Sensors & Transducers, Vol. 209, Issue 2, February 2017, pp. 56-64

www.sensorsportal.com

Sensors & Transducers
Published by IFSA Publishing, S. L., 2017
http://www.sensorsportal.com

Investigations on Evaluation of Some QoS Aspects of Service Oriented Computing System Based on Web Services

Subhash MEDHI, Abhijit BORA and Tulshi BEZBORUAH
Department of Electronics and Communication Technology, Gauhati University,
Assam, India, PIN-781014
Tel.: +91-361-2671262
E-mail: medhisubhash72@gmail.com, abhijit.bora0099@gmail.com, zbt_gu@yahoo.co.in

Received: 31 January 2017 /Accepted: 24 February 2017 /Published: 28 February 2017

Abstract: Service Oriented Computing is a design paradigm that utilizes autonomous heterogeneous service applications as the fundamental elements to develop new composite functionalities at reduced cost and time. Web service is the standard way to implement the service oriented computing concepts in which business functions and resources are published, described, discovered, orchestrated and invoked using open standards and protocols. The web services emerged as an intelligent middleware web based technology for sharing business processes and resources amongst the disparate enterprises over the internet. Performance evaluation of service is an important criterion to be assessed by end users and service providers before adopting web services to deal with the challenging global markets. In this perspective, we propose to implement a composite ATM services using. Net technology to evaluate trustworthiness of web services in dealing with massive users. The uniqueness of our proposed system is the hierarchically designed parent and child services where the parent service authenticates a user to access resources and redirects the user's query for executing child service for adequate solutions. The industry standard testing software tool, Mercury LoadRunner was deployed to test our proposed e-ATM system and record the performance metrics to analyse the quality aspects of the service. The outcome of the experiment will help in adoption and usage of the web services in diverse business enterprises. We present here the architecture, framework of testing, transaction status and reliability estimation of web services under massive stress of service users.

Keywords: SOC, Web Services, SOA, SOAP, QoS, Reliability, ATM, e-ATM.

1. Introduction

The rapid development of Internet brought more people to familiar with computer and mobiles and people embraced it for better communication, services and resource sharing. With the advent of Web Services (WS), the Internet has been changed from information repository to service repository. WS is a software component designed to support interoperability within machines over a network and public interfaces are described in a process able format termed as Web Service Description Language (WSDL) [1]. The WS technology implements Service Oriented Computing (SOC) design paradigm to develop and integrate heterogeneous services by using open standards and protocols. SOC is a resource and business functionalities in open environments [2]. The SOC system concentrated on composing autonomous services by using mutually agreed standards and protocols. Service Oriented Architecture (SOA) is an architectural design paradigm to organize and utilize distributed applications deployed in disparate environments that can be controlled by different owners [3]. The WS supports communications with other services and clients using technologies such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), WSDL, Universal Description Discovery Integration (UDDI) and internet protocols, such as, hyper text transfer protocol (HTTP) and hyper text transfer protocol secure (HTTPS) [4]. Once the WS is deployed, various heterogeneous organizations that are running on diverse networks can be interconnected for business processes. WS transactions are used for efficient and reliable communication of web applications across the internet where multiple users can access resources simultaneously [5]. It is an important issue to assess the trustworthiness of composite WS built on different networks and languages [6]. The ever growing reliance on the data and services provided by various WS vendors consent these services be superior in performance and reliable, as these services offered over the internet have rapidly permeates our lives due to the convenience and low cost factors [7]. In widespread application of internet technology, WS applications such as online shopping, banking, travel booking, stock trading have been adopted and developed tremendously [8]. WS is becoming more familiar to the people due to the recent development and applications over the internet, as many top companies such as Microsoft, IBM, Oracle and SUN have launched supports for technologies related to WS [9]. An important quality of software is the capability to which it can perform to its intended operation. The reliability is an important factor that represents the quality service delivery of WS system. Thus the reliability of WS system plays an important role in an organization as unreliable service system may lead to economic loss as well as reputation of stakeholders. The software failure is attributed to the quality of software, hardware and human intervention; hence it is essential to analyse the quality of software, hardware and principle of application before implementing in to an existing system. The software reliability is a probabilistic assessment which can be defined as the probability function that the software works failure free within a specified environment and a given amount of time [10]. Many software tools have been developed to estimate the reliability of software that helps to tracks the quality of software product right from the development processes, concept to commissioning. In this study we present a heuristic approach to investigate the quality of service properties in a composite e-ATM service delivery system. The Mercury LoadRunner performance testing tool is deployed to test the proposed model. The statistical analysis of the recorded data for reliability aspects of the WS is presented here.

2. Related Work

In the year 2006, Abdelkarim, *et al.* had addressed the requirements of reliability and fault tolerance of service delivery system and proposed some recovery policies to handle and recover from faults during the composition of WS [11].

In the year 2014, Shangguang Wang *et al.* had proposed a framework for solving multi user WS selection problem by predicting missing QoS values based on the historical QoS values of multi users. The authors stated that this prediction algorithm can predict the multi-QoS values more precisely at same time for different users [12].

