

Experimental Study on Mail Services: A Comparison with Activity-based Performance Prediction

Ch. Ram Mohan Reddy
Research scholar
M S Ramaiah Institute of
Technology, Bangalore – 54
India

K. Sailaja Kumar
M S Ramaiah Institute of
Technology, Bangalore – 54
India

D. Evangelin Geetha,
T.V. Suresh Kumar
M S Ramaiah Institute of
Technology, Bangalore – 54
India

ABSTRACT

Nowadays, mail services are used rigorously for communication purpose. Due to the widespread demand for mail services, performance degradation may occur for mail servers. Performance is an open issue that is affected by many factors including the technical factors. To identify the factors that have an impact on the performance of the mail services, we have carried out an experimental study by focusing on few prominent mail services. In this paper, the results of the experimental study and the results obtained from activity-based performance prediction approach are compared and discussed. Regression analysis is used for comparison, and the obtained value shows that both are closure to each other.

Keywords

Web applications, Mail Services, Software Performance Engineering, Experimental Study.

1. INTRODUCTION

These days every person has an email address. In fact, many users have more than one email address, and all these services are provided for free by many service providers. The E-mail service provider has to be more efficient and should satisfy all the clients requirements and also efficiently handle all issues regarding scalability, reliability and load balancing activities of customers on the service providers [25]. E-mail is the most popular way for information exchange through the communication networks. E-mail [11] is the most popular way to share information through the communication channels that solve the limitations of distance and time. In 1982, the standard protocol and the message format for E-mail was announced as RFC 821/822 [8, 12]. The existing E-mail system operates as two processes, which are message transport agent (MTA), and user agent (UA). UA provides a user interface for building and reading a text message. MTA on a sender carries a message to MTA on a recipient through networks. The standard protocol for message transporting is simple mail transfer protocol (SMTP). In the E-mail system, the UA, placed between the MTA and a user, plays a significant role. The E-mail system [21] provides GUI to ease a user's authoring or viewing a multimedia message. The email services had many defects regarding governing the privacy, authentication of its users, it was first used only by students. But because of its high usability and ease of use it gained considerable importance [9].

A mail service is an application that receives an incoming e-mail from local users (people within the same domain) and remote senders and forwards outgoing e-mail for delivery. The web application does not interact with the mail server; instead it just fills a user's table in a database. In the simplest case this table has a username field and a password field; need

a web service to communicate with the mail server without which the communication between two peers will not complete successfully. The email services are influenced by many environmental and technical factors like distributed systems, security features, complex internal processing, and concurrency of the software systems.

A web application has a user interface that can be used by any user on the internet. Any such use is called a web application. A web service sees that a web application works in the same way as intended when opened on different platforms over the net. At a low level, both Web application and web service are on the same thing. Since HTTP protocol is stateless protocol any service that is developed on HTTP. SOAP protocol is stateless service. SOAP protocol is dependent on HTTP. They both operate over HTTP(s). Web services use soap protocol for communication of messages. These technical factors will also have an impact on protocols like HTTP and SOAP etc. since both rules operate on HTTP's.

Performance is one of the key requirements of any software product and it is the primary factor considered in the arena of computing and also in user satisfaction. The mail system is a combination of hardware and software systems and its interactions. It also depends hugely on the user interaction and using the interfaces designed the user gets to use and implement both web applications in the foreground and web services in the background. Here, the performances of typical mail systems rely on standard Internet protocols. During the deployment of web services, the technical factors may have an impact on the performance of mail servers/services and the corresponding protocols. Hence, it is important to assess and predict the performance of mail services at the preliminary design stage and during the entire life cycle.

Any software is engineered or created with the intention of solving an underlying problem for which the software is developed. Software Performance Engineering (SPE) is a method for constructing such efficient and reliable software systems [4, 5]. The advantage of SPE is that it gives an idea about the performance of the software at a very early stage by giving an idea of all bad returns so that the developer can save time and change his design of the software. In SPE approach, various hardware resources and software requirements are to be considered. These requirements differ based on the category of the software systems. The objective is to find out the factors that have an impact on the performance of the mail services. To identify those factors we have carried out an experimental study by considering few majorly used mail services. The performance of mail services has to be evaluated at an early stage as after development if the service displays any issues; it will be a severe problem. The performance of mail services can be estimated using an experimental study

that helps to identify the working of the service and identify all the possible performance issues that can happen after development that leads to giving the developer enough time and support to change the structure and design of the software. An experimental setup can be established by different combinations of the hardware resources and the working of the software before development can be understood [3]. The approach is experimental, that is, based on the analysis of measurements collected for mail services from the system. In this paper, we have presented the results and observations. The observed performance parameters are compared with the estimated values obtained using activity-based performance prediction approach.

The rest of this paper is organized as follows. Section 2 briefly describes the related work on performance analysis of web services and SPE. Section 3 describes the proposed methodology to analyze the performance metrics. Section 4 describes the algorithm based on the methodology. Section 5 provides the results of the experimental study and the analysis of the results. Section 6 compares the results with Activity Based Performance Prediction (ABPP) Approach. Section 7 gives the analysis of the acquired results through different aspects. Section 8 concludes the paper with future work.

