QUEUING MODEL APPROACH FOR ENHANCING QOS OF WEB SERVICES

[1]Ms. SUNDHARAM. R. [2]Mrs. LAKSHMI. M
[1] Research Scholar, Faculty of Computer Science, Sathyabama University
[2] Prof. & Head, Faculty of Computer Science, Sathyabama University
[1]Sun_moonr@yahoo.com, [2]facultyhead.cse@sathyabamauniversity.ac.in

ABSTRACT

In the distributed and parallel computing system, the availability of web service is increased by replicating the web services over physically distributed servers. For the large scale applications single server cannot fulfill the request of services, it leads to slow down the response time which is merely equivalent to unavailable service. In order to increase the availability of web service and minimize the response time, we have proposed Framework for Enhancing the Availability of Web Services. It uses Poisson distribution and queuing model $(M/M/C):(GD/\infty/\infty)$ to predict the future arrival rates and future response time respectively. Based on predicted results, server availability factor (SAF) and attributes value of Replication Matrix (RM), the decision is taken dynamically to replicate the specific service on appropriate host before the response time violates the SLA time. Hence the requested web service is made available to client service continuously with increased availability. A simulated test environment has been created to evaluate the performance of framework and test results are compared with existing models. The comparisons indicate that our framework provides greater performance. Thus, our framework maintains the response time of web service within expected SLA response time even during peak load and it significantly improves the availability of web service.

Keywords - *Web service, High-availability, Replication and Service Level Agreement.*

1. INTRODUCTION

World Wide Web is one of the most important medium to access and exchange the required information. It is an information repository, so various users use the services of the web to exchange business related information for professional, entertainment, personal and other purposes. " *A web service is a URI based accessible application that can be described, advertised, discovered and triggered to satisfy various needs like flight ticket booking* "[1] [2][3]. The QoS of composite services depends on

the orchestration of each atomic service and their availability. The quality of web services and consumer satisfactions are significantly achieved by the QoS attributes (availability, response time, reliability, security) of web service. Replication is a technique which provides the services redundantly with higher availability and reduced response time of web services by providing services redundantly [4] [5]. In this paper we proposed a Queuing Model Framework for Enhancing Availability of Web services (QMFEA-WS) which uses arrival rate and response time as two main key factors to decide the requirement of replication.

Our proposed framework is designed to keep monitoring and calculating the current arrival rate, current response time, SAF periodically. When current response time exceeds PSLA1 (50% of SLA time), it predicts the future arrival rate, future response time using poison and queuing model $(M/M/C):(GD/\infty/\infty)$ respectively

Once prediction is done, the predicted results, SAF, PSLA2 (80% of SLA time) has been applied in the algorithms to take decision dynamically to replicate the service on another server before the response time violates the SLA time. Hence the requested web service is made available to the clients continuously with reduced response time also increases the availability of web service. The rest of the paper is organized as follows: Section 2 summarizes related works in this area of research. Section 3 describes about the over view of proposed framework and its components. The prediction process and techniques are described in section 4. The evaluation and test results are presented in section 5. Finally the conclusion and the future work are summarized in section 6.

2. RELATED WORKS

Several architectures [6] [7] [8] and specifications have been developed on web services to improve its availability in the World Wide Web. For instance, the architecture defined by Abraham Thomas and et al. in paper [6] suggested an enterprise level gateway which keeps monitoring the services availability. It is performance overhead for the gateway also it impacts the processing of client requests and responses. A recent trend to overcome the issues in web service availability is centered on replication of service [5] [9]. The work proposed in paper [9], implemented with replication of services and multicasting, also discusses about different replication components and techniques such as passive replications, active replication and semi-active. The components of the framework have ability to enabling the services with persistent state. In Paper [10], the authors suggested a middleware which supports reliable web services based on active replication to ensure the consistency of the replicas. Sattanathan et al. in paper [11] proposed a solution on achieving availability of web services which substitute traditional replica web services with communities of similar web services. Also, the authors in paper [3] share the same domain of interest. Mathias Bjorkqvist et al. in paper [12] describe optimization of service replication in clouds through arrival rate based and response time based policy. Architecture proposed in paper [13] suggests an idea of multicasting the request to replicated services over different protocols and the service gate way (Enterprise Service Bus, ESB) that control the flow of requests and responses to and from the replicated services. This architecture increases the availability of web services significantly with the help of replication, multicasting and ESB. But from the load balance perspective multicasting and parallel invocation are ineffective [5], because it always increases the traffic and operating cost by propagating client requests to all the servers.