In the year 2007, Pat, *et al.* had identified some parameters that can impact the WS dependability. They had elaborated the methods of dependability enhancement by redundancy in space and time [13].

In the year 2007, Michael P., *et al.* had proposed a business aware WSs transaction model and support mechanism. The proposed model allows business functions such as payment, delivery conditions, and dispute resolution policies and blends these functions with QoS criteria [14].

In the year 2010, Huiyuan Zheng, *et al.* had proposed an approach to calculate the QoS for composite services with complex structures. Using the proposed approach, users can explicitly specify their requirement on the mean path, the maximum and minimum value of Quality of Service (QoS) [15].

In the year 2014, Bora, *et al.* had presented an empirical study on hierarchical SOAP based WS implementing WS Security policy. The authors had observed that the WS response time with security incorporation is more than the time without security policy [16].

In the year 2014, Medhi, *et al.* had performed empirical and statistical analysis of hierarchical WS performance by implementing a financial model [17].

In the year 2014, Maurizio Gabbrielli, *et al.* had proposed a methodology which maps Coloured Petri nets-modelled SOAs in to Jolie SOA by exploiting some workflow patterns [18].

In the year 2015, Bezboruah, *et al.* had performed an evaluation of performance of hierarchical WSs using a cluster and non cluster web server [19].

In the year 2015, Bora A., *et al.* had carried out a study on the quality evaluation of interoperability for multi service multifunctional SOC system [20].

In the year 2016, Medhi, *et al.* had implemented a service model e-ATM system using WCF technology incorporating a message level security. The authors had observed that the security policy incorporation in service system influences the performances of WS system [21].

In the year 2017, Medhi, *et al.* had performed investigations on reliability of WCF system implemented in a prototype financial service model. The authors experimentally proved in the proposed model that the reliability of WS remains strong up to 600 virtual users (VU) and degrades with the higher levels of service consumers [22].

This work is an extended version of our study of a series of test conducted randomly by using different levels of VU such as 30, 50, 100, 300, 500, 800, 1000, 1400, 1500, 1900, 2000, 2200 and 2500 [23]. For statistical analysis and evaluation of the service quality of WSs, we have recorded the fault count (FC) for 30 times at 1500 VU. We have improved the average connection refusal to 1.07 % and also extended the test up to 2500 VU.

In this study we have performed the investigations on user perspective evaluation of some QoS characteristics of SOC system based on WS. The novelty of this work is that we have done a series of test randomly at different levels of VU by using an industry standard software testing tool, recorded some quality attributes, failure count and prepared a framework for reliability analysis for predicting the nature of WSs under massive stress of concurrent users.

3. Proposed Work and Methodology

The objective of the proposed work is to implement a prototype electronic automated teller machine (e-ATM) service delivery system using .NET technology to study some QoS attributes of WSs. The prototype service model has three components as presented in Fig. 1:

1) A Service Consumer;

2) Parent ATM service where the consumers identities are authenticated and,

3) Child ATM service to execute Business Processes. The authenticated users are authorised to access the resources that is users queries are served by the Child ATM service application.

To evaluate the availability and scalability of the service model, the software testing tool, Mercury LoadRunner was deployed and recorded some quality of service characteristics of the system. The prototype e-service system was designed to provide all the facilities of commercial bank ATM facilitates to a consumer. A service consumer was developed to invoke the services using the same design and development environment of the service applications. A database of size 15,000 is prepared for testing the proposed service. The service application has been tested at different levels of VU to evaluate the quality of the WSs. The statistical analysis is done on the recorded QoS attributes to estimate the availability, scalability and reliability of the services.

4. Software and Hardware Specification

The software used to develop the service applications at server system are:

a) IIS 7.5;

b) MS SQL Server;

c) Microsoft Visual Studio, 2012;

d) Internet Explorer;

e) Windows Server 2008 and the software supporting tools such as Microsoft SDK version 7.1.

The hardware components in the server includes Intel(R) Xenon(R) CPU E5620 processor with 2.4 GHz speed, 8 GB RAM and 600 GB hard drive. The software testing tool Mercury LoadRunner, Windows XP(OS), Internet Explorer, EasyFit5.6 were used by the load generator machine to generate the load on the services. We have created the script by capturing the VU's queries using the tool and replayed to test the service under higher level of loads. The load was given on the WS from a remote windows XP machine. The hardware components in the windows XP machine consists of:

1) Intel (R) Pentium (R) Dual CPUE2200;

2) Processor Speed- 2.2 GHz;

3) RAM- 1GB;

4) Hard Disk- 150 GB.

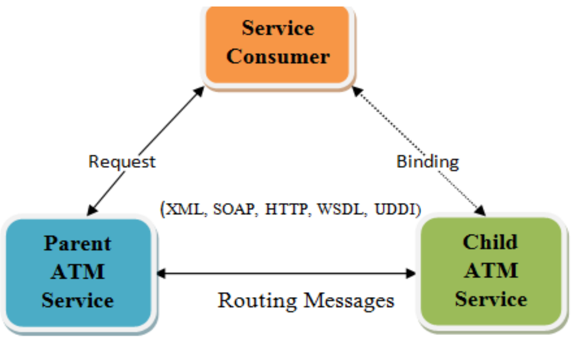

Fig. 1. Proposed E-ATM service architecture.

