Performance comparison of web service in IaaS cloud and standard deployment model

Jithin P[#1], Dr S.K.V Jayakumar[*2]

[#]Department of Computer Science, Pondicherry University, India

[*]Assistant Professor, Department of Computer Science, Pondicherry University, India

***Abstract*— In recent years due to the rapid development of e-business, there is huge demand for application that based on web services. Web services are chosen not only on the basis of functionality but also based on non functional properties (NFP) called quality of service (QoS). Among the NFP's performance is key factor used to distinguish one web service from other having same functionality. Researches related to web service performance are very less compared to other research area in web service like ontology and web service composition. Researches related to web service performance are mainly focusing on comparing available web service implementation platforms. Since performance factors of web service are depending deployment environment, if a web service is deployed in an Infrastructure as a Service (IaaS) cloud which is offering a Distributed, dynamic and a flexible deployment platform, the performance of web service may increase. In this work we are comparing the performance of web service deployed in an IaaS cloud and traditional deployment model.**

***Keywords*— Cloud computing, Web Service, Performance analysis.**

I. INTRODUCTION

A web service is a software application that uses slandered internet protocols such as SOAP, WSDL and UDDI to communicate with other web services [20]. Simple Object Access Protocol (SOAP) an XML based messaging protocol working on top of HTTP protocol plays a key role in interaction between web service, its publication and discovery. The functionality, data types and binding used by web service is described using an XML based language called Web Service Description Language (WSDL). An UDDI Server is the Web Services registry that allows users to register and search for Web Services.

The properties of web service like modular, programming language and platform independence, loosely coupled and self-describing nature attracts developers for developing highly distributed business application over the internet. Since web services are used by many applications in internet in large scale researches related web service particularly related to the performance of web service, has become more important.

Performance of web service is measured in terms of Non Functional Property (NFP). Most commonly used NFP are performance matrices such as response time, throughput and other matrices such as security, reliability and availability. NFP are properties that do not depend on the functionality of web service but it depends on service deployment environment. [need a reference] Researches related web service performance is mainly concentrating on web service implementation platforms and some of them are discussed in section II. Apart from implementation platform deployment environment also affect the performance of web service. This works compare the performance of web service in tradition deployment environment and cloud based deployment environment.

Cloud computing is an evolutionary technology that provides computing power, middle ware, and business logic as a service over internet. It automate the installation and running of application with the help of virtualization technology. According to NIST cloud computing is a model that allows users to access a shared pool of configurable computing resources such as hardware network, storage, applications, and services on demand. These computing resources can be rapidly provisioned and released with moderate effort and without service provider interaction. Cloud provides three types of services, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [21].

Since provisioning and running a server image is fast and easy in IaaS cloud web services hosted in IaaS cloud can scale up and scale down quickly. Distributed, dynamic and virtual nature of cloud gives more flexibility in the web service deployment environment. The advantage of hosting web service is that the web service developer and that service provider need not worry about the infrastructure. All networking and infrastructure related issues will be managed by cloud service provider.

Rest of this paper is organized as follows. Section II will describe the related works. Section III briefs Experimental Computer Science. Section IV is discussing about factors determining Web Service Performance. Section V gives an over view of our experimental setup and Section VI conclude the paper by presenting the evaluation results.

II. RELATED WORKS

The main objective of this paper is to compare the performance of web service in cloud and traditional deployment environments. In recent years several works related to the performance of web service has been carried out with respect to different aspects of web services. Some of these works are listed below.

E-learning is a modern type education which is offering a better educational platform than traditional learning mechanisms. In [1] Abdulmotaleb El Saddik addresses the scalability and performance measurement of web service based multi-tier e-learning system called UbiLearn. UbiLearn is a distributed learning portal implemented in Java. In this work they detailed the architecture and implementation data flow of the application. They used an intelligent test agent named TestMaker provided by PushToTest for the performance testing and found out that when number of users access UbiLearn web service increases through put of the service decreases and number of error response increases. In this work authors are using only one web service to do the performance testing and the deployment environment they used traditional deployment environment.