2. RELATED WORK

We surveyed the literature available in the area of SPE and web services. The review of the literature is as follows. The design of mail systems and the analysis of their performance is given in [2]. In this paper, a software lotus is used. There is a lot of workload on all the mail servers, and Lotus Notes the services and is designed by balancing this workload. It also uses the help of standard mail protocols to estimate this workload in [14]. The experimental setup is hence formed and studied, and the results are used and considered for the experiments. A high-availability high-performance email cluster is presented in [23]. The design of the system is based on the analysis of user needs and behavior. Benchmarking of mail servers is also addressed in [16], and a prototype of a benchmark tool is presented. The performance evaluation of mail services that rely on the Internet standard protocols SMTP; POP3 and IMAP4 are focused in [17]. The analysis of the workload of these systems shows the properties of each type of workload and the behavior of the users.

The application of Software Performance Engineering at design phase for real time/precise use is discussed in [4]. It explains how to compare performance alternatives of design characteristics by integrating SPE with design methods. SPE can also be used in implementation phase to ensure that performance goals are met. The authors have demonstrated that the information obtained from SPE can be used for identifying critical components for prototyping. The efficiency of the Web application is enhanced using the support of SPE wherein there are specifications as to how to design efficient software that is more effective in the areas of performance, responsiveness and scalability in [5]. This paper employs an approximation technique to obtain timely feedback from SPE models. It also explains the types of information the model provides and to evaluate the performance improvement made to the web application at the architectural phase of development.

A web services based infrastructure for Clinical Decision Support System for performing multi-domain medical data from obstetrical perinatal and neonatal care domains and the application of SPE on the infrastructure is proposed in [6]. The goal of this foundation is to reduce medical errors, to

support physician decision –making process and improve ultimately patient care. The SPE is applied at design phase to meet the performance requirement and evaluating it using layered queue network model. The response time is sensitive to both the web service processing time and the size of the electronic patient records stored in the database.

The overhead imposed on web services is non-negligible during server overload as the response time of web service is 30% higher, and the server throughput is 25% lower compared to traditional web interaction due to dynamically generated HTML pages is explained in [15]. A closed loop tool does not help to capture the real behavior of web interaction during overload.

The impact of system performance when interfacing web services with the tightly coupled application using implementation variants of Sun's Java Pet Store application using Web service interface and evaluates performance on responsiveness, overall throughput, and latency is discussed in [13]. The changes in the response time are less uniform and require high CPU consumption. This variant exhibits 30 percent longer execution interrupts for performing garbage collection and for non-transactional execution, the performance is minimal.

The performance analysis on the client side and server side of a web service are given in [18]. The performance of the prospective customers can be estimated using client grouping and profiling. It presents the common factor for client cluster and profiling and effect of the same on performance by WS client like geographical location, CPU load. This information also serves as performance feedback reporting for the web service providers.

Any web service that is available on the internet can be evaluated by assessing four significant aspects such as number, quality of service, complexity and function diversity in [24]. The study explains that average size and number of operations in WSDL files are small, there is no adequate description of the web services in WSDL files, the web service covers limited portion of application domain

A visual simulator that determine the impact of the individual utility metrics on the total number of hit count values based on the survey results is discussed in [1]. It aims at developing a metrics model for determining the number of visitors from various domains who expects most recent and updated information the user is searching for.

Analyzing the service problem of a virtual organization using distributed service and providing a graphic solution for a typical problem for user application by outlining the generic architecture is in [19]. Using this model setup we can get readings and provide dynamic solutions by using random users at different geographical locations and mashup technique.

To provide desirable Business to Business (B2B) solution for the services rendered by the web application to clients which support Business to Client (B2C), a portal is described in [20]. It exhibits web application as SOAP based services and presents an approach to replicate the portal to represent a single point of failure. A SOAP optimization technique is proposed in [10] is to improve the speed of processing SOAP messages. It introduces client usage pattern to the low-level SOAP process infrastructure that reduces the number of SOAP messages and improves latency on the client side. It reduces the services concurrency and reduces the number of SOAP messages transferred between clients and services.

As an example [22] takes an e-commerce application that uses a web service component and its performance is measured using a measurement-based performance analysis. From the setup established it was concluded that higher workload intensities increase the response time of the web services when compared to any other component.

3. METHODOLOGY

The proposed methodology is to analyze the performance metrics for important services through the experimental setup. It provides a well-defined procedure to obtain the performance metrics for the components of the services in different time periods and analyze their behavior. It also facilitates to compare the data obtained in an experimental study with the results obtained using activity-based performance prediction methodology [7].

The following are the steps to be followed:

- Select the servers of the same category (i.e. which provide the same kind of service. e.g. mail services) for which performance characteristics to be analyzed.
- Select the operational environment of the experimental setup.
- Obtain the page size and response time for the critical activities involved in the servers at different time stamps.
- Tabulate the data and plot graphs.
- Analyze the results and give the observations.
- Apply activity based approach and obtain the page size and response time by considering technical and environmental factors and complexity of actor and activities as given in Table 1, Table 2, Table 3 and Table 4 respectively.
- Compare the results with experimental results.
- Repeat the above steps for different operational environments

Table 1 Weighting factor for Actor

Category of actor	Representation	Weighting Factor
Simple	Application Programming Interface (API)	1
Average	System interacting with protocol (TCP/IP)	2
Complex	Person interacting with GUI or web page	3

Table 2 Unadjusted Weights for Activities

	Database/Classes	1	2	3 or more
User Interface				
Simple		1	2	3
Average interface design		2	3	4
Complex		3	4	5

Table 3 Environmental Factor's

Factor	Description	Weight
--------	-------------	--------

E1	Familiar with Rational Unified Process	1.5
E2	Application experience	0.5
E3	Object-oriented experience	1
E4	Lead analyst capability	0.5
E5	Motivation	1
E6	Stable requirements	2
E7	Part – time workers	-1
E8	Difficult programming language	2

