IJESMR
International Journal OF Engineering Sciences & Management Research

A FUZZY BASED ADAPTIVE PREDICTION FRAMEWORK FOR ENHANCING THE AVAILABILITY OF WEB SERVICES FORINTERNET APPLICATIONS

Ms. R. Sundharam*[1]& M.Lakshmi[2]
*[1]Research Scholar, Faculty of Computer Science, Sathyabama University,Chennai-600119, India
[2]Prof. &Head, Faculty of Computer Science, Sathyabama University, Chennai-600 119, India

Keywords: Web Service (ws); Fuzzy Logic(FL); Replication; Service Level Agreement(SLA).

ABSTRACT
The internet based applications are primarily supported by web services(*ws)*. The number of requests received on *ws* are varying dynamically from time to time which leads to slow down in response time during peak load periods. In order to overcome this problem, the availability of web service is increased by replicating the web services over physically distributed servers. In this paper we have proposed an adaptive prediction framework which uses Poisson and exponential distribution models to meet the Quality of Service(Qos) attributes (availability, response time) of web services under the high number of service request via service replication. The Poisson and exponential models are used to find the probability of request arrival rates, response time of requested services.It alsouses FL for efficient and Resource Control Algorithm (RCA) for decision making to determine the replication requirement.Simulated test environments have been created to evaluate the performance of our framework and test results are compared with existing models. The results confirms that our proposed framework maintains the response time of *ws* within expected SLA (Service Level Agreement) response time even during peak load and it significantly improves theavailability of *ws* with minimal intervention of system administrator.

INTRODUCTION
Web service is an emerging technology of well-defined software components which provides business applications over the web and it increases efficiency in service providing, exchanging and aggregating the data in the distributed environment[1].Since the rate of service requests received from the clientsvaries disproportionately on these popular *ws*, the Self-Management Automation (SMA) system becomes indispensable to monitor resource utilization constantly to scale up/down the resource capacity. This technique was introduced by IBM [2] to reduce the administrative tasks in the distributed system. So far, several attempts were made to develop SMA[3][4][5][6] in order to reduce administrator intervention in distributed resource management. The current trend in enhancing *ws* availability is centered on replication of *ws*[7][8] which helps to maintain the *ws* availability even during peak load periods.

In this paper we propose Fuzzy Logic based replication SMA framework "Fuzzy basedadaptive prediction framework for enhancing the availability of web services for internet applications" (FAPFEA) which predicts the future arrival rate and response time using Poisson distribution (PD) and Exponential distribution (ED) respectively. The predicted values are fed into fuzzy inference system to evaluate against the various fuzzy rules to decide whether replication requirement exists. Once the replication requirement decision is made, Resource Control Service (RCS) uses Resource Control Algorithm (RCA) to further evaluate the 'time' for which the replication is required to dynamically replicate the *ws* on another host before the response time violates the SLA. This framework is also designed to provide solution for situations resulting in *ws* unavailability due to service being stopped orservice/server hangs. Also, in case where response time of any individual server is low and is approaching the threshold limit, it replicates the *ws* on another server and stops the replica which has lower performance.

This paper has been organized in the following manner: Section 2 summarizes literature review in this area of research. Section 3 describes the over view of proposed adaptive replication framework and its components. The section 4 describes the prediction processes, techniques and algorithms used. The section 5 describes simulated test environment, different scenarios to be tested and test results. Finally the conclusion and the future work are summarized in section 6.

LITREATURE REVIEW
Several architectures and specifications [10][11][12][13] have been developed for *ws* to enhance its availability. A recent trend in *ws* availability is centered on replication of *ws*[15][16][17][18]. The author In a paper [15] proposed a framework which discusses as active, passive, semi-

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

Nomenclature

R_n = *Represents 'n'th servers*
SC_n=*Represents 'n'thweb services*
T_i = *Statistics_Read_Intrverval*
Y = *Replication required is 'Yes'*
N = *Replication required is 'No'*
S = *'Stop' existing replica*
P_j = *Measured arrival rate.*
$P_j^{\wedge}$ =*Predicted arrival rate.*
m^i = *Membership degree of rule i*
w = *Weight age of membership*
x = *Random Variable*
e = *Euler's number.*
T_c = *The average current response time*
T_f = *Future Response Time (FR)*
n_q= *No. of request*
n_s= *No. of server*
T_c^s = *current response time of server's'*
T_h = *Health status check time interval*

Greek Symbols
μ = *Average response rate*
μ_c = *Average current response rate*
γ_1 = *Fuzzy variable of replication rule1*
γ_2 =*Fuzzy variable of replication rule2*
γ_3 =*Fuzzy variable of replication rule3*
λ = *Mean No. of requests arrived/Ti*
λ_c = *Current Arrival Rate(CA) per Ti*
λ_f = *Future Arrival Rate*
σ_u = *Probability confidence for FA1*
σ_v = *Probability confidence for FA2*
θ_J = *Probability confidence of FR1*
θ_k = *Probability confidence of FR2*
λ_c^s = *Current arrival rate of server's'*
ω = *Value of 50% of arrival rate*
ψ = *Value of 50% of SLA response time*
δ = *Defuzzyfied crisp output*

activereplications and enable the different replication components and techniques such services with persistent state. In the paper [16] author proposed architectureto enhance the availability of *ws* with the help of enterprise service bus, replication of service and multicasting. However, from the load balance perspective, multicasting and parallel invocation is ineffective[17], because it always increases the traffic and operating cost by propagating the same client requests to all the servers. The author in [19] proposed an adaptive prediction framework for the enhancement of *ws* availability using replication. In this research work author suggested linear regression method to predict the future load based on data from past 15 days and accordingly *ws* 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. In paper [20] authors proposed a framework for improving the availability of *ws* by predicting the future response time.The framework issues a replication decision on another server host once the predicted response time violates 90% of SLA time. In paper [21] authors have suggested a framework for dynamic placement of service 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 *ws* failover and does not consider the arrival rate and response time while replicating *ws*

Statement Of Problem

 ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences & Management Research

So far numerous service replication algorithms and different architectures were introduced to replicate the *ws* for the enhancement of *ws* availability as discussed in section2. However, all these existing research work were used for different evaluation techniques, with different criteria for replication decision making; but none of the above mentioned frameworks as touched upon the use of prediction of arrival rate, response time and considering various possible combinations of them to decide the requirement of replication. Consideration of both arrival rate and response time becomes necessary since both of them are interrelated parameters (HwangmHaojun Wang *et al.*, 2007; Marco Conti *et al.*,2002) which help determine the availability of *ws* and need for server replication. Therefore, in the FAPFEA, we have focused on predicting arrival rate, response time, predict the requirement of replication and reclaim the resources precisely. The objective of FAPFEA is to achieve the following aspects of *ws* availability:

1. Availability in terms of Response time – The ws should respond to the client requests successfully within the expected SLA time (Qualitative).
2. Availability in terms of ws up time - The ws should be available for processing the requests at any time which will provide uninterrupted availability of ws (Qualitative).
3. Availability in terms of Capability - The framework should have the capability to process any amount of stress load on ws as the load isdynamically changing in current internet business world (Quantitative).

AN OVERVIEW OF PREDICTION FRAMEWORK

In **the *FAPFEA*,** any service request from client is always passed through the service gateway (Fig.1). The components of the service gateway process the request and the response is sent back to the client. The service gateway has three components namely Monitoring Service (MS), Intelligent Resource Control Manager (IRCM) and Load Balancing Service (LBS).

Monitoring Service(Ms)

The MS keep monitors the statuses of *ws* (SC_1, $SC_2 \ldots SC_n$), servers(R_1, $R_2 \ldots R_n$)and creates metrics for no. of request arrived, response time of each request andno. of requests processed by *ws* (SC_x)during a specific interval.

Intelligent Resource Control Manager (Ircm)

The IRCM is the core component of the framework which has three sub components namely: a). Future Prediction Service (FPS), b). Fuzzy Inference Service (FIS) andc).Resource Control Service (RCS)**.** The processes and functions used by IRCM for the replication cycle have been illustrated in Fig. 2.

Future Prediction Service (FPS)

FPS collects the no. of requests arrived and requestsprocessed by *ws* from the MS for every defined time interval (for example every 60 seconds). The time interval is defined as *Statistics_Read_Intrverval* (T_i) which can be parameterized by the administrator and it usually will be in multiples of the pole interval. With the help of current average arrival rate and current average response time, FPS predicts the future arrival rate and future response time using PD and ED respectively.

Fuzzy Inference System

FIS reads the predicted values from the FPS and analyzes using fuzzy logic to evaluate against various rules (Table 3) to determine the requirement for replication based on replication matrix (Table 4). The three possible fuzzy decisions are:1. Replication required -‘*R*’, 2. Replication NOT required -‘*N*’ and 3. Stop existing replica-‘*S*’.

ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences & Management Research

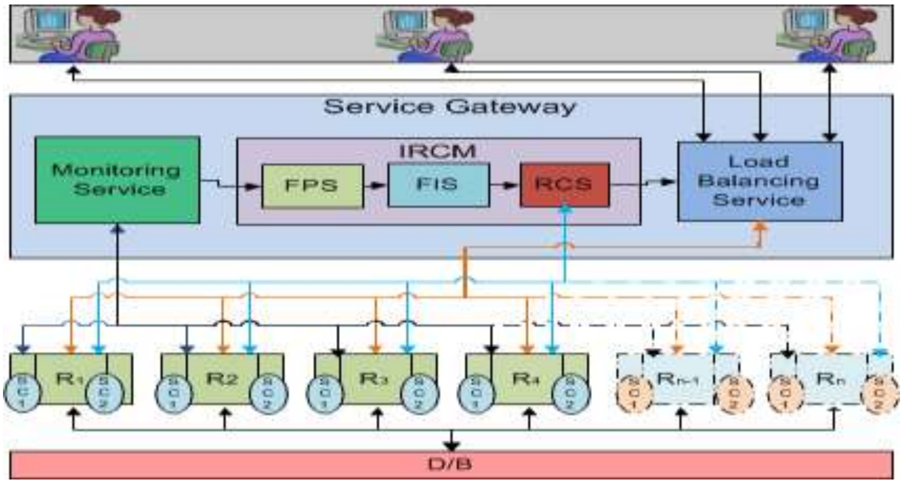


Fig. 1Frameworkof FAPFEA

Resource Control Service (RCS):
The RCS incorporates the FIS decision into the Resource Control Algorithm (RCA) to dynamically replicate new ws on another host or reclaim the existing replica host. This service has been explained in detail in section 4.4.

Load balancing service (lbs)
LBS receive requests from clients and distribute them to the pool of active replicated*ws* using round-robin technique. These replicated *ws* process the requests and sends responses back to the clients.

PREDICTION PROCESSES AND ALGORITHM

Arrival Rate
TheArrival rate (λ) is the mean number of requests arrived per unit time (T_i). The average current arrival rate of an individual server (λ_c^s) is calculated using the following Eq.(1).

$$\lambda_c^s = \frac{n_q^s}{T_i} \qquad (1)$$

When the system is running with multiple replica servers, the current arrival rate(CA) can be arrived throughEq.(2).

$$CA = \lambda_c = \frac{\sum_{s=1}^{n} \lambda_c^s}{n_s} \quad (2)$$

The FPS predicts the future arrival rate called FA('λ_f') using PD Eq. (3) (Jerry Banks 2001,page 168).

$$P(X = x) = \frac{\lambda_c^{\,x} \otimes e^{-\lambda_c}}{x!} \qquad (3)$$

In order to find the future arrival rate, we used the meanof probability between two probability confidencesσ_u andσ_v; hence σ_u + $\sigma_v \rightarrow \sigma_{u+v} \rightarrow \sigma_{100\%}$
First, future arrival rate(FA$_1$) is predicted at probability confidence $\sigma_u$usingEq. (4).

$$FA_1 = \lambda_{f1} = f(x;\sigma_u) = \frac{e^{-\lambda_c} \otimes \lambda_c^{\,x}}{x!} = \sigma_u \quad (4)$$

Then, future arrival rate(FA$_2$) is predicted at probability confidence σ_v using Eq.(5).

ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences & Management Research

$$FA_2 = \lambda_{f2} = f(x;\sigma_v) = \frac{e^{-\lambda_c} \otimes \lambda_c{}^x}{x!} = \sigma_v \quad (5)$$

the total future arrival rate is calculated using Eq. (6)[24].

$$FA = \{[FA_1 \otimes \sigma_u] \oplus [FA_2 \otimes \sigma_v]\} \quad (6)$$

Response Time

Response time is the time taken by a *ws* to process one request completely. It can also be defined as the time difference between the time that a request is submitted and the time that the response is received [23]. The average current response time (T_c^s)of an individual server's'is calculated using Eq.(7). When systemhas multiple servers, the Tcis calculated using Eq.(8). Response rate μ_c iscalculated using Eq. (9).

$$T_c^s = \frac{\sum_{q=1}^{n} T_c^q}{n_q} \quad (7)$$

$$CR = T_c = \frac{\sum_{s=1}^{n} T_c^s}{n_s} \quad (8)$$

Current Response Rate/min $\mu_c = \dfrac{T_i \otimes 60}{T_c}$ (9)

Where CR= current response time.

The μ_c is used by FPS to predict the future response time called FR (T_f)using the EDEq. (10)[24] as follows.

$$P(X = x) = T_f = \left\{\mu_c \otimes e^{-\mu_c \otimes x}\right\} \quad (10)$$

We used the mean of probability between two different probability confidence $\theta_J$and $\theta_K$to find the future response time, Hence $\theta_J + \theta_K \rightarrow \theta_{J+K} \rightarrow \theta_{100\%}$. First, the future response time (FR_1) is predicted at probability confidence $\theta_J$using the following Eq. (11).

$$FR_1 = f(x;\theta_J) = \left\{\mu_c \otimes e^{-\mu_c \otimes i}\right\} = \theta_J \quad (11)$$

Then, the future response time(FR_2) is predicted at probability confidence $\theta_K$using the Eq. (12).

$$FR_2 = f(x;\theta_k) = \left\{\mu_c \otimes e^{-\mu_c \otimes i}\right\} = \theta_k \quad (12)$$

From the Eq. (11) and Eq. (12), the average predicted future response time is calculated using the Eq. (13)[24].

$$FR = (T_f) = \{[FR_1 \otimes \theta_J] \oplus [FR_2 \otimes \theta_k]\} \quad (13)$$

Fuzzy Algorithm

Once the arrival rate and response time have been calculated, the next step involves mapping of these non-linear input data set (arrival rate and response time) to a scalar output data (decision on replication is required or not) which is done using fuzzy logic system. The four key processes of fuzzy logic system [25] are: 1). Fuzzifier, 2). Fuzzy Rules, 3). Fuzzy Inference Engine and 4). Defuzzifier. The processes and functions used by IRCM for replication cycle have been illustrated in Fig. 2.

***Defining linguistic variables**:*

In the fuzzy system, linguistic variable is concept that used to represent the input or output variable which carries linguistic label [26],generally it is decayed into a set of linguistic term. The linguistic variables of FA and FR have been defined in Table1, Table2 respectively

ISSN 2349-6193
Impact Factor: 2.805
IJESMR
International Journal OF Engineering Sciences & Management Research

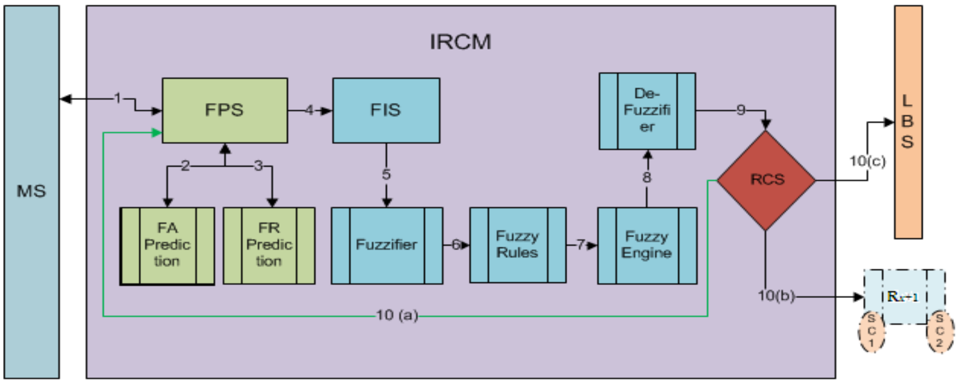


Fig. 2 Replication cycle.

Determining membership functions and converting to fuzzy values:

Our framework uses the Triangular method [27] to determine the range of fuzzy variables. The linguistic values of arrival rate are converted into triangular fuzzy numbers such as ω-4000, ω-3000, ω-2000, ω, ω+2000, ω+3000, ω+4000 (here 'ω' is 50% of arrival rate ie.5000) as shown in Fig.3. Correspondingly, the linguistic values of response time are converted into triangular fuzzy number such as ψ – 600, ψ – 400, ψ – 200, ψ, ψ + 200, ψ+400,

ψ+600 (where 'ψ' is 50% of SLA response time i.e. 650) [28] as shown in Fig. 4.

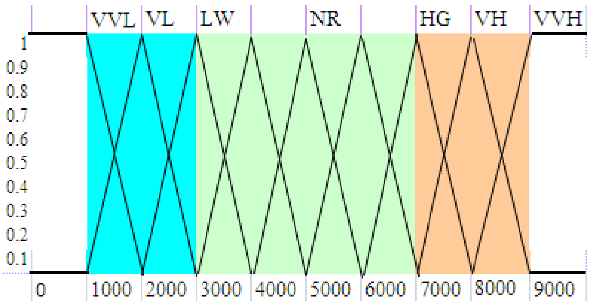


Fig. 3 Fuzzy Sets (α) of Arrival Rate

Table 1: Linguistic Variables for FA

Sl. No	Crisp Input Range	Linguistic Variables
1	0 - 2000	Very Very Low(VVL)
2	1001 - 3000	Very Low(VL)
3	2001 - 4000	Low(LW)
4	3001 - 7000	Normal(NR)
5	6001 - 8000	High(HG)
6	7001 - 9000	Very High(VH)
7	8001 - 10000	Very Very High(VVH)

Fuzzy rule base and combining rules result:

The fuzzy rules functions on a series of 'if – then - else' statements. Each of the two fuzzy sets (arrival rate, response time) has seven membership functions. The arrival rate and response time are termed as conditional attributes, it forms 49 (calculated as (72))fuzzy rules which have been summarized in Table 4. The 'replication requirement' is termed as the decision attribute. The decision attribute hasthree categorical values namely 'R' (Replication required), 'N' (Replication NOT required), 'S' (Stop existing replica). These decision attributes are arrived by formations of rules as explained in Table 3. These FL decisions are illustrated in Table 4 and stored in variable 'FL_flag_replication_required' which will be used further by RCA.

Transform output to non- fuzzy value

In our framework we have used center of gravity method [25] to defuzzify the strength of rule evaluation to identify the crisp output.

This method is illustrated in Eq. 14 as follows $$\delta = \frac{\sum_{i=1}^{n} m^i w_i}{\sum_{i=1}^{n} m^i} \qquad (14)$$

ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences & Management Research

Table 2: Linguistic Variables for FR

SL No	Crisp Input Range	Linguistic Variables
1	0 - 250	Very Very Fast(VVF)
2	50 - 450	Very Fast(VF)
3	250 - 650	Fast(FS)
4	450 - 850	Normal(NR)
5	650 - 1050	Slow(SL)
6	850 - 1250	Very Slow(VS)
7	1050 - 1450	Very Very Slow(VVS)

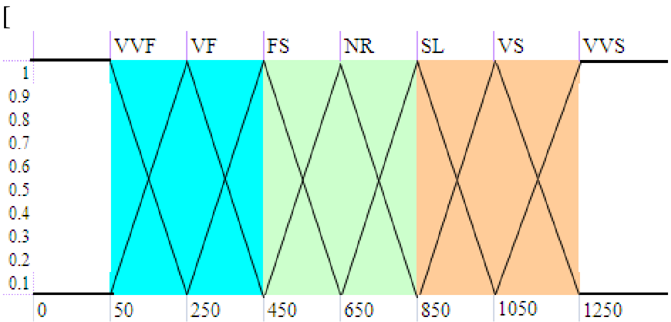


Fig. 4 Fuzzy Sets (β) of Response Time

Table 3: Fuzzy Rules

Rule#	Fuzzy Rules Definition
1	IF {FR is (VVS) AND FA is (VVH or VH or HG or NR)}OR {FR is (VS) AND FA is (VVH or VH or HG) } OR {FR is (SL) AND FA is (VVH or VH or HG)}OR {FR is (NR) AND FA is (VVH)} Then Replication Required = "R"
2	IF{FR is VVS AND FA is (LW or VL or VVL)} OR {FR is VS AND FA is (NR or LW or VL or VVL)} OR {FR is SL AND FA is (NR or LW or VL or VVL)} OR {FR is NR AND FA is (VH or HG or NR or LW or VL)}OR {FR is FS AND FA is (VVH or VH or HG or NR) } OR {FR is VF AND FA is (VVH or VH or HG or NR) } OR {FR is VVF AND FA is (VVH or VH or HG) } Then Replication Required = "N"
3	IF {(FR is NR) AND FA is (VVL)} OR {(FR is FS) AND FA is (LW or VL or VVL) } OR {(FR is VF) AND FA is (LW or VL or VVL) } OR {(FR is VVF) AND FA is (NR or LW or VL or VVL)} Then Replication Required = "S"

Table 4: Replication Matrix

FA / FR	VVH	VH	HG	NR	LW	VL	VVL
VVS	R	R	R	R	N	N	N
VS	R	R	R	N	N	N	N
SL	R	R	R	N	N	N	N
NR	R	N	N	N	N	N	S
FS	N	N	N	N	S	S	S
VF	N	N	N	N	S	S	S
VVF	N	N	N	S	S	S	S

Resource Control Service And Its Algorithm

The RCS periodically monitors the health status of *ws* at every T_h interval defined in the system (for example, every 10 seconds). If any of the *ws* hung or stopped, it will restart the hung/stopped *ws*. For cases where the restart option fails, the following status flag "Service_*Status_Abnormal"* will be set to 'Yes'. In addition to this, RCS also evaluates the performance of individual servers with respect to response time, if any *ws* exceeds 90% of of SLA, the flag "*RST_Th_Exceed" is set*to 'Yes'. It is to be noted that RCS does not instantaneously respond to the FL decision since the frequency of prediction interval is very short (60 seconds). In order to analyze the consistency of system behavior, RCS uses the following two time parameters: *1. time_wait_to_stop* (T_{stop}) – the waiting time of the system after FL decision is started to set to 'S', 2. *time_wait_to_repl* (T_{repl}) - the waiting time of system after FL decision is set to 'R'. These two values can be parameterized by the administrator. For the purpose of our experiments, we have used 5 min. for (T_{repl}) and 15 min for (T_{stop}). RCA uses these variables (detailed in Fig. 5) for the purpose of: 1). Replicate new wsor stop existing *ws*upon FL decision. 2). autonomously decide to replicate new *ws* immediately when it finds any *ws* status flagged as *abnormal*, 3). Replicate new *ws,* if response time of any individual server approaches SLA (*RST_Th_Exceed = 'Yes')* and 4. Stop the abnormal *ws*.

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

Fig. 5 Resource Control Algorithm (RCA)

```
If [( FL_flag_replication_required ='S')   AND
    (Waiting  time duration  >=T_stop)]
then
      Begin
          get (R_x, SC_x)which has greater CR
          Call function stop_replica_service(R_x, SC_x)
       End
Else-if  [(Service_Status_Abnormal(R_x, SC_x) = 'Yes']
then
      Begin
          Call function start_replica_service(R_x+1 ,SC_x)
          Call function stop_replica_service(R_x, SC_x)
      End
 Else-If  [( FL_flag_replication_required ='R' ) AND
             (Waiting  time duration is  >=T_repl )
 then
       Begin
Call function start_replica_service(R_x+1 ,SC_x)
       End
Else-if  [(RST_Th_Exceed(R_x, SC_x) = 'Yes')]
 then
           Begin
              if  [(FL_flag_replication_required <>'S')]
                 Begin
           Call function start_replica_service(R_x+1,SC_x)
                  End
              end-if
              Call function stop_replica_service(R_x, SC_x)
           End
Else
      continue
End-if.
```

Based on the RCA algorithm, RCS decides to replicate the *ws* or reclaim the resource dynamically and maintain the response time before it exceeds SLA time

RESULTS AND DISCUSSION

Environment

This section explains the test environment and lays down the various test scenarios used to evaluate the performance of FAPFEA and analyze the results.The environment used for evaluation comprised of VMware on Windows 2003 server platform with MS-SQL Server 2005. Applications Manager was used for monitoring and JMeter was used for testing purposes. Load balancing was done by ACE and Tomcat server, eclipse were used for Application development and implementation over a network bandwidth of 100mbps. We employed 2 *ws* (SC_1, SC_2) oneach replication servers $R_1 \ldots R_{15}$ which are connected through LAN (Local Area Network). The SLA time defined for *ws* SC_1 and SC_2 are1300ms and 2400ms respectively. Simulation of concurrent HTTP requests wasinitiated from 5 client machines using JMeter with benchmarking tool ranging from 1000 to 10,000 requests/min. The requests submitted will step up/down by every 15 min.

Experiments

*5.2.1.Conventional system test:*This test has been conducted to understand the behavior and performance of the conventionalsystem. Here ten servers were engaged with *ws* SC_1 and SC_2 and the HTTP requests sent to *ws* ranged from 1000 requests/minute to 10,000 requests/minute. The graphical representation of the test results are shown in Fig. 6.It can be inferred from the data and graph that when the arrival rate is greaterthan 7000

ISSN 2349-6193
Impact Factor: 2.805
IJESMR
International Journal OF Engineering Sciences & Management Research

requests/minute, the response time exceeds SLA time and request failure rate also rises. The system is found to be incapable of processing all the requests as the system does not possessthe ability to replicate new *ws* automatically. In this situation, the *ws*is considered as un-available.*5.2.2. Reliability test:*The aim of this test is to understand the behavior of the framework when system is subject to request over load and check its ability to accurately identify the replication requirement and replicate the *ws* dynamically without administrator intervention. Initially, the test was conducted on ten replica servers (R_1, R_2,..R_{10}) with the above mentioned two *ws* (SC_1 *and* SC_2) running on each server. For understanding the test process, we shall consider explaining only *ws*SC_1 in this segment. HTTP requests were sent to *ws* SC_1 at a frequency ranging from 7000 requests/minute to 10000 requests/minute and the request frequency was stepped up every 15 minutes by 1000 requests. For instance, here itis being explained that the

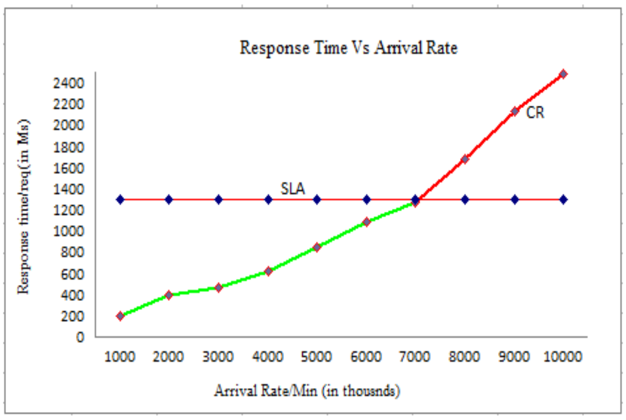


Fig. 6 CA vs. CR (Conventional system test)

Testing of average current arrival rate CA@ 7500 requests/minute continued for 15 minutes. Under this test scenario2, SC_1 was subjected to approximately 660000 requests. The statistics were collected from MS, and CA (7500) was arrived using Eq. (2) and CR (1004) was arrived at by using Eq. (8). FA and FR were predicted and the calculations and algorithms used in our framework are explained in detail for CA and CR as follows.

The FPS has predicted the FA_1 using Eq. (4) and Eq. (5) for CA@7500 with σ_u =90%; and σ_v =10%;

FA1 = f (x; 90)= 8600 requests/minute
FA2 = f (x; 10) = 6400 requests/minute

The total predicted future arrival rate is calculated using Eq. (6).

FA = 8600*0.9 + 6400*0.10
= 7740 + 640
= 8380 requests/minute

FR_1 is predicted using Eq. (11) for the CR (1004 ms) with θ_J =75%; and θ_k =25%.

FR1 = f (x; 75) = 1391.8395 ms;
FR2 = f (x; 25) = 288.8328 ms

The total predicted future response time is calculated using Eq. (13).

FR = T_f = 1391.8395*0.75 + 288.8328*0.25
=1043.8796 + 72.2082
= 1116.0878ms.

The FA and FR are converted into fuzzy sets and represented in Fig. 7 and Fig.8 respectively.The each fuzzy rules are fragmented into few segments using 'OR' clause and evaluated as follows:

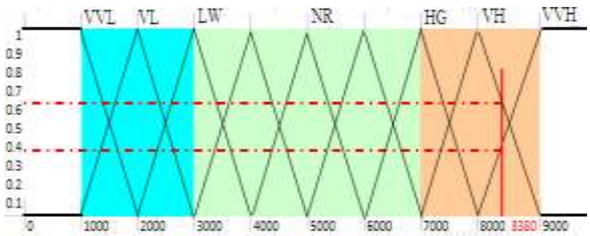
$\gamma_1 = \{A_1$ [OR] B1 [OR] C_1 [OR] D1$\}$
$\gamma_1 = \text{Max } (A_1 \cup B_1 \cup C_1 \cup D_1)$
γ_1 = Max (0.13, 0.62, 0, 0)
γ_1 = 0.62

ISSN 2349-6193
Impact Factor: 2.805
IJESMR
International Journal OF Engineering Sciences & Management Research

$\gamma_2 = \{A_2 \text{ [OR] } B_2 \text{ [OR] } C_2 \text{ [OR] } D_2 \text{ [OR] } E_2 \text{ [OR] } F_2 \text{ [OR] } G_2\}$
$\gamma_2 = \text{Max}\,(A_2 \cup B_2 \cup C_2 \cup D_2 \cup E_2 \cup F_2 \cup G_2)$
$\gamma_2 = \text{Max}\,(0, 0, 0, 0, 0, 0, 0)$
$\gamma_2 = 0$
$\gamma_3 = \{A_3 \text{ [OR] } B_3 \text{ [OR] } C_3 \text{ [OR] } D_3\}$
$\gamma_3 = \text{Max}\,(A_3 \cup B_3 \cup C_3 \cup D_3)$
$\gamma_3 = \text{Max}\,(0, 0, 0, 0)$
$\gamma_3 = 0$

Using center of gravity method Eq. (14), these values are defuzzified as illustrated in Fig.9 and derived as follows:

$\delta = 450 * 0.62 / 5*0.62$
$= 279 / 3.1 = 90$



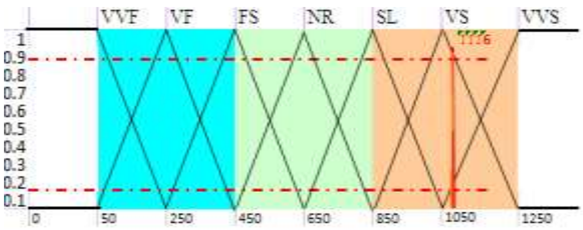


Fig. 7 Fuzzy Sets representation of FAFig. 8 Fuzzy Sets representation of FR

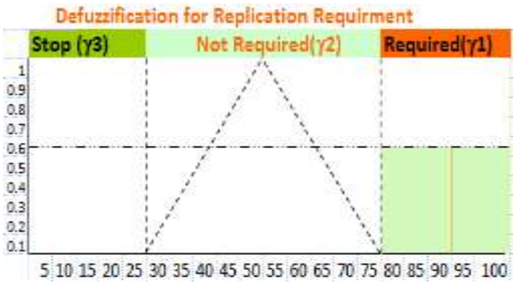


Fig. 9 Crisp output

The test results are represented in graphical form in Fig. 10. Since the FL decision returned is 'R', it is understood that replication is required. The RCS then analyzes the FL decision using RCA and monitors the FL flag for T_{repl} time (5 minutes) to decide on the replication requirement. Here, FL is set to 'R' for T_{repl} time continuously, so the RCS decides to replicate and server R_{11} is replicated for *ws* SC_1. Server R_{11} is now added to the server pool by RCS thereby distributing the load on the new replica *ws*, which is effected by LBS. As a result, CR of SC_1 is reduced before it crosses the SLA time as shown in Fig. 10. Similarly, when CA of SC_1 reaches 10000 requests/minute and CR approaches the SLA, our framework infers the replication requirement and replicates server R_{12} with *ws* SC_1. Thus, the response time of *ws* is maintained by our proposed framework even during heavy load period by replicating new servers dynamically as and when required.

IJESMR
International Journal OF Engineering Sciences & Management Research

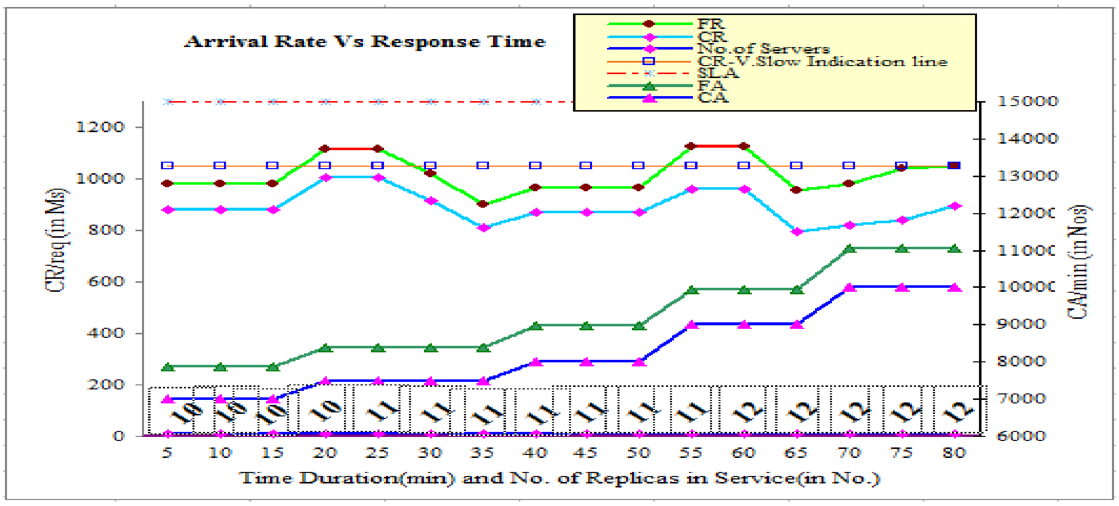


Fig. 10 AR VsCR (Reliability test for dynamic replication scenario)

CONCLUSION

In this paper, we compared a low-power 8-point DCT approximation that require only 14 addition operations to computations which has all good advantages compared to the other DCTs that are proposed in this survey and hardware implementation for all the transform including other prominent approximate DCT methods, including the designs by Bouguezel-Ahmad-Swamy DCT perform very close to the ideal DCT. However, the modified CB-2011 approximation and the 8-point Approximate DCT possess lower computational complexity and are faster than all other approximations under consideration.

5.2.3. Sustainability test:This simulation test aims to ascertain the ability of the proposed framework to identify situations that do not require replication of ws; instead, the response time of ws is maintained within SLA requirements. In order to test this scenario3, we activated seven servers with ws SC1 and HTTP requests to SC1 were sent at frequencies ranging from 3001 to 6000 requests/minute and stepping up the load every 15 minutes by 1000 requests/min. In total, approximately 270000 requests were generated and processed by SC1 for this test. The statistics were collected from MS, and CA (5000) and CR(621) was arrived by using Eq. (2) and Eq. (8) respectively

FA and FR were predicted by FPS; the test results are shown in Fig. 11. The FPS predicts the FA as 5720.0 using Eq. (4), Eq. (5), Eq. (6) and FR have been predicted as 690.32915 using Eq. (11), Eq. (12), and Eq. (13).

The defuzzifying of the above values using Center of Gravity method Eq. (14) is resulted as follows:

$$\delta = 525*0.76 / 10*0.76$$
$$= 52.5.$$

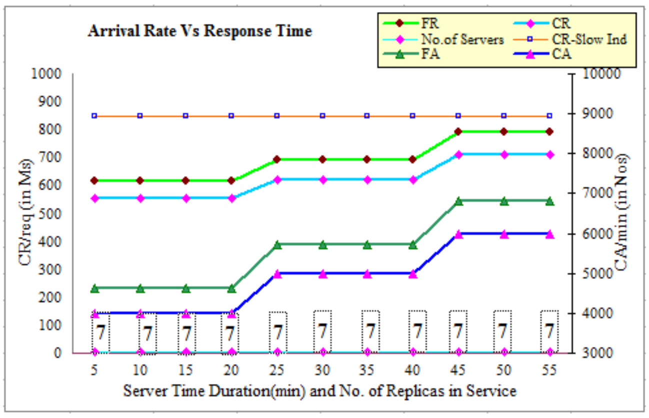


Fig. 11 AR Vs.CR (Sustainability Test scenario)

ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences & Management Research

By virtue of the above output, *FL_flag_replication_required* is set to *'N'.* As the FL decision reveals that replication is not required, RCS does not replicate the server and continue its prediction process for the next interval as the CR is within expected SLA time for the current time interval.

*5.2.4. Feasibility test:*The aim of this test is to check the ability of proposed framework to accurately identify the excess resources employed and the ability to reclaim the resources proactively. In order to test this scenario, we activated three servers with *ws* SC_1 and HTTP requests were sent to SC_1 with constant load at 900 requests/min. The statistics were collected from MS, and CA (900) and CR (150) were calculated using Eq. (2) and Eq. (8) respectively. FPS predicted the FA as 1220 using Eq. (4), Eq. (5), Eq. (6) and FR as 166.75 using Eq. (11), Eq. (12) and Eq. (13). The test results are shown in Fig. 12. Defuzzifying of the values using Center of Gravity MethodEq. (14) is resulted as follows:

$\delta = 70 * 0.7 / 5 * 0.7 = 14$

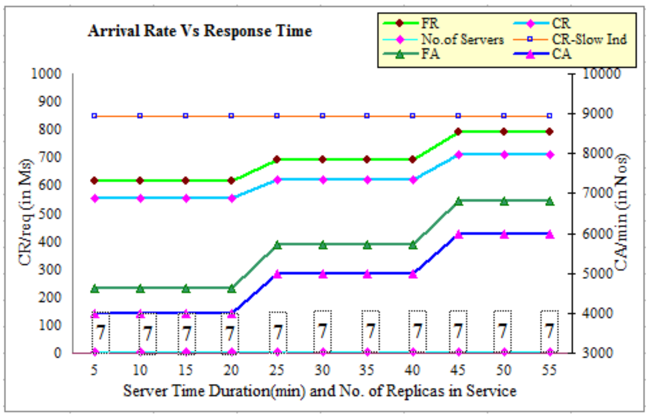


Fig. 11 AR Vs.CR (Sustainability Test scenario)

By virtue of the above output, *FL_flag_replication_required* is set to *'N'.* As the FL decision reveals that replication is not required, RCS does not replicate the server and continue its prediction process for the next interval as the CR is within expected SLA time for the current time interval.

*5.2.4. Feasibility test:*The aim of this test is to check the ability of proposed framework to accurately identify the excess resources employed and the ability to reclaim the resources proactively. In order to test this scenario, we activated three servers with *ws* SC_1 and HTTP requests were sent to SC_1 with constant load at 900 requests/min. The statistics were collected from MS, and CA (900) and CR (150) were calculated using Eq. (2) and Eq. (8) respectively. FPS predicted the FA as 1220 using Eq. (4), Eq. (5), Eq. (6) and FR as 166.75 using Eq. (11), Eq. (12) and Eq. (13). The test results are shown in Fig. 12. Defuzzifying of the values using Center of Gravity MethodEq. (14) is resulted as follows:

$\delta = 70 * 0.7 / 5 * 0.7 = 14$

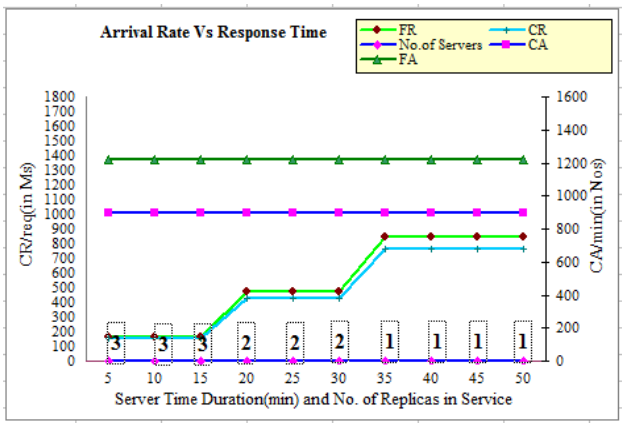


Fig. 12 AR Vs.CR (Feasibility test scenario)

Therefore, *FL_flag_replication_required is set to 'S'.* Here, the FL flag for replication was set to 'S' as the FL decision reveals that employed resources exceeded the actual requirement and therefore excess resource needs to bereclaimed. RCS uses the FL's decision and applies it in RCA which monitors the FL Flag for T_{stop} time. Since the FL Flag is 'S' for T_{stop} time, RCS decides to remove the excessive replica which runs with higher CR. Similarly, the system detects and removes such excess replicas running on high CR, every 15 minutes. Thus, our proposed framework has the ability to remove and reclaim the excessive resource as and when required.

*5.2.5. Comparative test:*With the aim to compare the efficiency of FAPFEA, we have analyzed and compared our test results with the results of Linear Regression Model (LRM) suggested by the author in [19] and Halts Linear Exponential Smoothing Model (HLESM) suggested by authors in [20] for the number of replicated *ws* provided by each model toprocess the same load. LRM replicates *ws* when Load >75% or predicts the 16th day load using two weeks history. On the other hand, HLESM computes the response time using QN model and predicts response time using HLESM and replicates *ws* when predicted response time exceeds 90% of SLA. The replication techniques for the two models are computed mathematically for different loads varying from 900 requests/minute to 2400 requests/minute. The results are compared with FAPFEA and shown in Fig. 13. The result shows that LRM replicates earlier than required because it uses only the server load as replication criteria for replication. While, HLESM replicates the *ws* at slightly more appropriate time than LRM. It can be inferred from the graph that FAPFEA predicts the replication requirement accurately and at the most appropriate time as compared to other existing replication models [19][20].

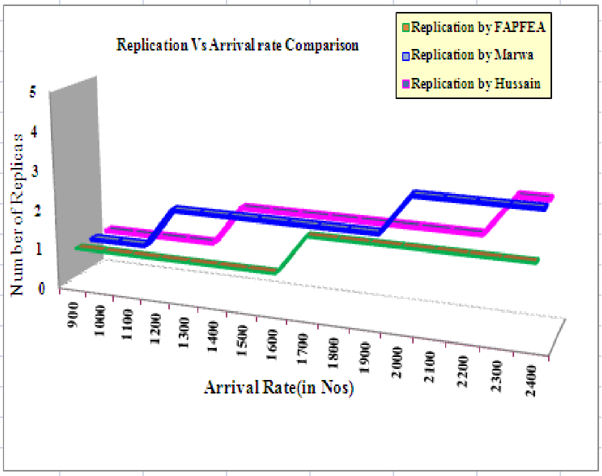


Fig. 13 Varying load replication

The test result proves that our proposed framework is efficient in dynamically replicating the *ws*on-demand which improves the availability of *ws* to respond to client requests within the defined SLA time for all constant, varied and robust load conditions. Hence, *ws* availability is achieved in qualitative as well as quantitative terms by maintaining the response time within SLA time and by maintaining the continuous availability of *ws* to provide uninterrupted responses respectively under any stressed scenario.

CONCLUSION AND FUTURE WORK

Through this paper, we have presented an adaptive prediction framework for improving the availability of *ws*, which predicts both arrival rate and response time of *ws* using PD and ED respectively. FL is used to evaluate various possible rules formed between arrival rate and response time and to determine the replication requirement. Depending on the FL decision, RCS uses RCA to replicate the *ws* on another server host whenever predicted response time violates the Service Level Agreement (SLA) or reclaim the existing excess resource. In addition to this, it detects the failure of any *ws* or server. In case of *ws* failure, it restarts the *ws* and if the restart process fails,it replicates the *ws* on another server. In a situation where the server fails, it automatically replicates *ws* on a new server. Also, in the case where response time of any individual server is low and is approaching the thresh-hold limit, it replicates the *ws* on another server and stops thereplica with lower performance. We have simulated several testing scenarios proving that our framework replicates the *ws* and reclaims the unnecessary resources dynamically based on demand.

IJESMR
International Journal OF Engineering Sciences & Management Research

The advantages of our proposed framework is that it is proactive, dynamic, on-demand and is an accurate *ws* replication model which responds to the requests within expected SLA response time. Thus, our framework is efficient in significantly improving the availability of *ws* to process all requests successfully within the expected SLA response time and to provide uninterrupted availability of *ws* for any volume of load with minimal administrator intervention. Since our framework meets the basic attributes such as sustainability, scalability and elasticity, it provides an optimal solution to the current booming internet business world. Our future works will be directed towards implementing our framework on the cloud computing *ws* and analyzing its performance for large scale applications.

REFERENCES

1. *Benaatallah, B., Sheng, Q., Z.,& Dumas, M., 2003. The self - Serve Environment for Web Services Composition.IEEE Internet Computing 7(1), p 40-48.[10.1109/MIC.2003.1167338]*
2. *Horn, P., 2001. Autonomic Computing: IBM's Perspective on the State of Information Technology, Technical Report, IBM Corporation. October 15, p 1-22.*
3. *Ganek, A., G. and Corbi, T., A., 2003. The dawning of the autonomic computer era. IBM Systems Journal 42(1), p 5-18. [10.1147/sj.421.0005]*
4. *Kounev, S., Nou, R., Torres, J., 2007. Autonomic QoS-aware resource management in grid computing using online performance models. In Proceedings of the 2nd International Conference on performance Evaluation Methodologies and Tools, ICT (Institute of Computer Sciences, Social-Informatics and Telecommunications Engg), Nantes, France.*
5. *Folino, G., and Spezzano, G., 2007. An autonomic tool for building self-organizing grid enabled applications. Future Generation Computer Systems 23, p 671-679.*
6. *Agarwal, M., Bhat, V., Liu, et al., 2003. AutoMate: enabling autonomic applications on the grid. Proceedings of the Autonomic Computing Workshop, p 48-57. [10.1109/ACW.2003.1210196]*
7. *Quan Z Sheng, Zakaria Maamar, Jian Yu., 2009.Robust Web Services Provisioning Through On-Demand Replication.Information Systems: Modeling, Development, and IntegrationSpringer 20, p 4-16. [10.1007/978-3-642-01112-2_3]*
8. *Mathias Bjorlqvist, Lydia Y., Chen, 2011.Optimizing service replication in Clouds.Proceedings of the IEEE Winter Simulation Conference, p 3307-3317. [10.1109/WSC.2011.6148027]*
9. *Marin Silic, GoranDelac, Ivo Krka, et al., 2014. Scalable and Accurate Prediction of Availability of Atomic Web Services.IEEE Transactions on Services Computing, 7(2).[10.1109/TSC.2013.3]*
10. *Subil Abraham, Mathews Thomas, Johnson Thomas 2005. Enhancing Web Services Availability.Proc. 26th IEEE International Conference on Software Engineering, p 352-355. [10.1109/ICEBE.2005.62]*
11. *Birman, K., Van Renesse, R., Vogels, W., 2004.Adding High Availability and Autonomic Behavior to Web services.Proc International Conference on 26th IEEE on Software Engineering,p 17-26.[10.1109/ICSE.2004.1317410]*
12. *Abd-El-Barr, M., 2007. Design and Analysis of Reliable a Fault-tolerant Computer Systems. Imperial College Press, London.*
13. *Maamar, Z., Sheng, Q., Z., Benslimane, D., 2008. Sustaining web services high - availability using communities. The Third International Conference on Availability, Reliability and Security, p 834-841. [10.1109/ARES.2008.7]*
14. *Sattanathan Subramanian 2009.Highly - Available Web Service Community.6th International conference on Information Technology, New Generation, IEEE, p 296-301. [10.1109/ITNG.2009.128]*
15. *Jorge Salas, Francisco Perez-Sorrosal, Marta Patino-Martinez, et al., 2006. WS-Replication: A Framework for Highly Available Web Services. 15th International World Wide Web, p 357 - 366. [doi.acm.org/10.1145/1135777.1135831]*
16. *Sundharam, R., Lakshmi, M., Abarajithan, D., 2010. Enhancing the High Availability of Web Services for Mission Critical Applications.International Conference on Trendz in Information Sciences and Computing, IEEE – TISC 2010, p 149-151. [10.1109/TISC.2010.5714627]*
17. *Nabor, C., Mendonca, Jose Airton, F., et al., 2008. Client side selection of replicated web services: An empirical assessment. Journal of Systems and Software. Elsevier Science Inc. 81(8), p 1346-1363. [10.1016/j.jss.2007.11.002]*
18. *Xinfeng Ye, Yillin Shen, 2005. A Middleware for Replicated Web Services. The Proceedings of International Conference on Web Services – ICWS'05, p 631-638. [10.1109/ICWS.2005.8].*
19. *Marwa, F., Mohamed, Hany, F., et al., 2013. A study of an adaptive replication framework for orchestrated composite web services. Springer Plus (2013), doi: [10.1186/2193-1801-2-511].*

20. *Mohamed K Hussein and Mohamed H Mousa., 2012. A Framework for adaptive QoS of Web Services using Replication. International Journal of Computer Science and Communication Networks 2(2), p 288-294.*
21. *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.85, p 2048– 2062.[10.1109/TENCON.2011.6129077]*
22. *Hwangm Haojun Wang, Jian Tang, Jaideep Srivastav 2007.A probabilistic approach to modeling and estimating the QoS of web-services-based workflows.International Journal of Information Sciences,177 (23), p 5484-5503. [10.1016/j.ins.2007.07.011]*
23. *Marco Conti, Mohan Kumar, Sajal K Das, et al., 2002.Quality of Service Issues in Internet Web Services.IEEE Transaction on Computers.51(6), p 593 – 94. [10.1109/TC.2002.1009145]*
24. *Jerry Banks, Chapter 5.Statistical Models in Simulations and Chapter 6. Queuing Models, Systems Simulation, pages 168, 172 and 222.*
25. *Li-Xin Wang, Jerry M Mendel 1992, Generating Fuzzy Rules by Learning from Examples, IEEE Transactions on Systems, Man and Cybernetics,.22(6).*
26. *Lotfi,A., Zadeh, 2008. Is there a need for fuzzy logic?*Information Sciences, an International Journal (178),p 2751–2779.*
27. *Murugan, S., Ramachandran, V., 2014.Fuzzy Decision Making Model for Byzantine Agreement Journal of Engineering Science and Technology, 9(2) 206 – 219.*
28. *Singhala, P.,Shah, D.,N.,Patel, B.,2014. Temperature Control using Fuzzy Logic, International Journal of Instrumentation and Control Systems IJICS, 4(1), January 2014. [10.5121/ijics.2014.4101]*