STUDY OF PARAMETERS FOR EVALUATION OF SOFTWARE AS A SERVICE

Amandeep Kaur
Assistant Professor
Khalsa College of Engineering and Technology, Amritsar

Lakhbirpal Singh
Center Head, SGRD PTU Center
Amritsar

Dr.Hardeep Singh
Professor, GNDU
Amritsar

ABSTRACT

Cloud computing is widely believed to be a revolution in computing that could soon become an industry standard, altogether replacing the traditional office setup. Due to the recency of these services, question marks exist over the performance of these systems, and consequently, over the corresponding billing schemes that the service provider issues. This study aims at addressing some of these doubts by introducing a measurement test for the network performance of these services, and comparing them with the performance offered by traditional web-hosting services. In this paper the parameters to be taken into consideration are studied and evaluated when actually one is going for purchasing software as a service, so that true value and benefits can be understood.

General Terms

Cloud computing, Software as a Service (SaaS).

Keywords

Scalability, Availability.

1. INTRODUCTION

CLOUD is a metaphor used for the internet and is an abstraction for the complex infrastructure that it conceals. Cloud computing is a general concept that incorporates software as a service (SaaS),Web 2.0 and other recent, well-known technology trends, in which the common theme is reliance on the Internet for satisfying the computing needs of the users. A SaaS (Software as a Service) application runs entirely in the cloud, using the servers of the service provider over the internet. The client is a simple browser or some other simple client. For example, Google App Engine provides common business applications online that can be accessed from a browser, while the software and the data are stored on Google's server.

The potential of cloud computing is undoubted, if suitably deployed. Speculations are that it may lead to a new revolution in computing that could become the industry standard .Analogous to how very few people today prefer to build a house on their own, but rather prefer to rent one, in the next generation of computing, people may prefer to opt for a scalable and reliable provider for their computing needs, that will actually minimize risks while launching a new application, rather than build an entire new enterprise for the purpose of launching products[4]. Our motivation for this measurement study stems from the hype created around the Cloud Computing concept. Although there is much talk about the advantages of using the cloud but here study is about the performance of the Cloud Computing service by considering various parameters under consideration under varied load conditions to answer if the hype created is actually justifiable.

2. DIFFERENT SERVICES

There are three different services provided by cloud which are explained as under and shown in fig1.

2.1 Application Services

The application services layer hosts applications that fit the SaaS model. It is sometimes referred to as "software on demand," is software that is deployed over the internet and/or is deployed to run behind a firewall on a local area network or personal computer. With this, a provider licenses an application to customers either as a service on demand, through a subscription, in a "pay-as-you-go" model, or (increasingly) at no charge. This approach to application delivery is part of the utility computing model where all of the technology is in the "cloud" accessed over the Internet as a service. Some SaaS applications are free to the user, with revenue being derived from alternate sources such as advertising, or upgrade fees for enhanced functionality (often referred to as "freemium"). Examples of free SaaS applications include large players such as Gmail and Google Docs, as well as smaller providers like Wave Accounting and Freshbooks[2].

2.2 Platform Services

Platform as a Service (PaaS) is an application development and deployment platform delivered as a service to developers over the Web. It facilitates development and deployment of applications without the cost and complexity of buying and managing the underlying infrastructure, providing all of the facilities required to support the complete life cycle of building and delivering web applications and services entirely available from the Internet. This platform consists of infrastructure software, and typically includes a database, middleware and development tools. Some PaaS offerings have a specific programming language or API. For example, Google AppEngine is a PaaS offering where developers write in Python or Java. Engine Yard is Ruby on Rails[5].

2.3 Infrastructure Services

Infrastructure as a Service (IaaS) is the delivery of hardware (server, storage and network), and associated software (operating systems virtualization technology, file system), as a service. It is an evolution of traditional hosting that does not require any long term commitment and allows users to provision resources on demand. A set of physical assets such as servers, network devices, and storage disks offered as provisioned services to consumers. The services here support application infrastructure -- regardless of whether that infrastructure is being provided via a cloud -- and many more consumers. Examples of infrastructure services include IBM Blue House, VMWare, Amazon EC2, Microsoft Azure Platform, Sun ParaScale Cloud Storage and more[6].