Table 4 Technical Factor's

Factor	Description	Weight
T1	Distributed system	2
T2	Response or throughput performance objectives	2
T3	End-user efficiency	1
T4	Complex internal processing	1
T5	Reusable code	1
T6	Easy to install	0.5
T7	Easy to use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Includes security features	1
T12	Provides access for third parties	1
T13	Special user training facilities are required	1

4. ALGORITHM

The algorithm of the methodology is given below:

```

Select the servers for which the performance metrics to be analyzed.
    Consider the operation environment.
for each server application {
    consider the critical activities:
        for each activity {
            obtain the page size and response time at different time stamps.
        }
}
Analyze the results to express the user's perception of high level TF's in mail Services.
for each activity {
    Obtain the size and response time using activity based performance prediction approach.
}
Compare the results with experimental data.
    
```

5. EXPERIMENTAL STUDY

A web server is a program that serves the files/web pages to web users. Two leading web servers are Apache and IIS, whereas a mail service is an application that receives an incoming e-mail from local users and remote senders and forwards the outgoing e-mails. Microsoft Exchange, Gmail, Exim, and Sendmail are among the more common mail service programs.

The design issues of these services have an impact on its performance, especially response time. The factors that affect the performance are many; mainly depending on the size of the text and images of web page size, the technical factors/ environmental factors, deployment and operational environment. Identifying the behavior of these factors is an essential process to build responsive mail services. To realize the behavior of these parameters we have carried out an experimental study on the web services that are widely used. Few prominent mail services are considered for the study. The components of the execution environment namely, the client machines, configuration, bandwidth of the network and the web page size are considered for carrying out the experiment. The mail services; Gmail, Yahoo, Hotmail, and Rediffmail, are focused on studying the performance behavior considering the technical factors namely Distributed Systems, Response or throughput performance objectives, End-user efficiency, Complex Internal Processing, portable, concurrent, etc. It is important to note that all these mail services are accessed at the same time for the same purpose with the limitations of unknown backend deployment environment.

5.1 Selection of the Services

We have considered the mail services: Yahoo, Rediff mail, Gmail, and Hotmail, which are typical web applications in nature that describe the high-level technical factors (TFs) like Distributed Systems, Response or Throughput Performance Objectives, End-user efficiency, Complex Internal Processing, Portable, Concurrent, etc.

5.2 Operational Environment

To analyze the impact of various technical factors on the performance of mail services, we have considered different client configurations and bandwidths.

Hardware environment

Though the deployment environments of the web services are difficult to be known, the study is carried out by changing the configuration of the client machines and network bandwidths.

The following three different client configurations are used for the study:

- C1: Intel(R) Pentium(R) 4 CPU @3.20GHz 1GB RAM
- C2: Intel(R) Core(TM) 2Quad CPU Q6600 @2.40GHz. 2 GB RAM For C1 and C2, the following bandwidths are considered for the analysis; 14.4 Kbps, 28.8 Kbps, 33.6 Kbps, 128 Kbps and 1476.56 Kbps.
- C3: Intel(R) Core(TM) 2Quad CPU Q9500 @2.83GHz. 2 GB RAM Bandwidth 5 Mbps

5.3 Software Requirements and Performance Metrics

The activities that are involved in the mail services: "Authenticate User Process" such as Home Page Load Time (HPLT), Registration Page Load time (RPLT), Register Time

(RT), Confirmation Time (CT) and Login Page Load Time (LPLT) are considered for analysis. The web page size of the web services is obtained from [www.websiteoptimization.com]. The response time for the activities is obtained by using firebug tool from Firefox.

5.4 Results and Discussion

Obtained the home page size and download time of home page of various mail services considering the configuration C1 and the results are presented in Table 5. From this Table, it is observed that, if the image size is high, the download time is also proportionally high. This feature can be captured by assigning high rate for the technical factor complex internal processing. The column Image size / Code size provides interesting observations. For Hotmail, even though the proportion of image size to HTML code size is high, the amount of data downloaded per second is less, compared to Rediffmail. This observation may lead to the information that the hardware configurations in deployment environment of the former may be higher than the latter. The correlation between the home page size and download time is estimated, and the value of correlation coefficient (r) is obtained as 0.98. It shows that both are highly correlated.

Table 5 Various Parameters of Mail Services (C1 with 14.4 kbps)

Mail Service	HTML Code Size	Image Size (kb)	page size (kb)	Download time	pz/d t	Image size/ Code size
Gmail	25	6.63	31.63	10.09	3.14	0.27
Rediff-mail	44	76.8	120.8	19.85	6.09	1.75
Hotmail	14	192	206	43.76	4.71	13.71
Yahoo	175	341	516	253.78	2.03	1.95

Table 6 Response Time for Mail Services Considering Various Bandwidths for Configuration C1

Mail Service	Bandwidth Rate (kbps)					
	14.4	28.8	33.6	56	ISDN 128	1476.56
HPLT(Seconds)						
Yahoo	253.78	133.49	116.31	75.06	32.15	14.84
Rediff-mail	19.85	10.73	9.42	6.29	3.04	1.77
Gmail	10.09	5.34	4.67	3.04	1.35	0.66
Hotm-ail	43.76	22.28	19.21	11.85	4.18	1.09

Table 7 Response Time for Mail Services Considering Various Bandwidths for Configuration C2

Mail service	Bandwidth Rate (kbps)					
	14.4	28.8	33.6	56	ISDN 128	1476.56
HPLT(Seconds)						
Yahoo	255.90	134.55	117.22	75.61	32.31	14.86
Rediff-mail	20.88	11.24	9.86	6.56	3.12	1.73

Mail service	Bandwidth Rate (kbps)					
	14.4	28.8	33.6	56	ISDN 128	1476.56
HPLT(Seconds)						
Gmail	10.09	5.34	4.67	3.04	1.35	0.66
Hotmail	43.77	22.28	19.22	1.85	4.18	1.09

The Home Page Load Time (HPLT) of mail services is obtained considering the configurations C1 and C2 and is presented in Table 6 and Table 7 respectively.