5. Automated Testing Tool

We have tested our proposed e-ATM service by using Mercury LoadRunner under stress of simulated VUs. Mercury LoadRunner is an industry standard performance and load testing tool for testing system behaviour and performance under stress of simulated users. The Mercury LoadRunner can simulate thousands of users using application software and users are allowed to access resources, record and latter analyse the performance of the system components [24].

5.1. Configuration of the Testing Tool

Before starting the performance test, we had configured the following parameter to emulate real like user activity and behaviour during testing.

1) Run Logic: The number of iterations.

2) Pacing: This is the controlling time between successive iterations of VU activities.

3) Think Time: The time that a user waits for performing successive operations [25].

4) Bandwidth (BW): The capacity of network that supports the transactions of VU.

5.2. Testing Parameter

We have recorded the Response Time (s), Hits/s, Throughput (byte/s), and CPU utilization which measures the quality of the system under workloads of concurrent VUs. During the load test, a user was scheduled to start in every 15 seconds and run for 300 s after completion of the ram up time. A think time of 10 s was set for every VU to pause to interact with the system.

The simulated users were used to load the system to observe the behaviour of the service applications at different loads and to measure the stress limit of the system. In this load test, the successful and failure transactions were recorded to analysis and predict reliability aspects of the system at higher loads.

6. Results and Analysis

We have conducted a series of test randomly at different stress level of VU, such as, 30, 50, 100, 300, 500, 800, 1000, 1400, 1500, 2000, 2200 and 2500 with maximum utilization of BW over the network. A test case for select operation has been prepared for accessing the service. The statistical analysis is carried out over the recorded performance attributes of the service. The different transaction status against massive stress level is presented in Table 1. For statistical analysis the test was conducted for 30 times repeatedly at 1500 VU by using the database of 15000 VUs. It has been observed that the system is responding well up to 1400 VU and consistently serving well with little refusal at larger number of workloads of VU. The average number of refusal at 1500 VU is calculated to be 1.07 %. As seen in the frequency Table 2 and Histogram in Fig. 2, the Histogram is right skewed, which is the feature of a specific distribution of data, so that we can assume that the data distribution is a Weibull distribution. The computed Bin and Frequency of QoS metrics Response Time(s), Throughput (byte/s) and Hits/s are presented in Table 3 and plotted in the Histogram as in Fig. 3, Fig. 4 and Fig. 5 respectively. Here, it is observed that the histograms for Hits/s and Throughput are left skewed and histograms for Response Time is right skewed. It predicts that the distributions of data are normal. From Fig. 6, Fig. 7 and Fig. 8 it is seen that the Throughput, Response time and Hits/s are rising with the higher number of VU. In Fig. 9 and Fig. 10 it is observed that the Response Time and Throughput are gradually rising with the higher number of Hits/s. In Fig. 10, the throughput is gradually rising against the Response Time. From these patterns of graphs we can infer that there is a close influences among the Reponses Time, Throughput and Hits/s. The parameter values are scaled up along with the progression of VU. A series of responses is recorded and shown in Table 4. Here, the Hits/s is increasing and the Response Time is very consistent and well acceptable up to 2500 VU. The throughput is increasing as expected against the larger values of VU. The CPU Time and Disk Time utilization are less than 1 % up to 2500 VU. From these out comes we can conclude that our proposed WCF based e-ATM system is quite scalable and robust against massive number of real like VU.

Table 1. The e-ATM service transactions at different VU.

Operation	VU	Pass	Fail	Total
SQL Select Operation	30	308	0	308
	50	638	0	638
	100	1897	0	1897
	300	13200	0	13200
	500	34500	0	34500
	800	85200	0	85200
	1000	131518	0	131518
	1400	254213	0	254213
	1500	267183	7884	275067
	1900	462236	1682	463918
	2000	511478	1843	513321
	2200	614424	2221	616645
	2500	774218	8148	782366

Table 2. Bin and Frequency of Failure Count.

Failure Ranges	Frequency
0-3	1
3-3757	22
3757-7511	2
7511-11265	3
11265-15019	1
>15019	1

Table 3. Bin and Frequency of QoS Metrics.

Response Time		Throughput		Hits/s	
Bin	Fre	Bin	Fre	Bin	Fre
0.039	3	48750.07	1	29.923	1
0.145	11	55263.86	3	33.869	2
0.251	9	61777.65	9	37.815	8
0.356	5	68291.43	2	41.761	3
0.462	1	74805.22	5	45.708	6
>0.462	1	>74805.22	10	>45.708	10

Table 4. QoS Parameter for the e-ATM Service.