Most of the web services are implemented using J2EE or Microsoft .NET platform [23]. PHP a well known server side scripting language has officially supported SOAP messaging since version five with the aid of C-based built-in library. In [2] authors studied quantitative and qualitative aspects of PHP based web service while comparing it with web service implemented in Java and C. A performance comparison is done with different combination of web servers and a stock quote web service implemented in PHP, Java and C. Authors reports that PHP as a web service engine performs competitively with Axis2 Java for web services involving small payloads, and greatly outperforms it for larger payloads. Like previous work [1] this work is also focusing on implementation platform.

Researches that combine the performance evaluation based on QoS and web service composition architecture are very less. In [3] authors developed prototype architecture named WSARCH (Web Services Architecture) that incorporates quality of service attributes. WSARCH introduced a new way of evaluating performance of web service and improved existing service discovery solution by incorporating of QoS metrics for ranking matches.

Most of the existing research works on web service performance are based on network level matrices and none consider users' experience. In [4] authors presented an end to end Web Service Evaluation System (WSES). WSE can accurately measure both network performance matrices and the quality of Web Service with special evaluation methods from users' viewpoint. The data presented in this work is from real implementation of the proposed system. WSES offer a complete evaluation solution which allows ISPs to monitor and evaluate the performance of network and Web Service effectively and promptly.

Cloud computing is an evolutionary technology that provides computing power, middleware and business logic on-demand with high availability and pay-as-you-go pricing model. However, to choose cloud computing to deploy mission critical applications and scientific program, all aspects such as cost and performance trade-offs should be analyzed. In [5] and [6], authors conduct an experimental research to discover performance of data-intensive applications in cloud. In [5] authors also discussed method of calculating Cloud and local resources costs.

Since QoS become an important factor in distinguishing web services having same functional behavior many research relating to QoS aware web service selection were came up in recent years. References [7], [8], [9], [10] are about QoS aware web service selection.

III. EXPERIMENTAL COMPUTER SCIENCE

Our methodology of evaluation related to the field of experimental computer Science [19]. Experimental computer science conform the theory prediction and find new phenomena with the help of experiments. An experimental computer science contain following steps an observation, a hypothesis and experimental test.

In computer science observation means the measurement of real system excluding the simulation. Measurement is the concrete basis of observation and main challenge of measurement in computer science is matrices. A measured observation will lead to a model which is the generalization of the result [19]. To verify whether the model is correct generalization we have to make hypothesis about the model and to turn the hypothesis in to knowledge it has pass the experiments. In this work our hypothesis is that when a web service is deployed in cloud its performance factors such a throughput and response time will increase compared to the traditional deployment model. To have solid proof we perform the same test in standard deployment model and cloud deployment model.

IV. WEB SERVICE PERFORMANCE FACTORS

Performance of a web service represents how fast a service request can be served by the web service [11]. Performance of an application or web service can be measured in terms of throughput and response time. These performance factors (throughput and response time) are sub quality factors of Service Level Measurement quality. Service Level Measurement quality is a set of quantity factors which describes the runtime behavior of web service in consumers view. More details about other sub quality factors of Service Level Measurement quality and other Quality factors are listed in [12]. In this work we are focusing on quality factors describing the performance of web service, that is throughput and response time.

A. Throughput of a web service

Throughput of web service represents the number of service request that a web service can complete in a given period of time. Through put can be used as a performance index to evaluate web service provider. Throughput of web service can be measured using following formula

$$Throughput = \frac{Number\ of\ request\ processed}{time\ to\ process}$$

B. Response time of web service

Response time of a web service is the time duration from sending the request to the web service to receive the response from the web service. The value of response time depends on where it is measured. Since we are focusing on measuring the performance change in web service in view consumers we measure the response time at service requester entity. Response time is also vary depending on the latency in client, network and server. Since both of our deployment platform are in same network latency is same, client latency also same because we are using same client to send request to both deployment model. Server latency will vary in our experimental setup.