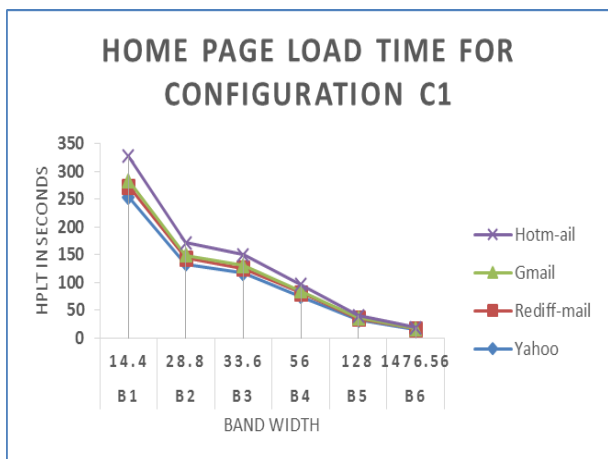


Figure 1 Home Page Load Time for various Mail Services considering Various Bandwidths for Configuration C1

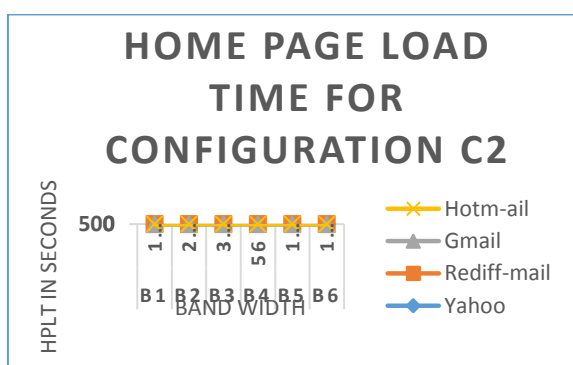


Figure 2 Home Page Load Time for various Mail Services considering Various Bandwidths for Configuration C2

From Table 6 and Table 7, it is observed that higher the bandwidth, lower the HPLT. Even though the network bandwidths are different, the increase or decrease in the HPLT is proportionate to each other. It can be observed from the graphs presented in Figure1 and Figure 2.

The response time for the activities RPLT, RT, CT and LPLT of “Authenticate User Process” is obtained using different mail services against different timestamps: T1 (10.20 a.m.), T2 (01.50 p.m.) and T3 (04.20 p.m.) and results are presented in Table 8, Table 9 and Table 10 respectively. Graphs are generated considering the response time of each activity and are presented in Figure 3. From the graphs, it is observed that the time taken during T1 is higher than the other two timestamps T2 and T3. It may be due to many concurrent users during that time stamp. This feature can be contributed to the technical factors, Distributed Environment, Response or Throughput Performance Objectives and Concurrency in the activity-based prediction performance approach.

Table 8 Response Time for Mail Services by considering configuration C3 in Timestamp T1

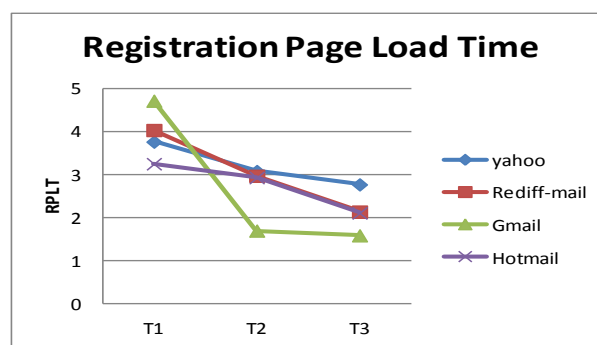
Mail Service	Response Time			
	RPLT (sec)	RT (sec)	CT (sec)	LPLT (sec)
Yahoo	3.76	4.12	3.50	2.77
Rediff-mail	4.03	4.34	2.34	2.55
Gmail	4.71	4.46	3.77	1.73
Hotmail	3.25	2.71	2.38	1.56

Table 9 Response Time for Mail Services by Considering Configuration C3 in Timestamp T2

Mail Service	Response Time			
	PLT (sec)	RT (sec)	CT (sec)	LPLT (sec)
Yahoo	3.08	2.28	5.44	2.36
Rediff-mail	2.97	2.93	2.69	2.55
Gmail	1.70	2.63	2.73	1.46
Hotmail	2.93	2.80	5.53	2.77

Table 10 Response Time for Mail Services by Considering Configuration C3 in Timestamp T3

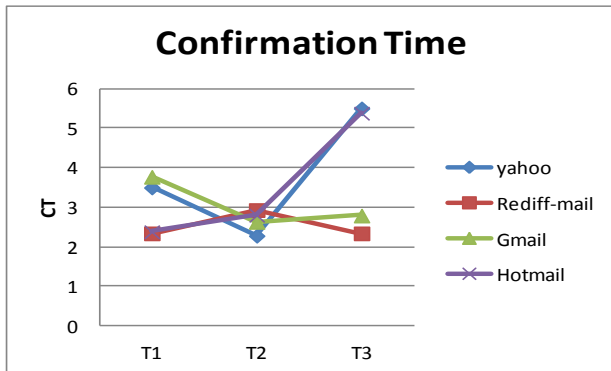
Mail Service	Response Time			
	PLT (sec)	RT (sec)	CT (sec)	LPLT (sec)
Yahoo	2.77	2.28	5.50	2.77
Rediff-mail	2.14	2.67	2.34	2.55
Gmail	1.59	2.63	2.79	1.75
Hotmail	2.11	2.55	5.38	2.56