The Paper [14] suggests that Queuing Network (QN) models provide accurate analytic performance model of a system. In this paper QN model is used to estimate the performance index of a medical information system with different number of request. The QoS predictions on web services were investigated in multiple workflows. But most of them reported on the graph reduction technique [15] [16] [17]. The authors in [13], proposed a prediction framework for the enhancement of web service availability through adaptive replication using linear regression method to predict the future load based on data from previous days and accordingly service is replicated to serve for the day. However, this approach may not help in dynamically varying loads as the replication is not predicted based on the current load. The framework proposed in [18] for improving the availability of web services is done by predicting the future response time. The framework issues a replication decision on another server host once the predicted response time violates 85% of SLA time. In paper [19], the authors have suggested a framework for dynamic service placement and service replication for improving the availability of services using team formation algorithm. The framework concentrates on cost management, performance and availability in the event of service failover and does not consider the arrival rate and response time while replicating the web service. We are using main parameters namely request arrival rate, response time, SAF and replication matrix (RM) to take decision for replicating services on a server. If the predicted response time violates the SLAs time and rate of server utilization is high, the proposed prediction framework replicates the specific service on selected server. Also it provides better statistics of server level metrics that is used in our framework to improve the services of the system.

2.1 Statement of Existing Problem:

So far numerous service replication algorithms and different architectures were introduced to replicate the web services [13, 18, 19] for the enhancement of web service availability as discussed in section 2. However, all these existing research work were used for different evaluation techniques like load or response time or failover with different criteria for replication decision making; but none of the above mentioned frameworks as touched upon the prediction of both arrival rate and response time and various possible combinations of them to decide the requirement of replication. Since both are interrelated parameters, it is necessary to considering both to determine replication requirement. Therefore, in the QMFEA-WS, we have focused on predicting both Arrival Rate and Response Time and analyzing with the help of Replication Matrix to predict and replicate the service on-demand. The objective of QMFEA-WS is to achieve the availability in terms of response time like it should respond to the client requests successfully within the expected SLA time.

3. OVERVIEW OF PROPOSED FRAMORK

In the QMFEA-WS, the client requests are always passes through the service gateway (Figure 1) for processing which has three components namely Monitoring Service (MS), Intelligent Resource Control Manager (IRCM) and Load Balancing Service (LBS).

3.1 Monitoring Service (MS): The MS keep monitors the servers, services , the statuses of CPU utilization, memory utilization, number of request arrived , number of request processed, response time of each request and number of requests processed by web service (S_1) for a specific interval (T_r) (for instance every 60 seconds) and stores into database (MS/DB).

3.2 Intelligent Resource Control Manager (IRCM): The IRCM is a service component which manages and controls the web service response time within the expected SLA time. The 'IRCM reader' reads the MS/DB periodically at given time period 'T_r'. It calculates the average current response time (CR) of the period 'T_r' and compares CR with P-SLA1. If CR is greater than PSLA1, the 'IRCM Prediction Services' predicts the future arrival rate, future response time using Poisson and Queuing model respectively. The 'IRCM decider' analyzes the previous interval's metrics and current interval's metrics and applies in algorithms (ALG-S, ALG-U) to decide the attributes (severity and urgency of replica requirement) . Once the RM attributes (severity and urgency) determined,' IRCM decider' applies the attributes in the replication matrix and ALG-R and dynamically replicates the service on another host at appropriate time and notify to load balancing service and monitoring service to send the request and monitor the new replica respectively.

