International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 3, March-2014, pp: (574-581), Impact Factor: 1.252, Available online at: www.erpublications.com

Performance Parameter Analysis of Mobile Agent based web service composition using WSDL Analysis

Mr. Sunil R. Dhore[1], Prof. Dr. M. U. Kharat[2]
[1]Member IEEE, Army Institute of Technology, Pune, M.S. India
[2]MET's Institute of Engineering, Bhujbal Knowledge City, Nashik, M.S. India

Abstract: There are various quality criteria that are important to Web services but execution performance of web service is important Quality criteria which play an important role in Web Services, as they differentiate similar services by their execution performance. There are various guidelines for Improving Web Services Performance. Quality based web services enable service requesters to choose and bind to a suitable Web service at run time based on their execution performance. This paper proposes a quality criteria classification that organizes web services qualities into four groups: performance, failure probability, trustworthiness and cost. The quality criteria classification is specified within the Web Service Description Language (WSDL). Also The paper demonstrates an integrated mobile agent approach with web service and effect of mobile agent approach on the quality parameters of the web services particularly on the performance. Also further we can select the web services based on these performance criteria of the web services and use the same for web service composition.

Index Terms: Web services, quality criteria, WSDL, mobile agent, web service composition.

I. INTRODUCTION

Web services are a technology, which allows applications to communicate with each other in a platform and programming language- independent manner over the Internet. Web services achieve system interoperability by exchanging an application development and service interactions using the XML-based standards such as Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). With the growing popularity of Web services, a quality criteria support for Web services will play an important role for the success of this emerging technology. This paper proposes quality criteria classification that organizes quality criteria into four groups: performance, failure probability, trustworthiness and cost. The current Web service core technologies (SOAP, WSDL, and UDDI) are immature and still under development by the W3C [1]. UDDI is just a registry database and allows service requesters to look for Web services based on their functionality but not quality information. WSDL is an XML format for describing Web services. These technologies do not address issues related to the description of quality aspects of a service. To overcome the WSDL and UDDI limitations, the following approaches are introduced. We present an other approaches along with WSDL analysis like architecting to include quality criteria classification and we extend the current Web service architecture with quality server to enable the UDDI to publish and discover services based on the proposed quality criteria classification by using the mathematical method. To increase the performance of the web service we are proposing the integration of web services with mobile agents. And then these best suited web services are used to build the composite web services.

II. QUALITY CRITERIA IN WEB SERVICES [16]

A. Quality Definition

Quality criteria may have different definitions in different domains. However, in the Web services context, Quality criteria can be defined as a set of non-functional criteria such as availability, performance and reliability that impact the performance of Web services. Quality is the measure of how well does a particular service perform relative to expectations, as presented to the requester. It determines whether the requester will be satisfied with the service delivered, that is, the quality is meeting requirements.

B. Quality Criteria Classification

The quality criteria classification in this paper is similar to the quality classification in [2], in that they classify the quality criteria into groups with different perspectives. The quality classification includes three groups: performance, safety and cost. Performance contains response time and throughput, safety contains availability and reliability and cost contains the service cost. The quality classification organizes the most important quality-of-service to Web services into four groups: QoS related to runtime, transaction support, configuration management and cost and security. The quality classifies the QoS parameters into the following groups: general, Internet service specific and task specific. General QoS parameters contain performance (throughput), performance (latency), reliability and cost. Internet service specific QoS parameters contain availability, security, accessibility and regulatory. Task specific QoS parameters contain task specific parameter. This section represents a quality criteria classification that organized into four groups: performance, failure probability, trustworthiness, and cost. These groups are organized regarding its characteristics and include generic criteria. The performance out of these is important criteria and for the performance Response time, throughput, capacity, latency, execution time, transaction time is analyzed and from the WSDL analysis the operation, data type also consider. these criteria can benefit all service requesters to select the best possible service for the composite of web service. When integrating web services with mobile agents it is expected that Response time, throughput, capacity, latency, execution time, transaction time will improve due to localization of web service.