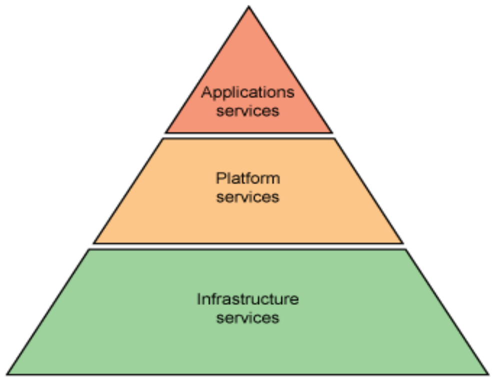


Fig 1: Cloud Anatomy

3. SOFTWARE AS A SERVICE

It typically pronounced [sæs]), sometimes referred to as “software on demand," is software that is deployed over the internet and/or is deployed to run behind a firewall on a local area network or personal computer. With SaaS, a provider licenses an application to customers either as a service on demand, through a subscription, in a "pay-as-you-go" model, or (increasingly) at no charge. This approach to application delivery is part of the utility computing model where all of the technology is in the "cloud" accessed over the Internet as a service. SaaS providers generally price applications on a per-user basis and/or per business basis, sometimes with a relatively small minimum number of users and often with additional fees for extra bandwidth and storage. SaaS revenue streams to the vendor are therefore lower initially than traditional software license fees, but are also recurring, and therefore viewed as more predictable, much like maintenance fees for licensed software.
SaaS application development may use various types of software components and frameworks. These tools can reduce time-to-market and the cost of converting a traditional on-premise software product or building and deploying a new SaaS solution. Examples include components for subscription management, grid computing software, web application frameworks, and complete SaaS platform products.

4. PARAMETERS TAKEN UNDER CONSIDERATION

4.1 SAAS Scalability

Scalability is an important topic to be considered when planning for a SaaS implementation. Since customers rely on the SaaS vendor to provide the solution reliably, it is critical that the application be able to handle the load of the current customer base and also be able to keep pace with the growth of that base[9]. “Scalability is a desirable property of a system, a network, or a process, which indicates its ability to either handle amounts of work in a graceful manner, or to be readily enlarged. For example, it can refer to the capability of a system to increase total throughput under an increased load when resources (typically hardware) are added[11].

4.1.1 Dimensions

4.1.1.1 Load Scalability

Load scalability deals with a system’s ability to effectively handle increasing levels of demand and throughput in a non-disruptive way. By non-disruptive it means that additional resources can be added to the mix in a predefined way (i.e., servers, processors, RAM, disk space, bandwidth, etc.) so that as demand increases, the environment can be expanded to handle the increase.

4.1.1.2 Administrative Scalability

Administrative scalability has to do with the SaaS vendor’s ability to manage multiple customers in a single environment. The ability to deliver consistent levels of service to each customer as the customer base grows is a function of how effective the administrative systems are that the staff uses to manage the customers.

4.1.2 Scalability Methods

4.1.2.1 Vertical

Scaling vertically, sometimes referred to as scaling up, means to add more resources to a node (a device connected to a network, such as a server or router) in a system. In terms of an application infrastructure, this might involve adding additional processors, RAM or disk space to a server. Planning for vertical scaling of hardware usually includes adding redundancy of all components (i.e., power supplies, fans, disk drives, etc.) to insure that machines remain running if a given component fails.

4.1.2.2 Horizontal

Scaling horizontally sometimes referred to as scaling out, means to add more nodes to a system. This may look like adding another server to an application server cluster. A cluster is a group of systems that are coupled together, generally via a fast network, so they are aware of each other. Work is distributed to individual systems in the cluster and if one becomes unavailable, the others are able to handle the processing seamlessly[7].