VU	Hits/s	ResT	ThrPut	CPU-Time	Disk-Time
30	1.537	0.043	2519.825	0.139	0.011
50	2.279	0.042	3727.144	0.105	0.010
100	4.054	0.048	6630.339	0.369	0.010
300	10.595	0.040	17385.96	0.144	0.008
500	16.71	0.038	27361.17	0.302	0.011
800	26.745	0.042	43778.66	0.180	0.009
1000	31.771	0.117	51981.78	0.125	0.014
1400	46.287	0.051	75746.08	0.099	0.007
1500	49.607	0.112	81153.09	0.138	0.189
1900	61.712	0.079	100943.6	0.591	0.293
2000	61.841	0.059	101171.3	0.300	0.010
2200	68.282	1.213	111938.9	0.309	0.013
2500	77.884	2.692	127329.84	0.348	0.009

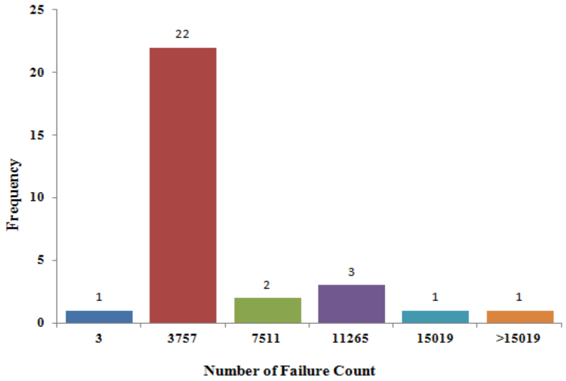

Fig. 2. Histogram for Failure Count.

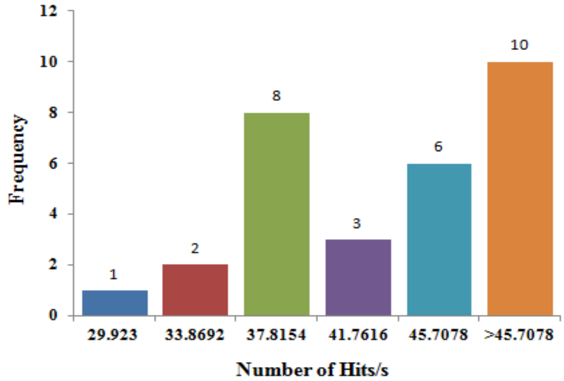

Fig. 3. Histogram for Number of Hits/s.

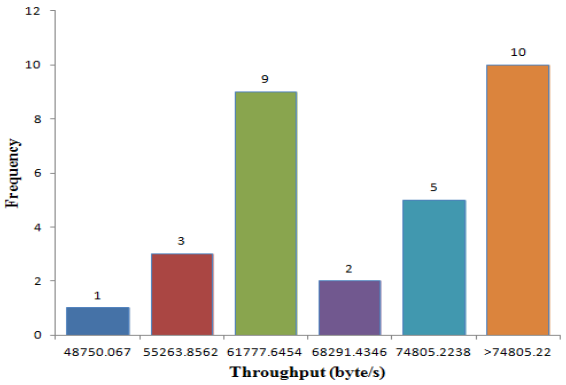

Fig. 4. Histogram for Throughput (byte/s).

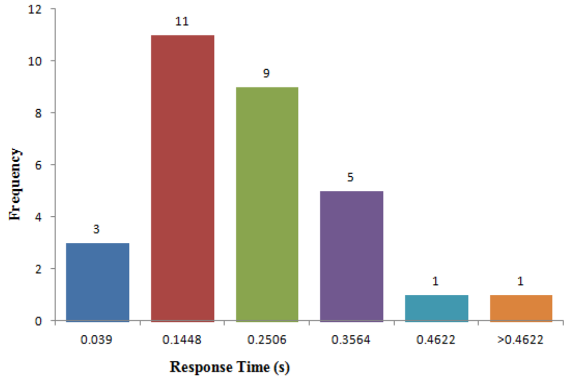

Fig. 5. Histogram for ResponseTime (s).

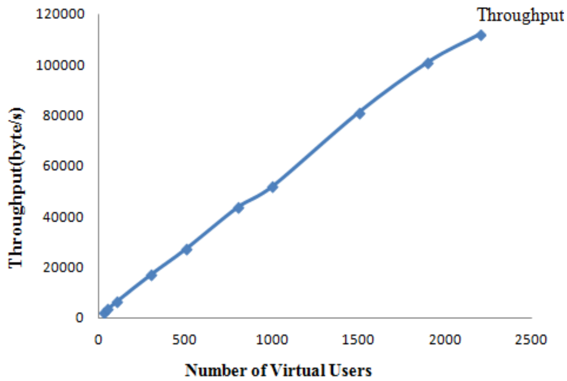

Fig. 6. Throughput (byte/s) VS Virtual Users.

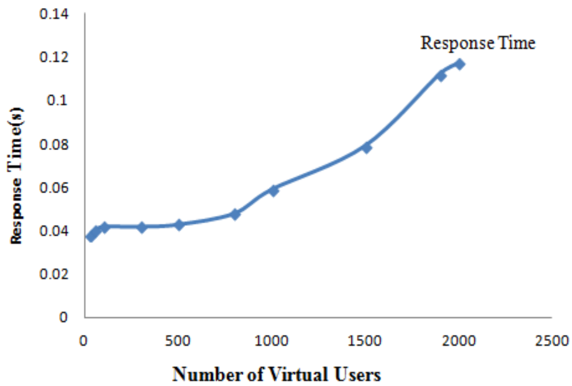

Fig. 7. Response time (s) VS Virtual Users.

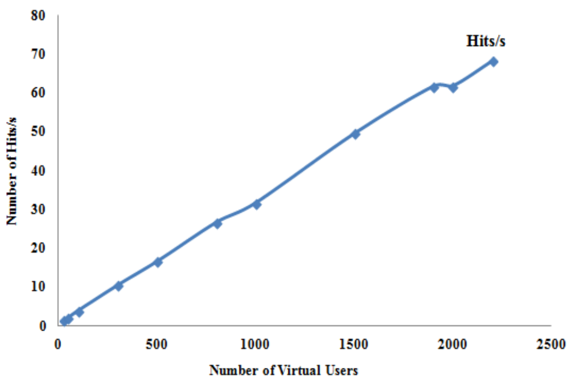

Fig. 8. Number Hits/s VS Virtual Users.

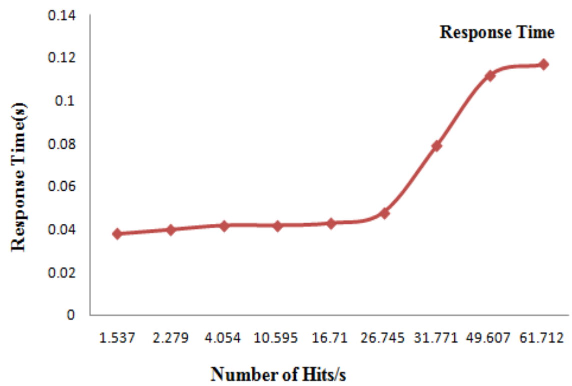

Fig. 9. Response Time (s) VS Hits/s.

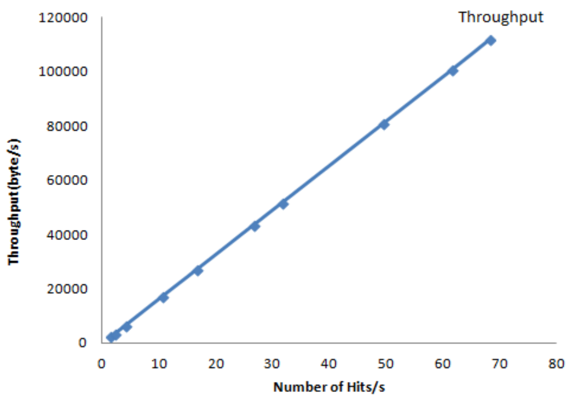

Fig. 10. Throughput (byte/s) VS Hits/s.

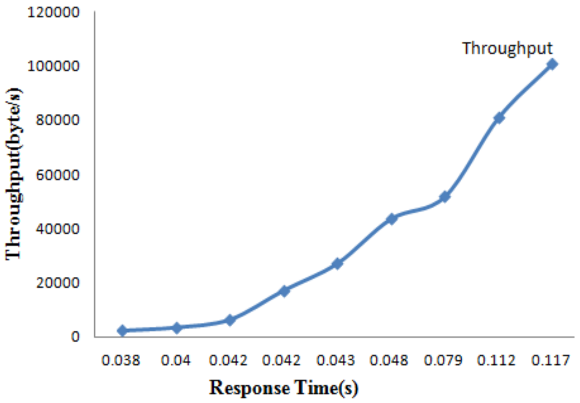

Fig. 11. Throughput (byte/s) VS Response Time (s).

To further study the distribution of data, we have used the EasyFit, a statistical software tool to estimate the values of shape (α) and scale parameter (β) which are found as α = 0.62606 and β = 2157.7 [26]. The Weibull cumulative distribution function (CDF) can be calculated by using the Equation (1) for the FC values [27].

$$\text{CDF}= 1- \exp \{-(FC/ \beta)^{\alpha}\} \qquad (1)$$

7. Goodness of Fit (GoF) Evaluation Using Kolmogorov-Smirnov Test

The Kolmogorov Smirnov (KS) GoF test is used to test whether a data sample comes from a population with a specific distribution. In the distance test, the assumed distribution is correct, if the maximum departure between the observed CDF (F_o) and the expected CDF (F_n) distributions are minimum. We have calculated the intermediate values for KS GoF test for Weibull distribution and is given in Table 5. It is seen that the KS GoF test statistic value Dmax (0.1783) is smaller than the KS table critical value (0.24) for α=0.05 and a sample of size n=30. Based on these outcome, we do not reject the hypothesis that the obtained CDF of the FC is distributed Weibull (α=0.6261, β=2157.7). Hence the population from where these data were obtained is distributed Weibully.

Table 5. Intermediate Values for K-S GoF Test.