C. Performance issues

- Web services should execute quickly
- Complete the requested task quickly
- Minimize delays in message delivery & processing times with increased traffic
  - CPU intensive Processing :SSL,XML Parsing, XSLT, Header & Payload processing
  - Web service performance can be analyzed from different view points
    Service consumer eg. Response time, connection errors
    Service producer eg. Transaction/sec., concurrent users
    Process perspective eg. Time to perform business.
  - Define Requirement & Gather Metrics :
    Define QoS for your web service
    Think SMART : Specific, Measurable, in Agreement with Responsibility, testable

D. Common Metrics

- End to end response time
- Site response time
- Throughput (request/sec)
- Throughput (Mbps)
- HTTP or other errors/sec
- Transaction per day
- Latency :Time between client initiating request and server processingIncludes SOAP message marshaling , un marshaling
- Execution Time :Time taken by endpoint to perform business task
- Response time :Latency + execution time, Viewed from a network node's exchanges
- Transaction time :Time taken to execute business task, May involve multiple SOAP message exchanges
- Throughput :Amount of data process by the endpoint
- Capacity: The limit of concurrent requests that the service support for guaranteed performance

III. MATHEMATICAL MODELS FOR THE PERFORMANCE METRICS OF THE WEB SERVICES

The performance of a Web services measure the speed in completing a service request. It can be measured by [14]:

A. Capacity

The limit of concurrent requests that the service support for guaranteed performance.

There are three variables that form the basic model of system capacity. These variables are

- Observation time (T), the amount of time that the server is monitored for activity
- Busy time (B), the amount of time that the server was active during the observation time
- Completions (C), the number of transactions completed during the observation period

With these three variables, you can calculate the six significant values, described in Table:01 that are used to develop a capacity planning model.

Data	Description	Formula
CPU Utilization	The percentage of CPU capacity used during a specific period of time.	U = B/T
Transaction throughput of the system	The average number of transactions completed during a specified period of time.	X = C/T
Average service time	The average time to complete a transaction.	S = B/C
Transaction capacity of the system	The number of transactions the server handles.	Cp = 1/S
Average queue length	The average number of transactions in queue.	Q = U/(1-U)
Average response time	The average time to respond to a transaction.	R = (QÃƒÂƒÃ‚,Â—S)+S

Table 01 : Capacity Planning Data Formulas

Here is an example of how to use these formulas to size a server. Suppose that you observe the server for 60 seconds (T), during which time there are 90 completed transactions (C), and the server is actually busy processing that workload for 48 seconds (B). Table 02 shows the resulting data values using this information.

Resource	Formula	Result
CPU Utilization	U = B/T	48/60 = 80 percent utilization
Average transaction throughput of the system	X = C/T	90/60 = 1.5 transactions/sec
Average service time	S = B/C	48/90 = .53 seconds
Transaction capacity of the server	Cp = 1/S	1/.53 = 1.875 transactions/sec
Average queue length	Q = U/(1-U)	.8/(1 - .8) = 4 transactions
Average response time	R = (QÃƒÂƒÃ‚,Â—S)+S	(3 ÃƒÂƒÃ‚,Â— .53)+.53 = 2.12 seconds

Table 02: Capacity Planning Resource Formula Results

The CPU utilization was at 80 percent, and handled an average of 1.5 transactions per second. The average service time for these transactions was .53 seconds, and transactions were completed in an average time of 2.12 seconds. On average, there were four transactions waiting to be processed at any given point in time during the observation period, and the server had the capacity to process 1.875 transactions per second.

B. ModellingCapacity

Describes the workload for web services capture resource demands and Workload parameters.
Intensity : requests/day, Messages per day per customer, transaction rate, concurrent users
Service demands : message size, Number of users , CPU/Memory utilization etc.
Response time : The maximum time that elapses from the moment that a web service receives a SOAP request until it produces the corresponding SOAP response. Response time is positively related to capacity.

Latency: The round-trip time between the service request arrives and the request is being serviced.

Throughput: The number of Web service request completed at a given time period. It is the rate at which a service can process requests. Throughput is related negatively to latency and positively to capacity.

Throughput = (Total requests * Average Size) / Time period

Ex : web service should processes 9000 HTTP Requests in 30 min with a SOAP response message size467 Kb
Throughput = (9000*467000) / 1800
= 1,425,39 Kbps

Execution (processing) time : The time taken by a Web service to process its sequence of activities
In general, high performance Web services should provide higher throughput, higher capacity, faster response time, lower latency, and lower execution duration.

C. WSDL Analysis

WSDL Contents[1]:

1. Operation type
2. Data types
3. Messages
4. Port types
5. Bindings

A WSDL document describes a web service using these major elements:

The request-response type is the most common operation type, but WSDL defines four types:

Operation type:

Type	Definition
One-way	The operation can receive a message but will not return a response
Request-response	The operation can receive a request and will return a response
Solicit-response	The operation can send a request and will wait for a response
Notification	The operation can send a message but will not wait for a response

Element	Description
<types>	A container for data type definitions used by the web service
<message>	A typed definition of the data being communicated
<portType>	A set of operations supported by one or more endpoints
<binding>	A protocol and data format specification for a particular port type

Observations:

Based on the contents of the WSDL, which describe the web service the performance of the web service may be calculated.