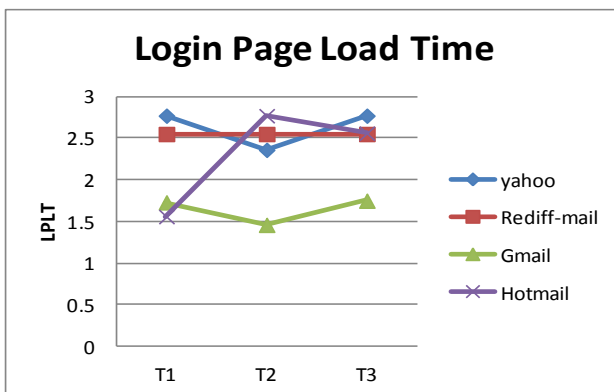
(a)



(b)



(c)



(d)

Figure 3 (a-d) Response time for RPLT, RT, CT and LPLT in Mail Services at Time Stamps T1, T2, and T3

6. ACTIVITY-BASED APPROACH

Activity based performance prediction (ABPP) for estimating the performance of activities of software systems. The methodology assesses the performance of software systems by considering the type of the users, activities of the software systems, Technical factors and Environmental factors that are available during feasibility study or preliminary design phase. It provides to take a decision on the software architecture considering the performance of software systems. It is based on the notion that amount of resources (efforts) required to execute each activity in the software system can be quantified and therefore, the response time for each activity can be assessed [7].

We have obtained the response time for the activity LPLT using the “Activity Based Performance Prediction (ABPP) Approach” and the results are presented in Table 11. The correlation coefficient between them is estimated by identifying the closeness between the response time obtained using experimental study and ABPP approach. The correlation coefficient (r) is 0.980748 that shows that both are highly correlated.

Table 11 Response Time Statistics for the Activity LPLT using Activity Based Approach

Mail Service	Response Time for LPLT(sec)	
	Experimental Setup	Activity-Based Approach
Yahoo	0.65	0.325
Rediff-mail	0.55	0.294
Gmail	0.122	0.241
Hotmail	1.66	0.403

7. STATISTICAL ANALYSIS

It is very much required to examine the data to obtain relevant statistics. Once the data has been analyzed, the analysis results are used to find the similarities or differences between the parts of the data and ensure that data are interpreted correctly and in a meaningful manner.

The statistical values minimum, maximum, mean and median are calculated for the response time of various activities of “Authenticate User Process” taken from Table 8, Table 9 and Table 10 respectively. The statistical values are tabulated in Table 12, Table 13 and Table 14 respectively. This analysis helps to identify the best-average-least response time. Also, the maximum response time obtained will be helpful to define the performance goal of the mail service.

Table 12 Response Time Statistics for the Activities at Time Stamp T1

Statistics	RPLT (sec)	RT (sec)	CT (sec)	LPLT (sec)
Min	3.25	2.71	2.34	1.56
Max	4.71	4.46	5.92	2.87
Mean	3.836	3.692	3.582	2.296
Median	3.76	4.12	3.5	2.55
Standard Deviation	0.6081	0.8107	0.7445	0.5969

Table 13 Response Time Statistics for the Activities at Time Stamp T2

Statistics	RPLT (sec)	RT (sec)	CT (sec)	LPLT (sec)
Min	1.7	2.28	2.69	1.46
Max	3.08	2.93	6.65	2.77
Mean	2.712	2.708	4.608	2.344
Median	2.93	2.8	5.44	2.55
Standard Deviation	0.6498	0.2815	1.6027	0.5749

Table 14 Response Time Statistics for the Activities at Time Stamp T3

Statistics	RPLT (sec)	RT (sec)	CT (sec)	LPLT (sec)
Min	1.59	2.28	2.34	1.75
Max	2.77	2.87	6.17	2.98
Mean	2.188	2.6	4.436	2.522
Median	2.14	2.63	5.38	2.56
Standard Deviation	0.4829	0.1756	1.6710	0.4499

From Table 12, Table 13 and Table 14, it is observed that for the activities RPLT, RT, and CT the response time is high compared to the activity LPLT. This is due to the technical factors: Distributed System, Complex Internal Processing, Concurrency, User Efficiency, Security and Third parties involvement in executing these activities. Also, it is observed that the response time for the activity LPLT is low compared to other activities at all the time stamps. It is also observed that the response time for the activity CT is high during the time stamp T3 compared to T1. It may be due to heavy workload during that time stamp T1. The technical factor concurrent can capture it in executing this activity.

7.1 Regression Analysis using statistical software ‘R.’

The statistical tool ‘R’ is the most versatile statistical open source computing environment. In R, the function lm() is used to fit a linear model. It is used to carry out the regression analysis with the help of regression model.

The response time for the activity RPLT from different mail services and the associated page size in mail services at time stamp T1, considering the configuration C3 are tabulated in Table 15.