Sl	Fo	Fn	Fn-1	D+	D-
1	0.0161	0.0333	0.0000 0	0.0172	-0.0161
2	0.0504	0.0667	0.0333	0.0163	-0.0170
3	0.1414	0.1000	0.0667	-0.0414	-0.0748
4	0.2014	0.1333	0.1000	-0.0681	-0.1014
5	0.2893	0.1667	0.1333	-0.1227	-0.1560
6	0.3982	0.2000	0.1667	-0.1982	-0.2315
7	0.4029	0.2333	0.2000	-0.1695	-0.2029
8	0.4039	0.2667	0.2333	-0.1372	-0.1706
9	0.4039	0.3000	0.2667	-0.1039	-0.1372
10	0.4130	0.3333	0.3000	-0.0796	-0.1130
11	0.4354	0.3667	0.3333	-0.0687	-0.1020
12	0.4408	0.4000	0.3667	-0.0408	-0.0741
13	0.4490	0.4333	0.4000	-0.0156	-0.0490
14	0.4599	0.4667	0.4333	0.0068	-0.0265
15	0.4787	0.5000	0.4667	0.0213	-0.0120
16	0.4814	0.5333	0.5000	0.0519	0.0186
17	0.4869	0.5667	0.5333	0.0797	0.0464
18	0.5054	0.6000	0.5667	0.0946	0.0612
19	0.5112	0.6333	0.6000	0.1221	0.0888
20	0.5182	0.6667	0.6333	0.1484	0.1151
21	0.5217	0.7000	0.6667	0.1783	0.1449
22	0.6340	0.7333	0.7000	0.0994	0.0660
23	0.6526	0.7667	0.7333	0.1141	0.0808
24	0.6889	0.8000	0.7667	0.1111	0.0777
25	0.8000	0.8333	0.8000	0.0333	0.0000
26	0.8866	0.8667	0.8333	-0.0200	-0.0533
27	0.8947	0.9000	0.8667	0.0053	-0.0280
28	0.8980	0.9333	0.9000	0.0354	0.0020
29	0.9035	0.9667	0.9333	0.0632	0.0298
30	0.9792	1.0000	0.9667	0.0208	-0.0126
			Dmax	**0.1783**	0.1449

7.1. Confidence Interval of CDF

The mean value for CDF is estimated at 95% confidence interval for 1500 VU. The populations mean values μ can be calculated by using the equation as in (2) [28].

$$\mu = \bar{x} \pm t_c SD/\sqrt{N} \qquad (2)$$

We consider the mean value of CDF as $\bar{x}$, the critical value from $t_c(0.05,29)$, the standard deviation as SD, sample size as N and the margin of error as $t_c SD/\sqrt{N}$. The estimated population mean μ is calculated and presented in Table 6. At 95 % confidence interval, the mean value of CDF lies between 0.51±0.09, i.e. 0.60 and 0.42.

Table 6. Estimated values for μ.

N	tc(0.05,29)	Parameter	$\bar{x}$	SD	tcSD/√N
30	2.045	CDF	0.51	0.25	0.09

7.2. Reliability Estimation

Reliability of a system is evaluated over phases of time and environment that the system executes without failure in the specified period. We have calculated the reliability of our proposed system by using average FC and HTTP transactions from 30 sample data i.e. FC = 2507.57, HTTP transaction = 225239.5. The probability of failure (P_f) is calculated as FC/Total HTTP transaction, i.e. 2507.57/225239.5, P_f = 0.01. The probability of success is the reliability and can be calculated as reliability (R) = 1-P_f = 98.89 % [29]. Thus the reliability can be defined as the probability of no failure within a specified period of time and stated environment. Reliability can also be estimated by using Equation (3) as in [30].

$$R=e^{-\lambda t} \tag{3}$$

Here the λ is the transactional failure rate. It is a probability density function for operational time λ (t). The execution time unit is considered as 1, so we set t=1, the reliability becomes $R=e^{-\lambda}$. In practice reliability can be estimated by using the Equation (4).

$$R=1-\lambda \tag{4}$$

Thus we obtained the following reliabilities at incremented values of VU: R(1500) = 98.89 %, R(1900) = 99.64 %, R(2000) = 99.64 %, R(2200) =99.64 %, and R(2500)=91.52 %. It is noticed that the reliability is consistent against the escalated number of VUs which indicates that the system is fairly scalable at the higher number of VUs. We performed the test at different values of VU randomly and observed that the reliability at 30, 50, 100, 300, 500, 800, 1000 and 1400 VU is 100 %. For the different level of service loads, the system will be serving as expected to the user's request. However, in higher load levels such as 1500, 1900, 2000, 2200 and 2500; the reliabilities estimated are 98.89 %, 99.64 %, 99.64 %, 99.64 % and 91.52 % respectively, i.e. more than 90 % request is served successfully. Thus, we can conclude that the reliability is quite stable and more than 90 % queries are served well in the VU range from 1500 to 2500 that is the users will get most expected responses with little refusals from the WCF service as the web server refuses at massive concurrent request.

The overall reliability evaluation framework with reliability assessment is presented in Table 7. In the FC histogram, it is observed that the highest density of FC is 22 and is occurred in the range from 3-3757 transactions. The histogram for FC follows right skewed. So, we can conclude that our FC distribution is a Weibull distribution.

Using CDF values we have computed the KS statistic by using EasyFit statistical software. It is a tool for statistical data analysis and simulation that allows for fitting probability distributions to make better decisions. From the analysis, it is revealed that the statistics value obtained (0.178) is smaller than the critical value (0.2417) at α =0.05 which indicates that the assumed samples were from a population of Weibull distribution. Based on the analysis results we can conclude that the observed data adequately fits in the Weibull distribution. We have evaluated the reliability of our proposed system by increasing VU. In lower number of VU, the reliability estimated to be 100 % and the reliability decreases at escalated number of VU.

Table 7. The reliability evaluation framework.

Parameter	Remarks
Transactions at VU 30, 50, 100, 300, 500, 800, 1000, 1400	100 % success
Transactions at VU 1500, 1900, 2000, 2200, 2500	Failures observed and increases gradually
FC Histogram at 1500 VU	Right skewed with highest failure density 22 in the range from 3-3757
GoF test using K-S at 95 % confidence level for 1500 VU	Failure data distribution qualified in the Weibull distribution
CDF	Mean CDF is 0.51 and lies in between 0.60 and 0.42
Reliability up to 1400 VU	R=100 %. Consumer will get expected services
Reliability for 1500 VU	R=98.89 %, acceptable reliability with probability of failure occurrences
Reliability more than 1500 VU	Service quality degrades gradually

8. Conclusion and Future Work

In this paper we have conducted a series of test at different levels of simulated users that accessed concurrently in the proposed e-ATM model and carried out a statistical analysis of some QoS attributes of WS. We have recorded some performance metrics which measures the quality of the system and plotted the quality metrics in histograms to see the behavior of the service under stress of higher loads. From the Table 4, it is observed that the average Hits/s is gradually increased from 1.537 at 30 VU to 60.138 at 2500 VU. The Response Time (s) was very consistent though the VU was increased gradually from 30 to 2500. The response time is recorded to be less than 1 second up to 2000 VU with a maximum of 2.69 second at 2500 VU. The Throughput is gradually increased with the progression of VU as expected, the minimum value of 2519 byte/s is observed at 30 VU. The maximum value of throughput is recorded to be 127329.84 at 2500 VU. The CPU Time and Disk Time utilization are showing the consistency over the growing limit of VU and the utilization is less than 1 % which indicates that our proposed WCF based e-ATM system is capable to handle large number of consumers on the available BW.

Using FC record and analysis approach, we have computed the reliability of our system at different levels of VU and are observed to be quite strong up to 1400 VU and slightly degraded with the progress of VU. More than 90 % services were successful up to 2500 VU. This occurrence of service failures may be attributed to database server or system resources. The experimental study revealed the strong evidence of availability, scalability and reliability of the proposed service to communicate with massive number of VU.

As part of the future work we propose to investigate some QoS aspects of WCF and webAPI Restful service which may highlight the reliability of service system based on different techniques.

Acknowledgement

The acknowledgement goes to the All India Council of Technical Education (AICTE), Govt. of India for the financial support towards the work (F. No. 8023/BOR/RID/RPS (NER)-84/2010-2011 31st March 2011).

References

[1]. W.W.W. Consortium http://www.w3.org/TR/ws-gloss

[2]. Lina Barakat, Simon Miles, Michael Luck, Efficient adaptive QoS-based service selection, *SOCA,* Vol. 8, 2014, pp. 261-275.

[3]. Marc N. Haines, William Haseman, Service Oriented Architecture Adoption Patterns, in *Proceedings of the IEEE 42nd Hawaii International Conference on System Sciences*, 2009, pp. 1-9.

[4]. Kenedy Mutange Senagi, George Okeyo, Wilson Cheruiyot, An aggregated technique for optimization of SOAP performance communication in Web services, *SOCA,* Vol. 10, Issue 3, 2016, pp. 273-278.

[5]. Ruben Casado, Javier Tuya, Muhmmad Younas, A family of Test Criteria for web services Transactions, in *Proceedings of the International Symposium on Advances in Transaction Processing*, Vol. 10, 2012, pp. 880-887.

[6]. Duhang Zhong, Zhichang Qi, A Petri Net Based Approach for Reliability Prediction of Web services, in *Lecture Notes in Computer Science*, Vol. 4277, 2006, pp. 116-125.

[7]. Gokhale Swapna S., Paul J. Vndal, Jijun Lu, Performance and Reliability analysis of web server Software Architectures, in *Proceedings of the 12th Pacific Rim International Symposium on Dependable Computing (PRDC'06)*, 2006, pp. 351-358.

[8]. Wang Haiyan, Wang Suichang, Ding Fei, Reliability Prediction–based Web Service Selection, *Advances in Information Sciences & Service Sciences*, Vol. 5, Issue 9, May 2013, pp. 391.