Response time of q web service can be measured using the following formula

$$Throughput = Response\ completion\ time - User\ request\ time$$

Here user request time is the time when user sends the request and response completion time is the time when last bit of data from the response received.

V. Experimental Setup

Our experimental setup is shown in Fig 1. Each component in experimental set is discussed below.

A. IaaS cloud solution

Prior to this work we done a survey on open source cloud computing solutions [13] and based on this survey we chosen Eucalyptus as cloud solution to implement our project. Eucalyptus is an open source cloud computing platform for implementation of on-premise infrastructure as a service clouds. Using eucalyptus an organization can create private, infrastructure as a service cloud with no retooling of existing IT infrastructure [14]. The Eucalyptus (IaaS) platform has Amazon Web Services (AWS) [15] compatible API, allowing support for hybrid IaaS clouds.

Main components of Eucalypts cloud are: Cloud Controller (CLC), Walrus, Cluster Controller (CC), Storage Controller (SC), Node Controller (NC) and an optional VMware Broker (Broker or VB). Other than the VMware Broker, each component is a stand-alone web service. Discussing the functionality of each component is out of scope of this paper and more details about each component can be obtained from [13] and [14]. In our experimental setup we installed CC, SC, Walrus, and CLC in a single machine and NC in separate machines. The system configurations of machines that we used are given in Table1.

Figure 2 shows a detailed view of Node controller. We are using KVM hypervisor as virtualization software. A load balancer is used to map the load between the virtual servers. The load balancer can be implemented inside the cloud or outside the cloud. We a added Microsoft web farm frame work version 1 module to balance load between IIS servers. The load balancing algorithm we used is weight total traffic with even distribution. Since provisioning and running a server image is fast and easy in IaaS cloud web services hosted in IaaS cloud can scale up and scale down quickly. Table 3 shows the measurement of VM boot times in our experimental environment. The method to measure boot timing of virtual machine is discussed in [16].

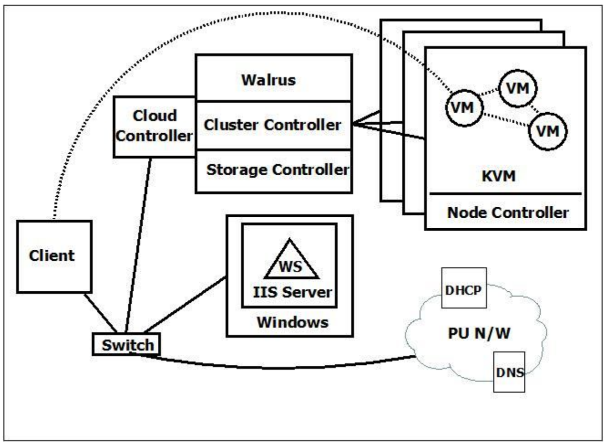

Fig. 1. Experimental Setup.

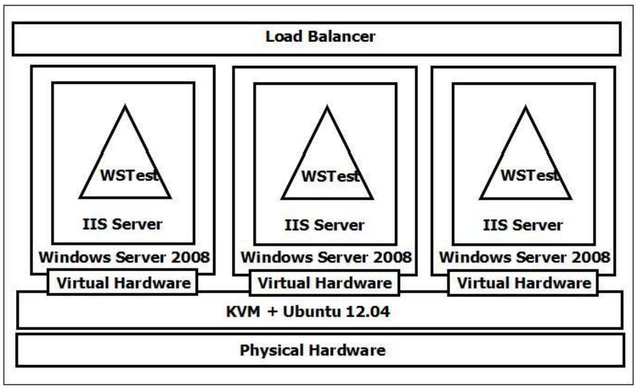

Fig. 2. Node controller setup.

B. Bench Mark

Bench mark is used to assess the relative performance of applications [24]. A benchmark must model typical application scenario so that the result from tests with a benchmark can serve as a general recommendation for the performance of a particular system configuration when the real application is deployed on it.

WSTest is a Web Service benchmark created by Sun Microsystems and augmented by Microsoft [17]. The benchmark tests various web service operations across varying SOAP object sizes. WSTest 1.1 consists of 4 distinct web service methods. They are

EchoVoid – sends and receives an empty message.

EchoStruct –receives an array of any length as the input parameter with each element consisting of a structure. The list is then sent back to the client as the return type. The repeating structures within the list contain one element each of type integer, float and string data types.

EchoList-sends and receives a linked list of any size, each element consists of the same structure used in EchoStruct.

EchoSynthetic- Sends and receives string and complex parameter (struct).

TABLE I
SYSTEM CONFIGURATION

Device name	Hardware	Software
Cloud Controller	Intel Core i3 Processor 4GB RAM 500 GB HDD 100 MBPS Ethernet Card	Ubuntu 12.04 LTS Eucalyptus 3.1-Cloud controller Cluster controller Storage controller Walrus
Node Controller 1	Intel Core i3 Processor 4GB RAM 500 GB HDD 100 MBPS Ethernet Card	Ubuntu 12.04 LTS Eucalyptus 3.1-Node Controller
Node Controller 2	Intel Core i7 Processor 4GB RAM 500 GB HDD 100 MBPS Ethernet Card	Ubuntu 12.04 LTS Eucalyptus 3.1-Node Controller
Standard deployment System	Intel Core i3 Processor 4GB RAM 500 GB HDD 100 MBPS Ethernet Card	Microsoft windows Sever 2008 IIS Server 7
Client System	Intel Core i3 Processor 4GB RAM 500 GB HDD 100 MBPS Ethernet Card	Microsoft Windows 7 Apache JMeter

GetOrder – this function simulates a request for a complete purchase order, taking two integer input parameters with an order object returned from the Web Service over SOAP. The order object is a more complex structure that includes order header information, a customer sub-class with shipping address and billing address structures, as well as any number of line item objects. For this test, the web services return exactly 50 line items for each request as part of the returned order object.

TABLE II
VM BOOT TIME ANALYSIS

Type	OS Image	OS Image Size (GB)	Average VM start-up time (sec)
m1. xlarge	Windows Server 2008	15	336
m1.small	Fedora	4.5	39

TABLE III
VM TYPES USED IN OUR STUDY

Type	CPU	RAM	Disk
m1.small	1	128	5
c1.medium	1	256	5
m1.large	2	512	10
m1.xlaarge	2	1024	20
c1.xlarge	4	2048	20

C. Measurement tool

There are several varieties of tools available to perform load and stress test on web servers [25]. Existing Web Server load testing tools ignore the real characteristics of the practical running and accessing environment of a web service, which leads to inaccurate test results. Load testing of web services should be done by sending valid SOPA message to the service and then analyzing the results. The testing tool must be not only able to generate SOAP messages but also able to create a situation similar to the concurrent access of thousands of users. Apache JMeter [18], Manage Engine QEngine [26] , IBM Rational Performance tester [27] are tools that can create several threads simultaneously .In this work we used Apache JMeter which is an open source Java application to load test WSTest. Using thread group of Apache JMeter used to set the number of users, how often the users should send requests, and the how many requests they should send. With the help of the sampler named Web Service(SOAP) Request [28] we can add web service SOAP request that we need to send to the server for performance testing from the WSDL file. Results can be captured using listener object in JMeter [18].

VI. EXPERIMENTAL RESULT ANALYSIS

Figure 4 shows the result for throughput for all benchmark web service in traditional deployment model, cloud with single virtual machine (VM) and cloud with multiple virtual machines. A web service's throughput will increases as the number of users' increases and reaches a stabilization point and beyond that point throughput decreases very slowly when number of users increases [29]. We used stabilization throughput value to produce graphs shown in Fig 4. The results shows that throughput is almost same in cloud with single VM and traditional model, even though the configuration of VM (m1.xlarge) is very. This shows that traditional deployment model is inefficient and it is wasting lot of resources. When we add multiple VMs the throughput increases more than 50%. Since in cloud provisioning of images takes very less time (Table II) and it is having automatic provisioning capability, web service deployed in cloud can have higher throughput than traditional one.

Figure 5 shows the result of response time for all benchmark web service in traditional deployment model, cloud with single virtual machine (VM) and cloud with multiple virtual machines.

Response time increases when number of users increases. That if there are more users then the server has to handle all of them [29]. Our analysis shows that response time of web service deployed in cloud with multiple VM is very less compared to traditional deployment model. Response time of cloud with single VM and traditional deployment model are almost same. Compared to all other web services response time of echovoid() is very less. Possible reason is time for sending and receiving an empty message will be very less.

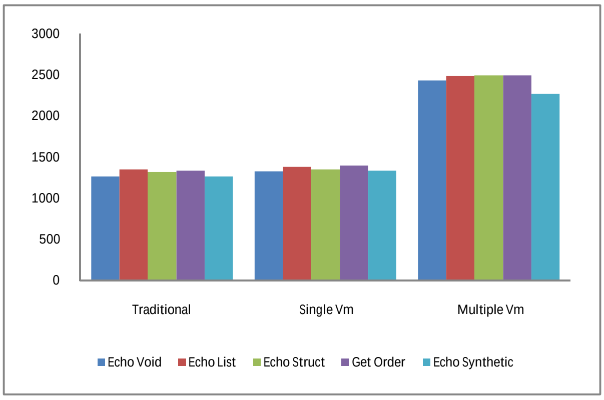

Fig. 3. Throughput measurements.

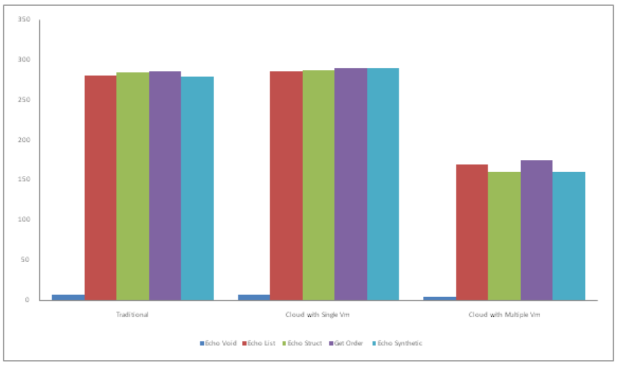

Fig. 4. Response time measurements.

VII. SUMMARY AND FUTURE WORK

In this work we introduced a method for deploying web service in cloud computing environment and we compared the performance of web service in both deployment platforms. We evaluated only .NET web services in windows platform. In future works we will evaluate performance and other NFPs of web services developed in all the available platforms.

ACKNOWLEDGMENT

This work has been funded by University Grants Commission of India under the project titled “Web Service Suitability Assessment in Cloud”..

REFERENCES

[1] Abdulmotaleb El Saddik, “Performance Measurements of Web Services-Based Applications”, IEEE transactions on instrumentation and measurement, VOL. 55, pp. 1599 - 1605, Oct.2006.

[2] Toyotaro Suzumura, Scott Trent, Michiaki Tatsubori, Akihiko Tozawa and Tamiya Onodera, “Performance Comparison of Web Service Engines in PHP, Java, and C”, IEEE International Conference on Web Services, pp. 385 – 392, Sept. 2008.

[3] Julio Cezar Estrella, Rubens Kenji T. Toyohara, Bruno Tardiole Kuehne, Thiago Caproni Tavares Regina H. C. Santana, Marcos J. Santana, Sarita Mazzini Bruschi, “A Performance Evaluation for a QoS-Aware Service Oriented Architecture”, IEEE 6th World Congress on Services, pp. 260 – 267, July 2010.

[4] Fei Sha, Ke Yu, Lin Zhang, Xiaofei Wu, “A Performance Evaluation Method And It's Implementation For Web Service”, 3rd IEEE International Conference on Broadband Network and Multimedia Technology, pp. 218 – 222, Oct. 2010.

[5] Alireza Angabini, Nasser Yazdani, Thomas Mundt and Fatemeh Hassani, “Suitability of Cloud Computing for Scientific Data Analyzing Applications; an empirical study”, International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp.193 – 199, Oct. 2011.

[6] Keith R. Jackson, and Lavanya Ramakrishnan, Krishna Muriki, Shane Canon, Shreyas Cholia, John Shalf Harvey J. Wasserman, and Nicholas J. Wright, “Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud”, 2nd IEEE International Conference on Cloud Computing Technology and Science, pp. 159 – 168, Dec. 2010.

[7] Wenli Dong, Li Jiao, “QoS-aware Web Service Composition Based on SLA”, Fourth International Conference on Natural Computation, pp. 247 – 251, Oct. 2008.

[8] Yutu Liu, Anne H.H. Ngu, Liangzhao Zeng, “QoS Computation and Policing in Dynamic Web Service Selection”, Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, pp. 66-73, 2004.

[9] Eyhab Al-Masri, Qusay H. Mahmoud, “QoS-based Discovery and Ranking of Web Services”, Proceedings of 16th International Conference on Computer Communications and Networks, pp. 529 – 534, Aug. 2007.

[10] Mahmood Allameh Amiri, Hadi Serajzadeh, “QoS Aware Web Service Composition Based On Genetic Algorithm”, 5th International Symposium on Telecommunications, pp. 502 – 507, Dec. 2010.

[11] W3 documentation on web service. www.w3.org/TR/ws-gloss/

[12] Eunju Kim (NCA), Youngkon Lee , “Quality Model for Web Services”, Sept.2005.

[13] Jithin P, Dr. S.K.V Jayakumar, "A survey on Open Source Private Cloud Computing Solutions", Recent Trends in Image Processing and Computing Techniques, pp. 1-7, April 2013.

[14] Eucalyptus home page. www.eucalyptus.com/

[15] Amazon EC2 home page. www.aws.amazon.com/ec2/.

[16] Ming Mao, Marty Humphrey, "A Performance Study on the VM Startup Time in the Cloud", Fifth International Conference on Cloud Computing, pp. 423 – 430, Jun. 2012.

[17] “Web Services Performance: Comparing Java 2TM Enterprise Edition (J2EETM platform) and the Microsoft .NET Framework A Response to Sun Microsystem’s Benchmark Microsoft” Corporation July 2004.

[18] Apache JMeter home page http://jmeter.apache.org/.

[19] D. G. Feitelson, “Experimental computer science,” Commun. ACM, vol. 50, no. 11, pp. 24–26, 2007.

[20] World Wide Web Consortium. www.w3.org.

[21] Frank P.coyle, “XML,Web services, and the Data Revolution”, Pearson Education, Fourth Impression 2007, pages 12-13.

[22] “The NIST Definition of Cloud Computing”, Special Publication 800-145.

[23] David Chappell, Tyler Jewell, "Java Web Services", Publisher: O'Reilly, First Edition March 2002.

[24] Robert J. Boxwell, “Benchmarking for Competitive Advantage”,McGraw-Hill, 1994

[25] Open source testing tools http://www.opensourcetesting.org.

[26] Manage engine home page http://www.manageengine.com

[27] IBM Rational performance tester www-03.ibm.com/software/products/us/en/performance/

[28] Khushali Tirghoda, “Web Services Performance Testing Using Open Source Apache Jmeter”, International Journal of Scientific & Engineering Research, Volume 3, May.2012.

[29] J.R.Ariass, F.J.Suarez, D.FGarcia, "Evaluation of Video Server Capacity with Regard to Quality of the Service in Interactive News-On-Demand Systems", Joint International Workshops on Interactive Distributed Multimediaa Systems and Protocols for Multimediaa Systems, pp.15-27, Nov 2002.