Table 15 RPLT and Page Size in Various Mail Services

Mail Service	Registration Page Size (PZ) Kbytes	RPLT(sec)
		T1
Yahoo	99.9	3.76
Rediff-mail	48.6	4.03
Gmail	101.8	4.71
Hotmail	70.3	3.25

The data for regression analysis is considered from Table 11. The regression analysis results are obtained using the function lm() in R for the activity RPLT at different time stamps for different mail services are presented below:

Table17 Random Numbers Generated Using Triangular Distribution for the activity RPLT

Parameters of Triangular distribution							
Minimum (a)	3.25						
Most Likely (c)	3.76						
Maximum (b)	4.71						
Random numbers generated considering the above parameters: File name: Data1							
1.	2.	3.	4.	5.	6.	7.	8.
2.	3.951349	3.800766	4.460566	3.73499	3.933655	4.163724	3.729812
3.	3.861614	4.033501	3.783956	3.492024	3.913766	4.298872	3.348957

#Regression Equation for the data collected at timestamp T1
#Coefficients:

$$\text{(Intercept) size} \\ 3.23621 \quad 0.00875$$

From this output, we can determine that the intercept is 3.23621 and the coefficient for the page size (PZ) are 0.00875. Therefore, the complete regression equation is

$$\text{RPLT} = 3.23621 + 0.00875 * \text{PZ} \text{----- (1)}$$

Equation (1) represents the predicted RPLT for the mail services for specified PZ. It is observed that the RPLT will increase by 0.00875 as the PZ increases at time stamp T1. The intercept value may be due to the technical factors Distributed Environment and Concurrency.

We used the above regression equation to predict the RPLT considering various PZs at stamp T1. The results are listed in

Table 16. From the results obtained we observed the fact that as the page size increases exponentially the response time also increases gradually at time stamps T1. It is because some users using the mail services will be large in number, and the number of requests to load the page will be more. The technical factors required in this case are to be addressed to optimize the response time.

Table16. Predicted Response Time for RPLT

Page Size (Kbytes)	Response Time Taken at Time stamp T1
10	3.323709
100	4.111182
1000	11.985911

7.2 Probability Analysis

To address the uncertainty associated with the response time, we used probabilistic analysis considering the probability of response time. The triangular distribution is opted to model the inputs since the samples collected were not sufficient enough to make the prediction and also the uncertainties in the data. By using the triangular distribution, both the minimum and maximum values can be produced easily.

The response time of the activity RPLT is considered as a random variable to be modeled using the triangular distribution since it is observed from the data that the most suitable distribution is the Triangular. The Minimum, Maximum and Most-likely values obtained from the data are tabulated in Table 17. Also, the random numbers obtained using the triangular distribution is listed in the Table.

4.	3.879495	3.580811	4.031066	3.537616	4.312165	4.01723	4.065328
5.	3.988588	3.635906	3.941687	3.8587	3.537105	4.390815	4.090809
6.	3.840183	4.285977	3.838866	3.933928	3.645249	3.864034	3.823312
7.	3.725437	4.139364	3.932127	4.165761	4.584883	4.11152	4.192861
8.	3.801791	3.491593	4.145701	4.38955	4.146242	3.93892	3.687479
9.	3.532747	3.835368	4.379883	3.926122	3.394587	3.717711	3.709303
10.	4.398963	3.923043	3.626239	3.476859	3.898217	3.891421	4.227916
11.	4.110522	4.426664	4.11217	3.793873	3.997173	4.138487	4.053387
12.	3.664049	3.955863	4.108881	4.030067	4.41928	3.545763	3.681179
13.	4.289926	4.226673	3.672889	3.779176	3.490397	4.601934	3.898504
14.	3.617888	4.42118	4.053646	4.169679	4.57289	4.159058	4.308132
15.	4.052279	4.409934	3.613838	4.646613	3.554513	4.628422	3.759898
16.	3.710414	3.906087	3.634879	3.692963	3.599799	3.33553	3.802771
17.	3.909307	3.948965	4.264652	3.838338	3.704745	4.055706	3.764333
18.	3.441278	4.382947	4.531727	4.073981	3.734395	4.104938	3.650098

7.3 Statistical measure for distribution fitness

We need to conduct the goodness-of-fit test to evaluate the chosen distribution and the associated parameters. The random numbers generated from the input model are evaluated using the chi-square goodness-of-fit test to accept the hypothesis about the distribution to be triangular.

The `Chisq.test()` function in R is used for both the goodness of fit test and the test of independence, and which it does depends on what kind of data to be tested. These are a goodness of fit test with equal expected frequencies. The "p" vector does not need to be specified since equal frequencies are the default.

The Chi-Square summary for the triangular distribution fitting the data provided in Table 17 is obtained using the statistical tool R and it is presented below:

```
#Chi-squared test for given probabilities
data: Data1
X-squared = 2.1259, df = 96, p-value = 1
```

From the above results it is observed that the calculated χ -squared value is 2.1259, the degrees of freedom for the tabulated value is 96. The p-value obtained is one which is greater than the 0.05 significance level; hence we can consider the input model that says that the data generated using the input model will follow the triangular distribution. Hence, the chosen distribution is a good approximation of the RPLT data. A similar methodology can be used for the remaining data sets also.

7.4 Input Validation using Arena Input Analyzer

The data generated using the input model by considering the activity RPLT is validated using the simulation tool Arena. The Arena's Input Analyzer is used to validate this data. The best fitting solution suggested from the generated histogram is the Triangular Distribution. The histogram summary is provided in Figure 4. The histogram summary suggests that the distribution is Triangular. The distribution and data summary is given in Figure 5. A similar approach can be used to validate the data using the input model for various performance metrics.