[9]. Hou Zhai-Wei, Zhai Hai-Xia, Gao Guo-Hong, A Study on Web Services Performance Optimization, in *Proceedings of the Third International Symposium on Electronic Commerce and Security Workshops*, 2010, pp.184-188.

[10]. Goel A. L., Software reliability Models: Assumptions, Limitations, and Applicability, *IEEE Transactions on Software Engineering*, Vol. SE 11, No. 12, 1985, pp. 1411-1423.

[11]. Abdelkarim E., Maheswary P., Vladimir T., Recovery Policies for Enhancing Web service Reliability, in *Proceedings of the IEE International Conference on Web Services (ICWS'06)*, 2006, pp. 189-196.

[12]. Shangguang Wang, Ching–Hsien Hsu, Zhongjun Liang, Multi-user web service selection based on multi-QoS prediction, *Inf. Syst. Front*, 16, 2014, pp. 143-152.

[13]. Pat P. W. Chan, Michael R. Lyu, Miroslaw Malek, Reliable Web Services: Methodology, Experiment and Modeling, in *Proceedings of the IEEE International Conference on Web Services*, USA, 2007, pp. 679-687.

[14]. Michael P. Papazoglow, Benedikt Kratz, Web services technology in support of business transactions, *SOCA,* Vol. 1, 2007, pp. 51-63.

[15]. Huiyuan Zheng, Jian Yang, Weiliang Zhao, QoS Analysis and Service Selection for Composite Services, in *Proceedings of the IEEE International Conference of Services Computing*, 2010, pp. 122-129.

[16]. Bora A., Bezboruah T., Investigation on Security Implementation and performance Aspects of MedWS: a Hierarchical SOAP based Web Service, *International Journal of Database Theory and Application*, Vol. 7, No. 4, 2014, pp. 169-188.

[17]. Medhi S., Bezboruah T., Investigations on implementation of e-ATM Web services based on .NET technique, *International Journal of Information Retrieval Research*, Vol. 4, No. 2, April-June 2014, pp. 41-56.

[18]. Maurizio Gabbrielli, Saverio Giallorenzo, Fabrizio Montesi, Service-Oriented Architectures: From Design to Production Exploiting Workflow Patterns, in *Proceedings of the 11th International Symposium on Distributed Computing and Artificial Intelligence*, Salamanca, Spain, Vol. 290, Jun. 2014, pp. 131-139.

[19]. Bezboruah T., Bora A., Performance Evaluation of Hierachical SOAP Based Web Service in Load Balncing Cluster-Based and Non-Cluster Based Web Server, *International Journal of Information Retrieval Research*, Vol. 5, Issue 4, 2015, pp. 20-31.

[20]. Bora A., Bezboruah T., Some aspects of QoS for Interoperability of Multi Service Multifunctional Service Oriented Computing, in *Proceedings of the IEEE International Conference on Research in Computational Intelligence and Communication Network*, India, 2015, pp. 363-368.

[21]. Subhash Medhi, Abhijit Bora, Tulshi Bezboruah, Security Impact on e-ATM Windows Communication Foundation Services using Certificate based Authentication and Protection: An implementation of Message Level Security based on .NET Technique, *International Journal of Information Retrieval Research*, Vol. 6, Issue 3, July-September, 2016, pp. 1-15.

[22]. Subhash Medhi, Abhijit Bora, Tulshi Bezboruah, Investigations On Some Aspects of Reliability of Content Based Routing SOAP based Windows Communication Foundation Services, *International Journal of Information Retrieval Research*, Vol. 7, Issue 1, January-March 2017, pp. 17-31.

[23]. Subhash Medhi, Abhijit Bora, Tulshi Bezboruah, Investigations on user perspective evaluation of some reliability aspects of web services, in *Proceedings of the 1st International Conference on Advancement of Computer Communication & Electrical Technology (ACCET'16)*, 21-22 October 2016, pp. 65-69.

[24]. Retrieved at http://genesis.co.kr/image/product/pds/LoadRunner/Manual/LoadRunner_Generator.pdf
[25]. Loadrunner, retrieved from http://www.en.wikipedia.org/wiki/HP_LoadRunner, 2017.
[26]. EasyFit Distribution tools at http://www.mathwave.com/easyfit-distribution-fitting.html, Retrieved on 2017.
[27]. Chandra M., Singpurwala N. D., Stephen M. A., Kolmogorov Statistics for Test of Fit for the Extreme Value and Weibull distribution, *Journal of the American Statistical Association*, Vol. 76, No. 375, 1981, pp. 729-731.
[28]. Speigel M. R., Theory and problems of probability and Statistics (Schaum's Outline Series), 2 ed., *Schaum's Outlines*, 2000.
[29]. Shooman M. L., Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design, *John Wiley & Sons, Inc.*, 2002.
[30]. Lijun Wang, Xiaoying Bai, Lizhu Zhou, Yinong Chen, A Hierarchical Reliability Model of Services Based Software System, in *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference*, Vol. 1, 2009, pp. 199-208.