IV. MOBILE AGENTS AND WEB SERVICE

A. Models and Functions of Web Services and Mobile Agent Systems[13]:

One of the most relevant differences between MA systems and WS is the nature of the interactions between their components. While WS adhere to a typical synchronous paradigm in which a client requests service execution to a server endpoint, MAs interact with clients by following a typical asynchronous behavior: after creation, MAs can migrate between network nodes and interact with their needed resources, with no necessity to maintain continuous connectivity with the associated client.

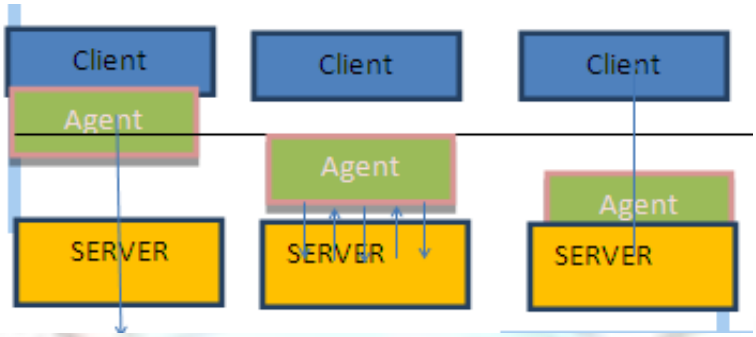


Figure 01: Mobile Agent Architecture

B. Advantages of Mobile Agents:

Reduced network bandwidth
Disconnected operation
- Short "On-Line" times
- Low-power requirements
- Support for mobile units

Low-latency interaction

C. Performance issues after integration of the web services with mobile agents :

As it is clear from the architecture that the operations happen after the integration of the web services with mobile agents will make the all the operation asynchronous so there is more improvement and solidity in calculating the performance parameters, Some of the observations on the performance parameters after integration will be as follows:

Response time: Due to localization response time of the web service will be reduce by at least 50%.
Throughput: No. of services provided per unit period will not change much rather it will be reduces by 10%.

Operation Type (WSDL):As the web services are locally run the all the operations are one way only so it reduce the traffic by 50%, Data Type (WSDL): There is no change in the performance for these parameters. But if it could reduce the complexity of the web service if we can convert the composite data type in primary data type during the integration of web service to mobile agents. Latency: OverallLatency will be reduce as response type reduces and all the operations are becoming one-way

Execution time: There is no change in execution time.
Transaction time: So overall transaction period will be reducing.

D. Enabling Web Services Composition with Software Agents:

So, We proposed a framework to support QoS aware Web Service Composition. We add two layers between service requirement and web service candidates, as shown in figure 04. In the first layer, each Web Service candidate is linked to a home norm base, which can be used by home agent to negotiate among other home agents.

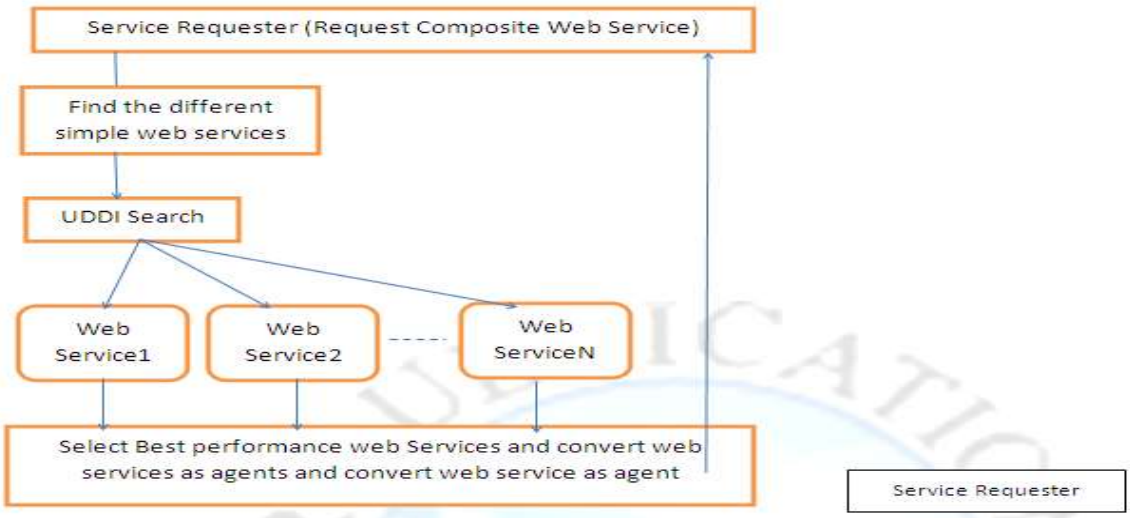


Figure 02: Proposed Model

V. COMPOSITION OF WEB SERVICES

A. ABSTRACT AND COMPOSITE WEB SERVICES:

As shown in Fig.03 we distinguish in the composition process between the following two concepts:

- An abstract composite service, which can be defined as an abstract representation of a composition request CSabstract = {S1, . . . , Sn}. CSabstract refers to the required service classes (e.g. flight booking) without referring to any concrete web service (e.g. Lufthansa flight booking web service).
- A concrete composite service, which can be defined as an instantiation of an abstract composite service. This can be obtained by binding each abstract service class in CSabstract to a concrete web service sj , such that sj € Sj . We use CS to denote a concrete composite service.

Input: Abstract Composition

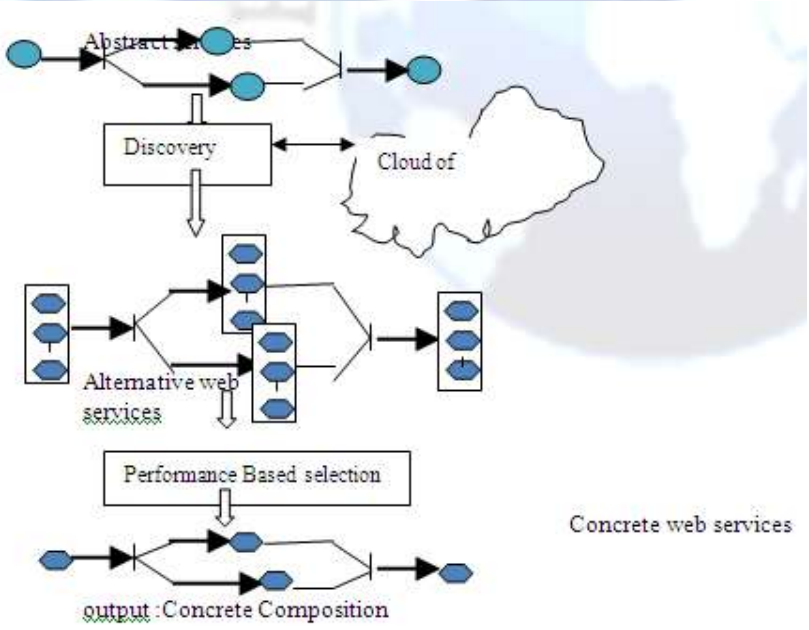


Fig. 03 : Conceptual overview of Web Services

B. SELECTING THE BEST WEB SERVICE:

Step-1: Construct matrix for the different performance parameter.
Step-2: Calculate the weight vector of these parameter
Step-3: Normalize the propose performance matrix

Step-4: Construct a weighted normalized performance matrix

Step-5:Calculate the final objective Function from weighted normalized performance matrix

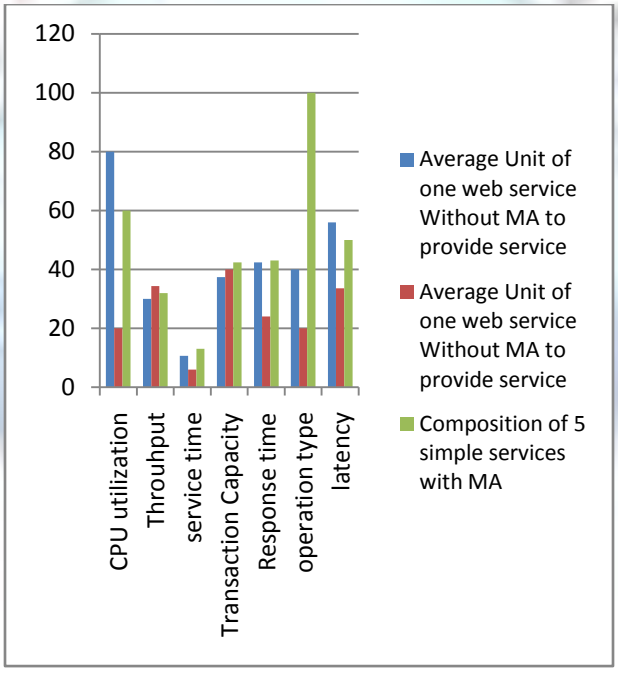
Step-6: Rank services in preference order

This is done by comparison of the values calculated in Step-5. Obviously, the Web service with smallest value $E^* = \min\{E_1, E_2, ..., E_n\}$ gives the closest match to the requester performance requirements and should be selected as the best one.

VI. CONCLUSION AND RESULTS

The performance parameters unit for the different web services is compared with and without mobile agents and found that there is improvement in the entire parameters unit except the throughput. Also the performance parameters units are compared when we are composing the composite web service using five simple web services and found that after the integration of web service with mobile agents there is improvement in all the parameters units.

.



REFERENCES

[1]. W3C Working Group, "Web Services Architecture," Feb. 2004.
[2]. AmnaEleyan ,Liping Zhao "Extending WSDL and UDDI with Quality Service Selection Criteria "Proceedings of the 3rd International Symposium on Web Services: 3rd International Symposium on Web Services; 21 Apr 2010-22 Apr 2010; Dubai. 2010.
[3]. Ramesh Nagappan , Sameer Tyagi, sun Microsystems "High Performance web services: tackling scalability & speed" Javaone :Sun's 2004 worldwide java developer conference
[4]. D. Gouscos, M. Kalikakis, and P. Georgiadis, " An Approach to Modeling Web Service QoS and Provision Price," in 4th International Conference on Web Information Systems Engineering Workshops (WISEW'03), Roma, Italy, December 13,2003, pp. 121-130.
[5]. Z. Chen, C. Liang-Tien, B. Silverajan, and L. Bu-Sung, "UX-An Architecture Providing QoS-Aware and Federated Support for UDDI," in Proceeding of the first International Conference on Web Services (ICWS03), Las Vegas, Nevada, USA, 2003.

[6]. H.-Y. J. Y.-J. Seo, and Y.-J. Song, "A Study on Web Services Selection Method Based on the Negotiation Through Quality Broker: A MAUT-based Approach " in First International Conference on Embedded Software and Systems (ICESS 2004), Hangzhou, China, 2004.

[7]. AmnaEleyan ,Liping Zhao ."Extending WSDL and UDDI with Quality Service Selection Criteria " Birzeit University, University of Manchester Palestine, United Kingdom

[8]. http://www.ibm.com/developerworks/webservices/library/ws-quality/index.html "Understanding quality of service for Web services"

[9]. Antonio Brogi, Sara Corfini "Behaviour–aware discovery of Web service compositions" International Journal of Web Services Research , Vol.5, No.7, 2010

[10]. Nitin Kumar Verma, Neeta Singh "A STUDY ON PERFORMANCE ANALYSIS OF WEB SERVICES" 2nd National Conference in Intelligent Computing & Communication, ISBN: 9788175157538

[11]. Hui Deng "Web service Enabled Mobile agent System" University of Hawai, phD thesis

[12]. Freddy L´ecu´e, Eduardo Silva, and Lu´ıs Ferreira Pires "A Framework for Dynamic Web Services Composition "France Telecom R&D, France

[13]. Mustafa Adaçal,andAy¸se B. Bener, "Mobile Web Services: A NewAgent-Based Framework "Published by the IEEE Computer Society 1089-7801/06/$20.00 © 2006 IEEE IEEE INTERNET COMPUTING

[14]. Velmurugan, K. and M.A.M. Mohamed "Study of Network Traffic Pattern and itsImpact on Performance in Implementation of Web Services" American J. of Engineering and Applied Sciences 5 (1): 63-69, 2012 ISSN 1941-7020 © 2012 Science Publications

[15]. http://nirajrules.wordpress.com/2009/09/17/measuring-performance-response-vs-latency-vs-throughput-vs-load-vs-scalability-vs-stress-vs-robustness/

[16]. Sunil R Dhore, Prof. Dr M U Kharat "QoS Based Web Services Composition usingAnt Colony Optimization: Mobile Agent Approach "International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 7, September 2012, ISSN : 2278 – 1021

[17]. Zibin Zheng, Yilei Zhang, and Michael R. Lyu "Distributed QoS Evaluation for Real-World Web Services"2010 IEEE International Conference on Web Services

[18]. Velmurugan, K. and M.A.M. Mohamed "A Study of Network Traffic Pattern and itsImpact on Performance in implementation of Web Services" American J. of Engineering and Applied Sciences 5 (1): 63-69, 2012 ISSN 1941-7020 © 2012 Science Publications

[19]. Yilei Zhang, Zibin Zheng, and Michael R. Lyu" WSExpress: A QoS-Aware Search Engine for Web Services" 2010 IEEE International Conference on Web Services, 978-0-7695-4128-0/10 $26.00 © 2010 IEEE.