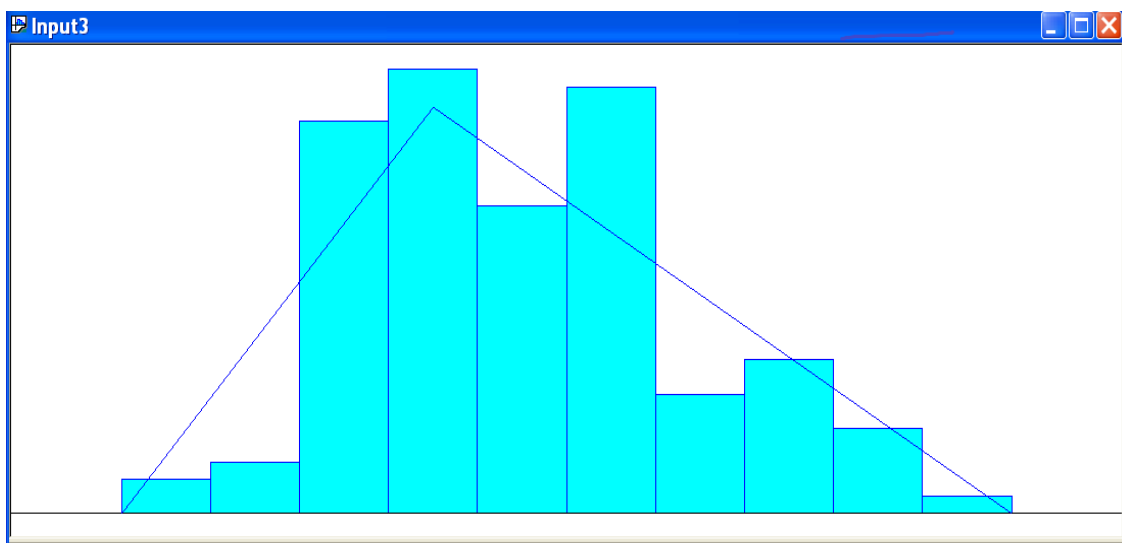


Figure 4 Histogram Summary for the Triangular Distribution

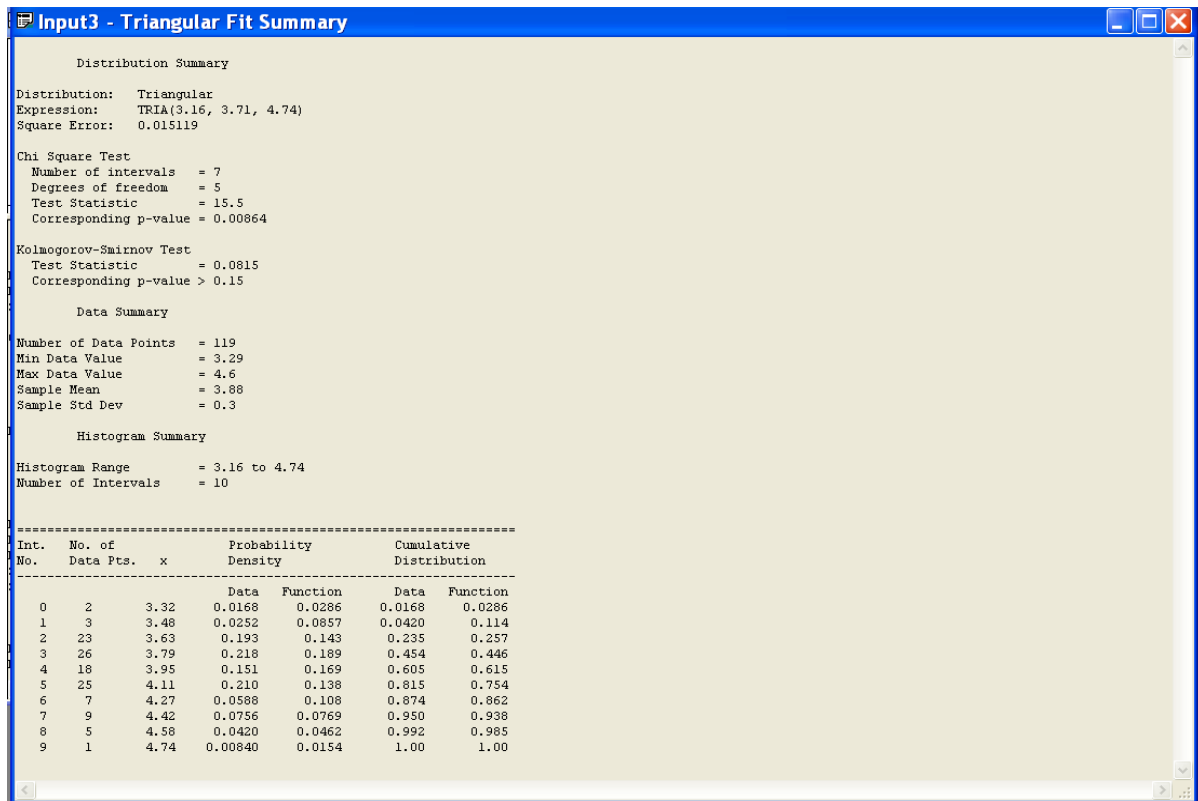


Figure 5 The distribution and Data summary for the Triangular Distribution

8. CONCLUSION AND FUTURE WORK

The technical and environmental factors of software systems that affect the performance of web services (mail services) are studied by considering the prominent mail services: Gmail, Yahoo, Hotmail and Rediffmail. To realize the behavior of these factors in mail services an experimental study is conducted for the activities involved in “Authenticate User Process.” The response time obtained for various activities using these mail services at different time stamps are analyzed. It is observed that the response time is varying due to the technical factors namely Distributed Environment, Response or throughput performance objectives, Complex Internal Processing, Concurrency, type of the user, time of the day, web page size, etc. Hence to obtain the optimal response time the user has to address these factors that in turn improve the performance of web services. Also, the uncertainty associated with the response time is analyzed and addressed using probabilistic analysis, and the simulated results are evaluated and validated with the help of prominent tools R and Arena for the statistical analysis. The works do have limitations like uncertainty about the deployment environment. As the future work, we simulate the deployment and operational environment for varying workload and analyze the behavior of hardware resources.