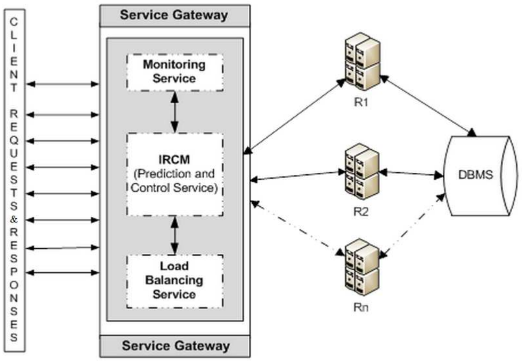


Figure 1: Queuing Model Framework for Enhancing Availability of Web services

3.3 Load Balancing Service (LBS): LBS receive requests from clients and distribute them to the pool of active replicated web services using round-robin technique. This replicated web services process the requests and sends responses back to the clients.

4. PROCESSES AND ALGORITHMS

4.1 Prediction Process:

The current arrival rate (λ_c) is the mean number of requests arrived per unit time *(T_r)*. The current response rate (μ_c) is number of request processed per unit time *(T_r)*. At the end of interval *(T_r)*, the IRCM reads the MS data base and observes rate of arrival (λ_c) , current response rate (μ_c) and average response time (T_c) of web services. The IRCM compares the current response time (T_c) with PSLA1, if T_c is greater than PSLA1, IRCM starts the prediction process to predict the future arrival rate (λ_f) and future response time (T_f) using Poisson and queuing model $(M/M/C):(GD/\infty/\infty)$. At the end of every interval, the metrics are moved from current to respective previous variables for future reference. The IRCM predicts the future arrival rate 'λ_f' using the Poisson distribution eqn. (1).

$$P(X=n)=\frac{\lambda_c{}^n e^{-\lambda_c}}{n!} >= 0.99 \qquad (1)$$

The value of future arrival rate (λ_f) is predicted at 99% probability confidence. When the system running with multiple replica servers, the Current Arrival rate can be arrived through equation (2) as below.

$$\lambda_c = CA = \sum_{R=1}^{n} \frac{\lambda_c{}^R}{R} \qquad (2)$$

Response time is the time that taken by a service to process the request completely. In other words the response time of a request can be defined as the time difference between the time of a request submission and the time that the response is received [20]. The IRCM uses QN M/M/C formula to predict the future response time (T_f) using equation (4) and equation (5). According to queuing model $(M/M/C):(GD/\infty/\infty)$ equation, the server utilization (ρ) is obtained from below equation (3).

$$\rho = \frac{\lambda_c}{\mu_c} \qquad (3)$$

$$T = \frac{1}{\lambda}\left[\frac{\rho^{R+1}}{(R-1)!(R-\rho)^2}\left\{\sum_{n=0}^{(R-1)}\frac{\rho}{n!}+\frac{\rho}{R!(1-\frac{\rho}{R})}\right\}^{-1}+\frac{1}{\mu}\right] \qquad (4)$$

$$FR = T_f = \frac{1}{\lambda}\left[\frac{\rho^{R+1}}{(R-1)!(R-\rho)^2}\left\{\sum_{n=0}^{(R-1)}\frac{\rho}{n!}+\frac{\rho}{R!(1-\frac{\rho}{R})}\right\}^{-1}+\frac{1}{\mu}\right] \qquad (5)$$

The future response rate *(μ_f)*can be derived using T_f in equation (6).

$$\mu_f = \frac{t}{T_f} \qquad (6)$$

When μ_f increases T_f will be reduced. So one of the way to increase the μ_f is increasing the number of server during the peak load.

Nomenclatures:

μ	= Average response rate
λ	= Mean No. of requests arrived/Tr
λ_f	= Future Arrival Rate (FA)
λ_c	= Current Arrival Rate (CA) per T_r
λ_c^R	= Current arrival rate of server's'
μ_c	= Average current response rate
e	= Euler's number.
n	= Poisson variable
R	= number of servers
ρ	= the server utilization
T_c	= Average current response time(CR)
T_f	=Future Response_Time (FR)
T_r	=Statistics_read_interval
R_n	= Represents 'n'th servers
S_n	= Web Service
λ_p	= Previous_Arrival_Rate (PA)
T_p	= Previous_ResponseTime (PR)
CPR	= CR_PR_Ratio
SRA	= Server Resource availability
SCA	= Server CPU availability
SRU	= Server resource utilization
SCU	= Server CPU utilization
SAF	= Server Availability Factor

4.2 Determining SAF

As the server resource utilization is playing major role in response time of request, it has been considered as important parameter in deciding replica. The SRU, SCU for server is obtained from MS.

$$SRA = 1 - SRU / 100 \qquad (7)$$

$$SCA = 1 - SCU / 100 \qquad (8)$$

SAF is being arrived through the below equation (9).

$$SAF = \frac{SRA \wedge 2 * \sqrt{SCA}}{1000} \qquad (9)$$

Several trials have been conducted to derive this relationship between resource availability, CPU availability and SAF is considered as one of the key decision factor used to arrive the attributes of replication matrix. In a queuing model $(M/M/C):(GD/\infty/\infty)$, a system will be considered as stable system when the server utilization should be less than 1 as per eqn. (10).

$$\rho = \sum_{R=1}^{n}\frac{\lambda_R}{\mu_R} < 1 \qquad (10)$$

The equation (10) reveals that μ is inversely proportional to server utilization which depends on SCU and SRU. In other words μ is directly proportional to SRA and SCA. So the above system native properties are taken into account while deciding the replica requirement. When resource availability is high, the response rate will be high. When native properties of server are low, then response rate will be low and response time will be high, which indirectly means that the service availability is reduced. The Figure 2 shows an interactive communication flow of QMFEA-WS.

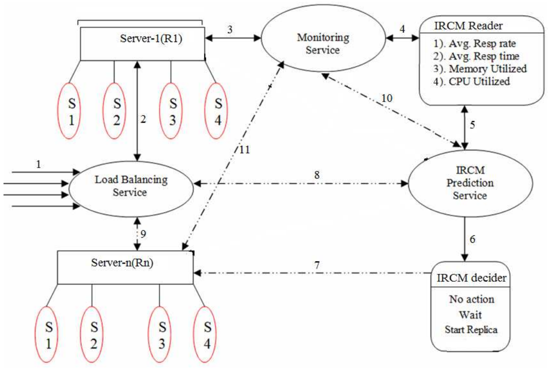


Figure 2: Interactive flow of QMFEA-WS

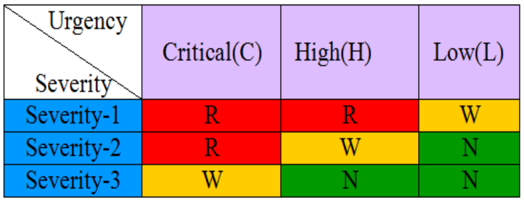
4.3 Determining the Attributes of RM

Severity: Severity is an attribute of replication matrix that defines performance behavior of the server with respect to how fast the requests are being responded. Severity has been determined into three category namely Severity 1, Severity 2 and Severity 3 using algorithm 'ALG-S'.

Urgency: The Urgency is another attribute of RM that defines the 'time' that how soon the replica is required to maintain the response time within SLA time. The Urgency has been determined into three categories namely Critical, High and Low. The Urgency is determined based on the validation and comparison of several directly or indirectly impacting interrelated parameters which are described in ALG-U

4.4 Replication Matrix

Replication Matrix (RM) attributes are defined by algorithm 'ALG-R' which helps the IRCM to take decision on replication. RM formed as shown in Figure 3 to identify the requirement of replication and the time when replication is required. Once the replication matrix values are populated, it is used by the IRCM to decide the requirement of replication as follows:



Urgency / Severity	Critical(C)	High(H)	Low(L)
Severity-1	R	R	W
Severity-2	R	W	N
Severity-3	W	N	N

Figure 3: Replication Matrix

1). When Severity is 1 and Urgency is 'C' or 'H', IRCM sets the RM value as 'Y' which denotes replication is required. Thus IRCM issues the command to replicate the specific service on appropriate host and sends request to add the new replica in active server pool. But when Urgency is 'L', IRCM will not take any immediate action; it decides to 'wait' for next interval to decide about the replication requirement.

2). When Severity is 2 and Urgency is 'C', IRCM sets the RM value as 'Y' which denotes replication is required. Thus IRCM issues the command to replicate the specific service on selected host based on the load of server. But if Urgency is 'H', IRCM sets the RM value as 'W' which denotes to 'wait' for next interval to decide about the replication. If Urgency is 'L', IRCM sets the RM value as 'N' which denotes to 'no action is to be taken', hence it continues for next interval analysis.

3). When Severity is 3, if Urgency is 'C', IRCM sets the RM value as 'W' which denotes to 'wait' for next interval to decide about the replication. If Urgency is 'H' or 'L', IRCM sets the RM value as 'N' which denotes 'no action is to be taken', hence it continues for next interval analysis.

ALG-S:

```
/ **** Determine Severity ****/
if CR > PSLA2 {
        set severity = 1 }
if (PSLA1< CR < PSLA2) {
        set severity = 2 }
if CR < PSLA1 {
          set severity = 3; }
If (severity = 1 || severity = 2)
      {
        Function1 : future_ArrivalRate();
        Function2 : Future_Resp_time();
        Function3 : Find_SAF();
      }
End-if.
```

ALG-U:

```
/ **** Determine Urgency ****/
 If [CR >PSLA2 &&
     ( CPR >1.3 ||
       FR >90% SLA) &&
       SAF < 0.65]
          {
             Set Urgency=C;
          }
  else if [(CR between PSLA1 && PSLA2) &&
             CPR Between 1.0 & 1.3 ||
             FR >PSLA2) &&
              SAF < 0.5]
                {
                   set Urgency=H;
                }
  else {
           set Urgency=L
          }
  End-if.
  if [severity = 2 && CPR >1.3 &&
            ( SAF < 0.30 ) ]
            {
              set Urgency = C;
            }
  else if [severity = 2 && (CPR >1.0 ||
             (FR Between PSLA2 & 90% of SLA) &&
              (SAF between 0.3 to 0.5)) ]
             {
              set Urgency=H;
             }
  else       {
            set Urgency=L;
             }
```

ALG-R:

```
/ *** Determine Replica requirement: ***/
  If [(severity = 1) &&
      (Urgency =C ||
       Urgency =H)]
        {
           Replication_Required=R;
        }
  else-If ((Severity = 2) && (Urgency =C))
         {
           Replication_Required=R
         }
  else if ((Severity = 2) && (Urgency =H))
         {
           Replication_Required=W
         }
```

```
else if [(Severity = 1) &&
        (Urgency =L)]
        {
          Replication_Required=W
        }
else {
        Replication_Required=N
        }
end-if.
```

5. EVALUATIONS AND TEST RESULTS

5.1 Environment

This section explains the test environment to evaluate the performance of QMFEA and analyze the results. The test environment was setup with virtual machines on Hyper-V with the Windows 2005 server platform with 4GB RAM and MS-SQL Server 2005. The software tools ‘Applications Manager’ from Manage Engine, ‘JMeter’ from Apache, and ‘ACE (Application Control Engine)’ from Cisco were used for monitoring, testing, and load-balancing purposes, respectively. Tomcat server and Eclipse were used for application development and implementation. We employed a web service, namely the balance enquiry service (C_1) on each of the replication servers $R_1…R_n$ which were connected through a LAN (local area network) with a bandwidth of 100 mbps. This web service retrieves the balance details of the account. The SLA time defined for C_1 is 1300 ms. To test various scenarios, concurrent HTTP requests were initiated from seven client machines with the frequency ranging from 1000 to 20,000 requests/minute using JMeter with a benchmarking tool. The number of concurrent HTTP requests is represented with threads every 60 seconds. For instance, to submit 1000 requests/minute, we initiate 170 threads every 10 seconds, meaning that JMeter takes approximately 10 seconds to initiate all 170 threads. This is done to avoid congestion in the server caused by a large volume of threads at the beginning of the test.

5.2 Experiments

The aim of these test scenarios is to evaluate the ability of our proposed framework to automatically enhance web service availability through replication with minimal intervention from the system administrator. Table 1 provides the overview of the test scenarios we conduct in this paper.

Table 1: Overview of experimental scenarios

Test#	Scenario
1	Testing of QMFEA at excessive load to find the reliability of the framework for timely replication.
2	Testing of QMFEA at continuous over load to find the sustainability of the framework for multiple replications at timely manner.
3	Compare the QMFEA with existing models for response time
4	Compare the QMFEA with existing models for time of replication.

5.2.1 Reliability test: To understand the performance of our framework we tested with arrival rate @ 1250 req/min to 10000 req/min with two replicas. The test results are collected and the graphical representation is shown in Figure 4. The CPU utilization and memory utilization has been collected from MS as shown in Figure 5 and Figure 6 and for the current response time (CR = 1175ms) and CA 7000/min, IRCM calculates the FA, FR and SAF using eqn. 1 thru 10. It determines the severity and urgency using ALG-S, ALG-U. The RM attribute are

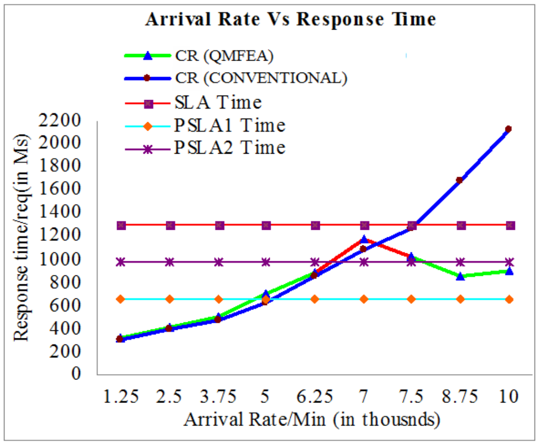


Figure 4: Arrival Rate vs. response time

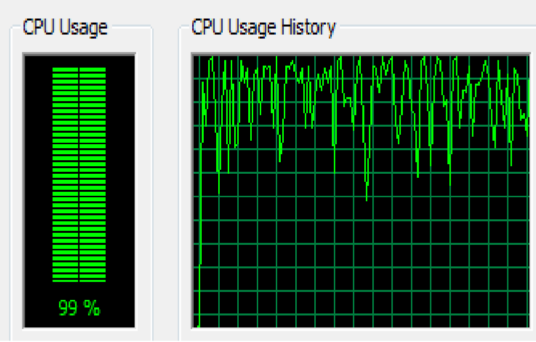


Figure 5: CPU Utilization Snap shot at peak load

applied in ALG-R and determined that replication_required as 'R'. It denotes that replication is required immediately to process the request with in the SLA time. Hence IRCM replicates a specific service on appropriate host and it sends request to LBS to add the new replica in active server pool. Hence LBS distribute the load on new replica service. Thus CR is reduced as shown in Figure 4. Similarly, the test has been conducted with arrival rates @7500 req/min, @8750 req/min and @10000 req/min and the test results are shown in Figure 4.

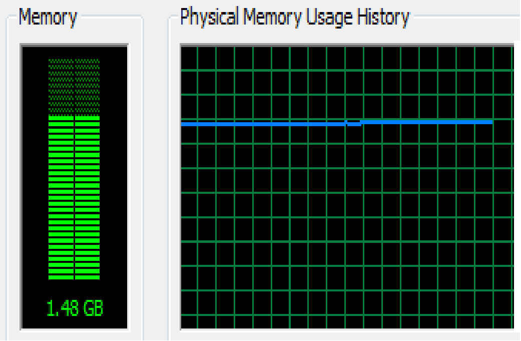


Figure 6: Memory Utilization Snap shot at peak load

5.2.2 Sustainability test: The aim of this test is to understand the behavior of the QMFEA when system is subjected to a varying load from 7,000 to 23,000 req/min by increasing the load at every minute. λ_c and T_c are collected from the MS and λ_f and T_f are calculated from Eq. (1) to Eq. (10) and graphical representation is shown in Figure 7. It proves that when framework subject to continuous over load, it replicates new replica at appropriate time. Thus it maintains the response time within the agreed SLA time.

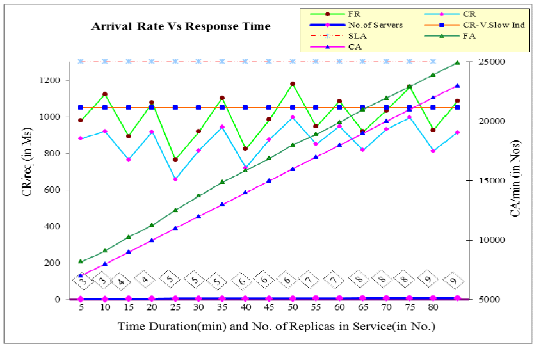


Figure 7: Arrival rate vs. response time

5.2.3 Comparison test with respect to CR

This test is conducted to compare the response time delivered by each system in processing the same number of requests. Here, we compared QMFEA-WS, LRM and HLSEM for the various loads and collected the response times and shown in Figure 8. It shows the comparison of the response times delivered by each system in processing the same load ranging from 5000 requests/min to 35,000 requests/min. The Figure 8 indicates that QMFEA maintains a slightly higher response time than the other two models because the QMFEA-WS utilizes less number of replicas compared with the other two methods (Figure 9), but it maintains the response time is always within the SLA. Hence, our framework provides near optimal replication decisions, thereby reducing the operational cost while still obeying the SLA.

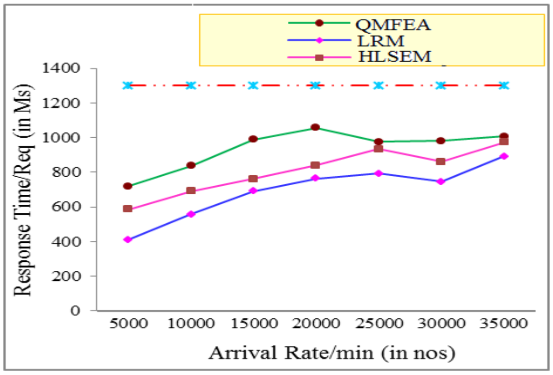


Figure 8: Varying Load response time comparison

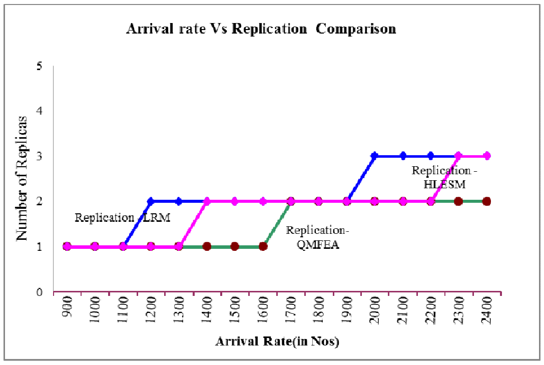


Figure 9: Varying Load replication comparison

Comparison test with respect to number of replica

In order to compare the efficiency of QMFEA-WS, we analyzed and compare with Linear Regression Model (LRM) suggested in paper [13] and Halts Linear Exponential Smoothing Model (HLESM) suggested in paper [18] for the number of replicated services provided by each models to process the same load. The LRM replicates services when Load >75% or predicting the 16th day load using past two weeks history. On the other hand the HLESM find the response time using QN model and predict response time using HLESM and replicates services when predicted response time greater than 90% of SLA. These models replications techniques are computed for different load varied from 900req/min to 2400req/min. The results are compared with QMFEA-WS and shown in Figure 9. The result shows that the LRM [13] replicates early than it actually required because it just uses the server load as replication criteria for dynamic replication and the HLESM [18] replicates the service at slightly more appropriate time than [13]. The results show that, QMFEA-WS predicts the replication at most appropriate time compare to other existing replication models [13], [18].

6. CONCLUSION AND FUTURE WORK

In this paper we have proposed an adaptive prediction framework for improving the availability of web services, which uses the Poisson , QN models to predict the future arrival rate and future response time of service requests. Based on predicted results, server availability factor (SAF) and attribute values of Replication Matrix (RM), the decision was taken dynamically to replicate the particular service on-demand on appropriate host before the response time violates the SLA time. The advantage of our proposed framework is it provides proactive, on-demand, dynamic service replication model to respond the requests at desired response time limit. The test result and its performance are compared with existing models and results indicate that our framework provides greater performance. Thus, our framework maintains the response time of web services within expected SLA response time even during peak load and it significantly improves the availability of web services. Our future works will be in the direction of implementing our framework on the sensor web and Fuzzy logic to analyze vague situations in arrival rate and response time for enhancing the availability of web services for large scale applications.

REFERENCES:

[1]. B. Benaatallah, Q. Z. Sheng & M. Dumas, 2003. "The self-Serve Environment for Web Services Composition". *IEEE Internet Computing* 1: 40-48.

[2]. M. P. Papazoglou & D. Georgakopoulos, 2003. "Introduction: Communications of the ACM",*ACM* 46 (10) : 24-28.

[3]. Sattanathan Subramanian, 2009. " Highly - Available Web Service Community". *6th International conference on Information Technology, New Generation, IEEE.*

[4]. Marco Conti, Mohan Kumar, Sajal K. Das & Behrooz A. Shirazi, 2002. "Quality of Service Issues in Internet Web Services". *IEEE Transaction on Computers* 51: 6.

[5]. Nabor C. Mendonca, Jose Airton F. Silva, Ricardo O. Anido, 2008. " Client side selection of replicated web services: An empirical assessment". *Journal of Systems and Software, Elsevier Science Inc.* 81(8): 1346-1363.

[6]. Subil Abraham, Mathews Thomas & Johnson Thomas, 2005. "Enhancing Web Services Availability". *Proceedings of 26th IEEE International Conference on Software Engineering.*

[7]. K Birman, R Van Renesse & W Vogels, 2004. "Adding High Availability and Autonomic Behavior to Web services". *Proceedings of the 26th International Conference on Software Engineering, IEEE.*

[8].Ms. R Sundharam, M. Lakshmi,D. Abarajithan, 2010. "Enhancing the High Availability of Web Services for Mission Critical Applications". *International Conference on Trendz in Information Sciences and Computing, IEEE, TISC* 2010.

[9]. Jorge Salas, Francisco Perez - Sorrosal, Marta Patino - Martinez & Ricardo Jimenez - Peris, 2006. "WS - Replication: A Framework for Highly Available Web Services". *International World Wide Web Conference Committee , ACM* , Edinburgh, Scotland.

[10]. Xinfeng Ye & Yillin Shen, 2005. "A Middleware for Replicated Web Services". *International Conference on Web Services.*

[11]. Maamar Z, Sheng QZ, Benslimane D, 2008. Sustaining web services high- availability using communities". *The Third International Conference on Availability .*

[12]. Mathias Bjorlqvist & Lydia Y. Chen, 2011. Optimizing service replication in Clouds. Proceedings of the IEEE Winter Simulation Conference.

[13]. Marwa F Mohamed, Hany F EIYamany & Hamed M Nassaer, 2013. A study of an adaptive replication framework for orchestrated composite web services. Springer Plus , Computer Science.

[14]. Bai, Y - W. & C – Y. Cheng, 2004. The performance Estimation by Queuing Network Models for a Web-based Medical Information System. *In Proceedings of the 17th IEEE Symposium on Computer –Based Medical Systems. IEEE Computer Society*.

[15]. J. Cardoso, A. Sheth, J. Miller, J. Amold & K. Kochut, 2004. “Quality of service for workflows and web service process”. *Web Semantics: Science, Services and Agents on the World Wide Web* 3:281-308.

[16]. M. Jaeger, G. Rojec - Goldmann & G. Muhl, 2004. “Qos aggregation for web service Composition using workflow patterns”. *Enterprise Distributed Object Computing Conference, IEEE*.

[17]. Alfredo Goldman & Yanik Ngoko, 2012. “On Graph Reduction for QoS Prediction of Very Large Web Service Compositions”. *IEEE Ninth International Conference on Service Computing*.

[18]. Mohamed-K Hussein & Mohamed-H Mousa 2012. “A Framework for adaptive QoS of Web Services using Replication”. *International Journal of Computer Science and Communication Networks 288-294*.

[19]. Boon–Yaik Ooi, Huah-Yong Chan, Yu-N. Cheah, 2012. “Dynamic service placement and replication framework to enhance service availability using team formation algorithm”. *The Journal of Systems and Software*.

[20]. Hwangm Haojun Wang, Jian Tang & Jaideep Srivastava, 2007. “A probabilistic approach to modeling and estimating the QoS of web – services -based workflows”. *International Journal of Information Sciences* 1:5484-5503.