4.1.3 Tool used to measure Scalabilty-Apache JMeter

It is open source software, a 100% pure Java desktop application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions. Apache JMeter may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. It is used to make a graphical analysis of performance or to test the server/script/object behavior under heavy concurrent load. Apache JMeter is a tool that can be used to test applications utilizing HTTP or FTP server. It is Java based and is highly extensible through a provided API. A typical JMeter test involves creating a loop and a thread group. The loop simulates sequential requests to the server with a preset delay. A thread group is designed to simulate a concurrent load. JMeter provides a user interface. It also exposes an API that allows running JMeter-based tests from a Java application. To create a load test in JMeter build a test plan, which is essentially a sequence of operations JMeter will execute[3]. The simplest test plan normally includes the following elements:

a.Thread group

These elements are used to specify number of running threads and a ramp-up period. Each thread simulates a user and the ramp-up period specifies the time to create all the threads. For example with 5 threads and 10 seconds of ramp-up time, it will take 2 seconds between each thread creation. The loop count defines the running time for a thread. The scheduler also allows setting the start and end of the run time.

b .Samplers

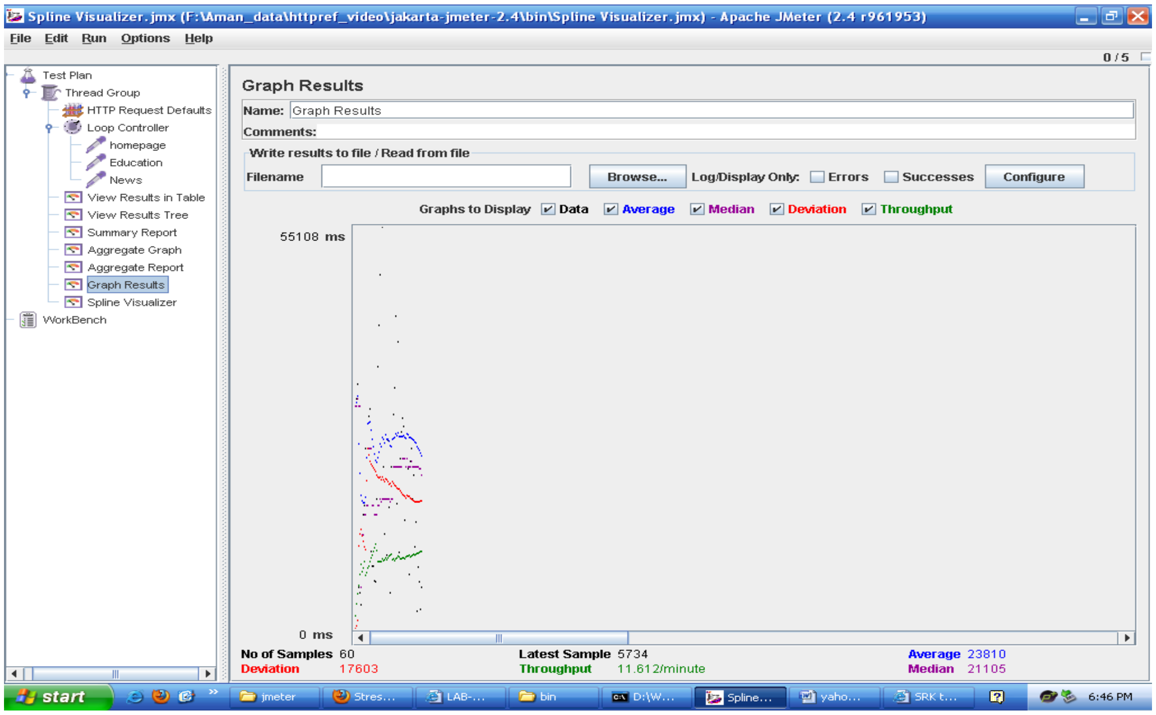
These elements are configurable requests to the server HTTP, FTP, or LDAP requests.It focuses on the Web service requests only[14].

c. Listeners

These elements are used to post process request data. For example, it can save data to a file or illustrate the results with a chart. At the moment the JMeter chart does not provide many configuration options; however it is extensible and it is always possible to add an extra visualization.