9. REFERENCES

- [1] Baskaran Alagappan, Murugappan Alagappan, S.Danish Kumar, 2009. “Web Metrics based on Page features and Visitor’s Web Behavior,” Second International Conference on Computer and Electrical Engineering, IEEE.
- [2] B.Pope, “Characterizing Lotus Notes Email Clients”, 1998. In Proc. IEEE, Third Int. Workshop on Systems Management, Pages 128-132.
- [3] Ch. Ram Mohan Reddy, D Evangelin Geetha, T V Suresh Kumar, K RajaniKanth, 2014. “Performance Analysis of Web Services: An Experimental Study,” Proceedings of International Conference on Circuits, Communication, Control and Computing (I4C 2014), 21-22 November.
- [4] Connie U. Smith, Senior Member, ZEEE, and Lloyd G. Williams, 1993. “Software Performance Engineering: A Case Study Including Performance Comparison with Design Alternatives,” IEEE Transactions on Software Engineering, VOL. 19, No. 7.
- [5] Connie U. Smith, Lloyd G. Williams, 2000. “Building Responsive And Scalable Web Applications,” By Software Engineering Research And L&S Computer Technology, Inc. In Proceedings Computer Measurement Group Conference.
- [6] Christina Catley; Dorina C. Petriu; Monique Frize, 2004. “Software Performance Engineering of a Web Service-Based Clinical Decision Support Infrastructure,” WOSP 04, ACM.
- [7] Ch Ram Mohan Reddy, D Evangelin Geetha, T V Suresh Kumar, K Rajani Kanth, 2015. “Activity Based Performance Prediction for Software Systems,” Technical Report, Department of Computer Applications, M S Ramaiah Institute of Technology, Bangalore.
- [8] David H. Crocker, 1982. “Standard for the Format of ARPA Internet Text Messages, RFC 822.”
- [9] E. Kageyama, C. Maziero, and A. Santin, 2008. “A Pull-based e-mail architecture”, SAC, pages 468–472, March 2008.

- [10] Hao Wang, Yizhu Tong, Hong Liu, 2006. Taoying Liu, "Application-aware Interface for SOAP Communication in Web Services", IEEE.
- [11] J R. Von Behren, S. Czerwinski, A D. Joseph, E A. Brewer, and J. Kubiawicz, 2000. "NinjaMail: the design of a high-performance clustered, distributed e-mail system," International Workshops on Parallel Processing, pp. 151-158.
- [12] Jonathan B. Postel, "Simple Mail Transfer Protocol, RFC 821", 1982.
- [13] K. Juse, S. Kounev, and A. Buchmann, 2003. "PetStore-WS: Measuring the Performance Implications of Web Services", 29th International Conference of the Computer Measurement Group (CMG) on Resource Management and Performance Evaluation of Enterprise Computing Systems.
- [14] L. Bertolotti and M. Calzarossa, 2001. "Models of Mail Server Workloads. Performance Evaluation", An International Journal, 46(2/3):65-76.
- [15] M. Tian, T. Voigt, T. Naumowicz, H. Ritter, and J. Schiller, 2003. "Performance Impact of Web Services on Internet Servers," International Conference on Parallel and Distributed Computing and Systems, Marina Del Rey, USA.
- [16] M. Calzarossa, 2003. "A Tool for Mail Servers Benchmarking," In G. Kotsis editor, Performance Evaluation - Stories and Perspectives, OCG Schriftenreihe, Pates 231-240, Austrian Computer Society.
- [17] Maria Carla Calzarossa, 2004. "Performance Evaluation of Mail Systems," Performance Tools and Applications to Networked Systems, pp 51-67, Springer.
- [18] Niko Thio and Shanika Karunasekera, 2005. "Client Profiling for QoS-Based Web Service Recommendation," Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05) IEEE.
- [19] Shah J Miah and John Gamlnack, 2008. "A Mashup Architecture for Web End-user Application Designs," 2008 Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008) IEEE.
- [20] Simon Woodman, Graham Morgan & Simon Parkin, 2003. "Portal Replication for Web Application Availability via SOAP," Proceedings of the Eighth IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, IEEE.
- [21] Seh-Joon Dokko, Sung-Hyun Yun, Tai-Yun Kim, 1997. "Development of Multimedia E-mail System Providing an Integrated Message View," IEEE.
- [22] Venu Datla and Katerina Goseva-Popstojanova, "Measurement-based Performance Analysis of E-commerce Applications with Web Services Components", 2005. Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE), IEEE.
- [23] W. Miles, 2002. "A high-availability high-performance E-Mail Cluster," In Proc., of the ACM SIGUCC Conf., pp 84-88.
- [24] Yan Li, Yao Liu, Liangjie Zhang, Ge Li, Bing Xie, Jiasun Sun, 2007. "An Exploratory Study of Web Services on the Internet," International Conference on Web Services (ICWS 2007), IEEE.
- [25] Ying-Wen Bai and Chung-Pian Chang, 2013. "Performance measurement and analysis of e-mail cluster systems by using three IP load-balancing technologies," 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE.