diverse use cases. Instance types encompass unreliable mixtures of memory, CPU, storage, and networking capacity and offer the litheness to decide the suitable mix of resources for the required applications. Every instance type comprises one or more instance ranges, permitting to level the resources to the necessities of the target workload.

$$W_{i,r} = \frac{instructions}{Pr}$$

Table-2: Tabulation of Normalized Throughput

No. of users	Normalized Throughput(Kbps)	
	Existing Genetic Programming	FLB Mechanism
3	2010	2510
6	2160	2610
9	2240	2810
12	2470	2930
15	2520	3000
18	2760	3275
21	3020	3630

Table 2 describes normalized throughput based on the users. At the same time, if the user count increases, throughput is improved. The normalized throughput of FLB mechanism and genetic programming is illustrated through the graph given below.

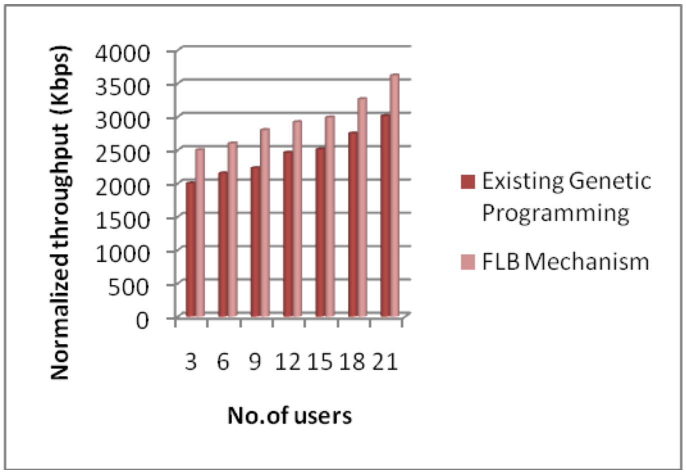


Figure-8: Measure of Normalized Throughput

Figure 8 illustrates the normalized throughput, where FLB mechanism is 15 – 22 % improved when compared with the genetic programming [1]. The FLB mechanism uses replica to convert these set of feature vectors into a discriminative model that predict the effectiveness with normalized throughput using FLB mechanism. Replica learning from instruction phase used the Amazon EC2 dataset for the evaluation of throughput.

Table-3: Tabulation of Precision Matchmaking

Information Size (Bytes)	**Precision Matchmaking (%)**	
	Existing Genetic Programming	**FLB Mechanism**
33	79	91
65	80	92
94	81	93
121	83	94
156	83	97
184	84	96
215	87	98
249	88	99

Table 3 describes precision matchmaking effectively in FLB mechanism and genetic programming based on the information size.

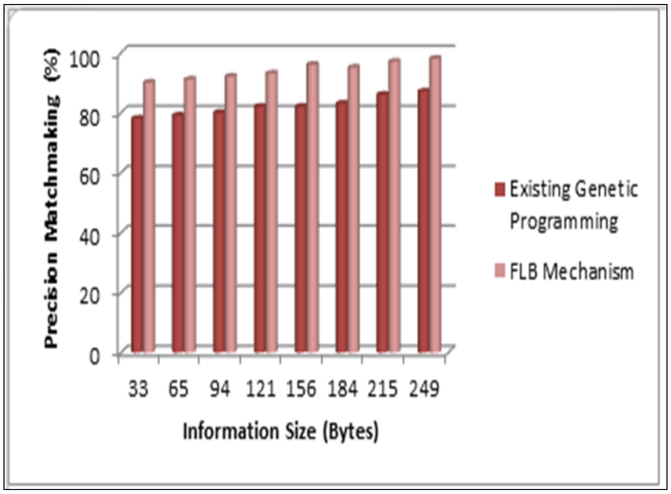
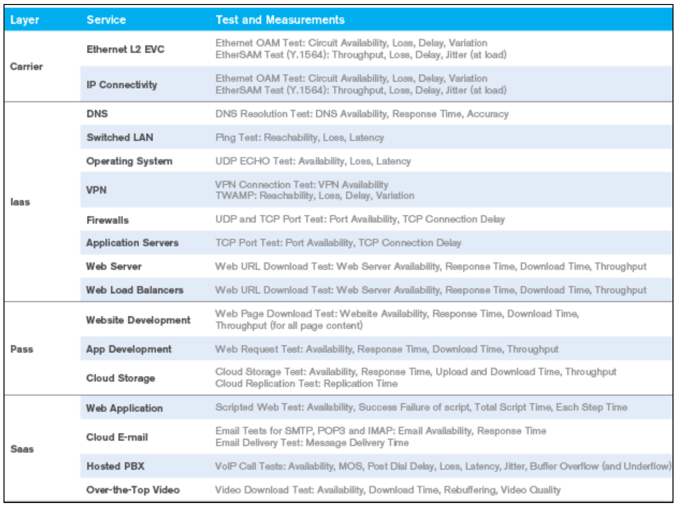


Figure-9: Measure of Precision Matchmaking

Figure 9 describes the precision matchmaking on FLB mechanism and genetic programming. As the information size varies, the precision matchmaking is 10 – 15 % improved in FLB due to the similarity between the instances is identified $\frac{\sum_{i=1}^{40}(p_i * q_i)}{\sqrt{\sum_i^{40} p_i^2 * \sum_i^{40} q_i^2}}$ using the 40 features. Each fault localization instance viewed as a 40 dimensional cloud vector matches the relevance effectively in FLB when compared with the genetic programming.

The table lists some of the tests that could be used by carriers, IaaS, PaaS and SaaS providers, and the measurements that they take.



Layer	Service	Test and Measurements
Carrier	Ethernet L2 EVC	Ethernet OAM Test: Circuit Availability, Loss, Delay, Variation EtherSAM Test (Y.1564): Throughput, Loss, Delay, Jitter (at load)
	IP Connectivity	Ethernet OAM Test: Circuit Availability, Loss, Delay, Variation EtherSAM Test (Y.1564): Throughput, Loss, Delay, Jitter (at load)
Iaas	DNS	DNS Resolution Test: DNS Availability, Response Time, Accuracy
	Switched LAN	Ping Test: Reachability, Loss, Latency
	Operating System	UDP ECHO Test: Availability, Loss, Latency
	VPN	VPN Connection Test: VPN Availability TWAMP: Reachability, Loss, Delay, Variation
	Firewalls	UDP and TCP Port Test: Port Availability, TCP Connection Delay
	Application Servers	TCP Port Test: Port Availability, TCP Connection Delay
	Web Server	Web URL Download Test: Web Server Availability, Response Time, Download Time, Throughput
	Web Load Balancers	Web URL Download Test: Web Server Availability, Response Time, Download Time, Throughput
Pass	Website Development	Web Page Download Test: Website Availability, Response Time, Download Time, Throughput (for all page content)
	App Development	Web Request Test: Availability, Response Time, Download Time, Throughput
	Cloud Storage	Cloud Storage Test: Availability, Response Time, Upload and Download Time, Throughput Cloud Replication Test: Replication Time
Saas	Web Application	Scripted Web Test: Availability, Success Failure of script, Total Script Time, Each Step Time
	Cloud E-mail	Email Tests for SMTP, POP3 and IMAP: Email Availability, Response Time Email Delivery Test: Message Delivery Time
	Hosted PBX	VoIP Call Tests: Availability, MOS, Post Dial Delay, Loss, Latency, Jitter, Buffer Overflow (and Underflow)
	Over-the-Top Video	Video Download Test: Availability, Download Time, Rebuffering, Video Quality

8. CONCLUSIONS

Cloud providers and cloud carriers have complex business relationships that are constantly evolving. Cloud carriers can use their own infrastructure to operate a cloud, partner with existing cloud providers, or broker and audit multiple cloud providers. Cloud providers can buy network interconnections from cloud carriers, sell their cloud services through carriers as resellers, or partner with a carrier to provide an integrated offering.

The best way to confirm whether the service is really working is to test it. Carriers and providers want to perform tests as close as possible to the customer, which means conducting testing from within the carrier's network by testing the cloud and cloud datacenters where the application is running. Tests performed at the appropriate points in the service lifecycle are capable of accomplishing a great deal. During development, the service under load should be tested in order to determine whether it scales. During customer provisioning and turn-up, tests should be conducted to verify whether the service is up and running, and whether it has the anticipated capacity and performance. In normal, daily 24/7 operations, the services must be periodically tested in order to measure SLAs, manage end-to-end performance and proactively detect problems before users do. During troubleshooting, testing must be performed to isolate and diagnose any problems, and once again when the service is fixed to verify its success.

Cloud consumers utilize the cloud because it saves money. They like that they can buy only what they need, and elastically expand and contract their purchase on demand. They can also deliver better quality of experience to their users by choosing a cloud provider that has data centers in close proximity to users. The cloud can provide higher availability due to the simple fact that cloud providers are experts in operating large data centers. But, there is no denying the fact that cloud consumers fear the loss of control inherent to using the cloud. To help consumers overcome this fear, the cloud carrier and cloud provider must provide information about the cloud, especially information on users' quality of experience. Measuring QoE is dependent upon the cloud carrier working with the cloud provider to test the cloud. And together, they must make what they find available to their users. When it comes to user experience, the cloud must be transparent, not cloudy.

REFERENCES

[1] Imad M. Abbadi., and Anbang Ruan., "Towards Trustworthy Resource Scheduling in Clouds," IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8, NO. 6, JUNE 2013

[2] Rongxing Lu., Xiaodong Lin., Xiaohui Liang., and Xuemin (Sherman) Shen., "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," ACM journal., 2010

[3] Balakrishnan.S., Saranya.G., Shobana.S., Karthikeyan.S., "Introducing Effective Third Party Auditing (TPA) for Data Storage Security in Cloud," IJCST Vol. 2, Issue 2, 2011

[4] Smitha Sundareswaran., Anna C. Squicciarini., and Dan Lin., "Ensuring Distributed Accountability for Data Sharing in the Cloud," IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 4, JULY/AUGUST 2012

[5] Yan Zhu, Hongxin Hu, Gail-Joon Ahn, Senior Member, IEEE, Mengyang Yu "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2012

[6] Ming Li., Shucheng Yu., Yao Zheng., Kui Ren, and Wenjing Lou., "Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2012

[7] Cong Wang, Kui Ren, Jia Wang., and Qian Wang., "Harnessing the Cloud for Securely Outsourcing Large-scale Systems of Linear Equations," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2012

[8] Vivek Nallur., Rami Bahsoon., "A Decentralized Self-Adaptation Mechanism For Service-Based Applications in The Cloud," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 2012

[9] Zhiguo Wan., June Liu., and Robert H. Deng., "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing," IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 7, NO. 2, APRIL 2012 743

[10] Xuyun Zhang., Chang Liu., Surya Nepal., Suraj Pandey., Jinjun Chen., "A Privacy Leakage Upper-bound Constraint based Approach for Cost-effective Privacy Preserving of Intermediate Datasets in Cloud," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, TPDSSI-2012

[11] BIOGRAPHIES

Dr.V.Nethaji received B.Sc, M.C.A and M.Phil degree in Computer Science from Periyar University in 2003, 2006 and 2008 respectively andDoctor of Philosophy (PhD) in Computer Science from Karpagam University, Coimbatore, Tamilnadu. He has over 8 years experience in Software Testing field.

Dr.M.**Durairaj** completed PhD at Karpagam University, India. He holds a BSc in Computer Technology and an MCA. He has over 8 years of experience in Network Management system and implementation of large and complex network management in cloud environment.