4.1.4 JMeter Related Metric

The test is based on www.yahoo.com with different users. Firstly users are taken as 60 and through Jmeter various results are taken as shown in fig2.



Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Log/Display Only: Errors Successes
Graphs to Display ✓ Data ✓ Average ✓ Median ✓ Deviation ✓ Throughput

No of Samples 60
Deviation 17603
Latest Sample 5734
Throughput 11.612/minute
Average 23810
Median 21105

Fig 2: Graph results for 60 users

Then the results are taken for 180 users .The comparison table is given in table 1, showing results for average, median, min ,max and error percentage.

Table 1: Result of the two comparisons

S.No	Sample	Average	Median	Min	Max	Error
1	60	23810	21105	2426	79430	0%
2	180	42322	30492	1392	150599	0.56%

- **Samples**:

A sample mean one sampler call. One request to web page in our case. So the value of 60 means total 60 web page requests to the page http://www.yahoo.com were made by JMeter

- **Average**:

This value is the average time taken to receive the web pages. In our case there were 60 values of receiving time in 1st case which were added and divided by 60 and this value is arrived by JMeter. This value is a measure of performance of this web page. This means on an average 23810 milliseconds time is required to receive this web page for our network conditions.

- **Min and Max**:

These are the minimum and maximum value of time required for receiving the web page.

- **Std. Dev**:

This shows how many exceptional cases were found which were deviating from the average value of the receiving time. The lesser this value more consistent the time pattern is assumed.

- **Error %**:

This value indicated the percentage of error. In our case 60 calls were made in the first case and all are received successfully this means 0 error.

- **Throughput:**

The number of requests processed by the server per minute.
Throughput (server can handle) = Number of request / minute = (average time (in sec) * number of threads) / 60 sec

4.2 Availability

In Cloud Computing there is a need to set the expectations for various platform components to deliver high availability service. In general terms, business-critical applications need to be available between 99.9% (8.76 hours/year downtime) to 99.95% (4.38 hours / year downtime). In a SaaS environment, software architecture, software quality control, operational deployment procedures, network and server architecture and proactive service monitoring are all links of the chain designed to provide a highly available environment. SaaS platform instrumentation for user experience, availability and capacity are important to ensure that any potentially service-affecting trends are proactively monitored. High availability services are a 24x7x365 commitment.

4.2.1 Tool used for Availability-CloudSleuth

Day by day, the hype of cloud computing is constantly catching the fire and there is always a need to benchmark every single technology and test whether it is going to give desired results or not. So, Compuware has released Cloud Sleuth, a free cloud and web service performance monitor that develops an independent view of how well services on the web are running and gives that feedback to a related application-monitoring system in the data center..Cloud Sleuth can tell how well an application is performing to an end user looking at a Microsoft Explorer window versus Mozilla's Firefox. It can tell if application running on a server in Amazon's EC2 data center in Virginia is performing up to speed, or for that matter, application components in the Go Grid, Microsoft Azure, or Google App Engine clouds. So Cloud Sleuth is the new web-based benchmark tool for the performance of Cloud's IaaS (Infrastructure as a Service) and SaaS (Software as a Service) providers. Their downtime, uptime, response time and constant availability parameters, all of them are tested by this online tool to help decide which service provider fits according to organizational benefits. Cloud Sleuth uses the Gomez Performance Network (GPN) to measure the performance of an identical sample application running on several popular cloud service providers[1].

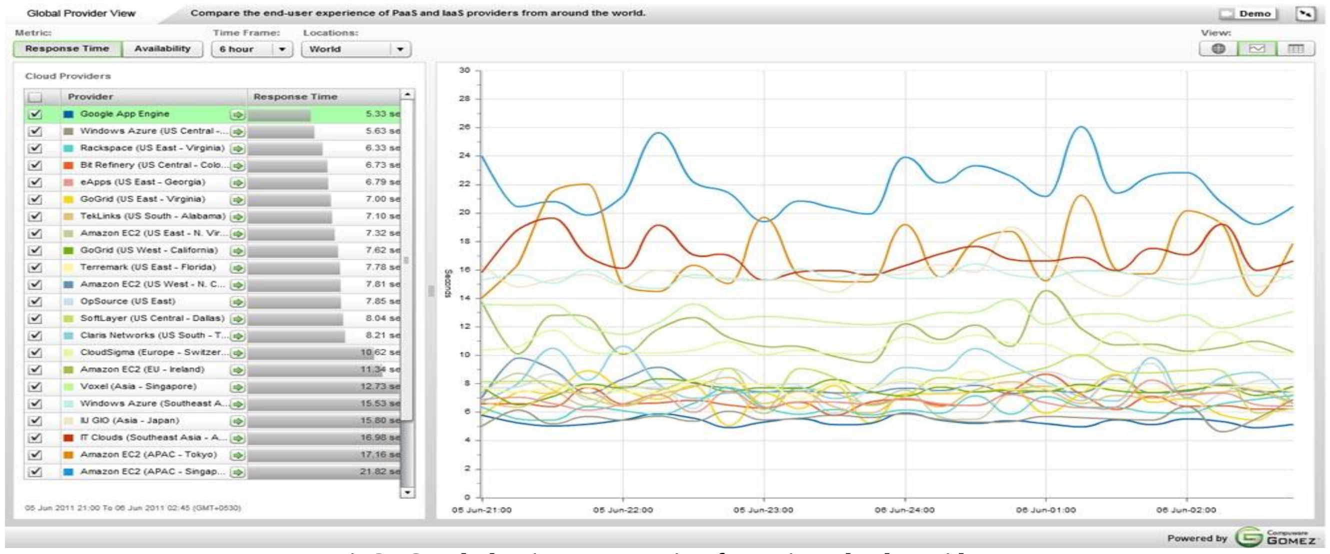
4.2.2 Cloud Availability Metrics

4.2.2.1 Response Time

Response time is the total time elapsed while downloading both Web pages in the multi-step test transaction. Each page's end-to-end response time includes the page's root object as well as all referenced image objects, JavaScript, Cascading Style Sheets, and any other related content. For aggregate reporting results, the response time is the average response time of all successfully completed tests over the period. The response time for various cloud providers is shown in fig 3.

4.2.2.2 Availability

Availability measures the percentage of test transactions that completed successfully out of the set of transactions attempted. An unsuccessful test transaction is a transaction that returns a status code other than "200," one that provides some other critical error or that fails to download a page in the maximum allowable timeframe.



	Provider	Response Time
✓	Google App Engine	5.33 se
✓	Windows Azure (US Central -...	5.63 se
✓	Rackspace (US East - Virginia)	6.33 se
✓	Bit Refinery (US Central - Colo...	6.73 se
✓	eApps (US East - Georgia)	6.79 se
✓	GoGrid (US East - Virginia)	7.00 se
✓	TekLinks (US South - Alabama)	7.10 se
✓	Amazon EC2 (US East - N. Vir...	7.32 se
✓	GoGrid (US West - California)	7.62 se
✓	Terremark (US East - Florida)	7.78 se
✓	Amazon EC2 (US West - N. C...	7.81 se
✓	OpSource (US East)	7.85 se
✓	SoftLayer (US Central - Dallas)	8.04 se
✓	Claris Networks (US South - T...	8.21 se
✓	CloudSigma (Europe - Switzer...	10.62 se
✓	Amazon EC2 (EU - Ireland)	11.34 se
✓	Voxel (Asia - Singapore)	12.73 se
✓	Windows Azure (Southeast A...	15.53 se
✓	IIJ GIO (Asia - Japan)	15.80 se
✓	IT Clouds (Southeast Asia - A...	16.98 se
✓	Amazon EC2 (APAC - Tokyo)	17.16 se
✓	Amazon EC2 (APAC - Singap...	21.82 se

05 Jun 2011 21:00 To 06 Jun 2011 02:45 (GMT+0530)

Fig 3: Graph showing response time for various cloud providers

4.2.3 Features Provided By Cloud Sleuth

4.2.3.1 Location Feature

Its 'Location Feature' is extremely wonderful; in any part of the world cloud's availability and response according where the business resides can be found[1].

4.2.3.2 Detailed Graphical Analysis

Another 'rocking feature' of this tool, the detailed and graphical representation of the response times of different service providers and comparison between them[1].

4.2.3.3 Real-Time Cloud Computing Testing

It provides a real time cloud computing test environment for all the developers in this field, they can run and test there cloud apps here as well; they just need to register for free.

4.2.3.4 A Free Service

It is a free service for anyone without any operating or travelling costs to determine the best cloud provider for the business.

4.2.3.5 Range Value

As of July 2010, a Range value as been introduced. This Range value is calculated based on the aggregate values of the different backbones nodes that have been chosen. The Range value is provided to provide a nice look at the Range of values on a given backbone node for the given geographical location.

4.2.3.6 Reporting Windows

Cloud Sleuth reports metrics as a moving average over a user selectable reporting window. Currently, four reporting windows are available: 6 hours, 24 hours, 7 days, and 30 days.

5. CONCLUSION

Software as a Service (SaaS) is the software that is available as a service from various cloud providers. And users are charged on basis of usage and are freed from other parameters to be taken care i.e. maintainenace, up gradation, etc.. All these headaches are for the SaaS provider rather than the SaaS user. By having a study on its characteristics and various parameters, it is concluded that it's better to opt for SaaS rather than choosing for traditional software. The SaaS is easily scalable, as the load increases the throughput also increases. SaaS availability can be easily seen by real cloud virtualization tool.

6. REFERENCES

[1] https://www.cloudsleuth.net/web/guest/applications

[2] http://docs.google.com/demo

[3] http://jakarta.apache.org/jmeter/usermanual/index.html

[4]B,Furht,A,Escalante(eds.),"Handbook of Cloud Computing", Springer Science Business Media ,2010,pp 3-18,ISBN 978-1-4419-6523-3

[5] Yadav.S and Hua,Z, " CLOUD: A Computing Infrastructure on Demand," IEEE Computer Society,2010, pp. 50- 55 vol. 40, no. 12

[6] Yong,L., Tao,Z.," A Trusted Computing Environment Model In Cloud Architecture", IEEE Computer Society, Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao, 11-14 July 2010.

[7] Idziorek,J., "Discrete Event Simulation Model For Analysis Of Horizontal Scaling In The Cloud Computing Model", Proceedings of the 2010 Winter Simulation Conference, 2-5 June-2010

[8] Vallini,M.,"Software as a Service (SaaS) ethical issues". White Paper, 1st Editon, May 2009

[9] Joyent, "Performance and Scale in Cloud Computing", White Paper, 2nd Edition, June 2010

[10] 'Introduction to SaaS Architecture' White Paper 1st Edition, June 2009.

[11] Valis,M.,"Software Scalability". White Paper, 1st Edition, May 2008

[12] M.N.Yigitbasi, D.H,J.Epema, A.Iosup, "C-Meter: A Framework for Performance Analysis of Computing Clouds" ,University of Innsbruck, Oct 2010

[13] S.Ankit, "Network-based Measurements on Cloud Computing Services", Department of Computer Sciences, The University of Texas at Austin, May 2010

[14] M.Kelly, "Running load test with JMeter", May 2009