Indonesian Journal of Electrical Engineering and Computer Science
Vol. 16, No. 3, December 2019, pp. 1407~1415
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v16.i3.pp1407-1415

Parallelism in web services: design of parallel XML parser for web services

Rohitkumar R. Wagdarikar, Sandhya P.
School of Computing Science & Engineering, Vellore Institute of Technology, India

Article Info

Article history:

Received Apr 26, 2019
Revised Jun 28, 2019
Accepted Jul 8, 2019

Keywords:

Multi-core Machine
OpenMP
Parallel XML parser for SOAP
Parallel XML parser for UDDI
Parallel XML parser for WSDL

ABSTRACT

A WS provides the communication between heterogeneous systems. While performing this operation, we need to focus on QoS of consumer, provider and registry directory. There will be some parameters like WS selection, prediction and rank these are parameters need to consider while QoS implementation in web services. While performing integration in web services we need to focus on QoS requirements regarding server and network performance. Performance of WS is related to locations i.e the network distance and the Internet connections between consumer and provider. There will be more QoS approach which works on consumers collected QoS data, based on this data system can predict the QoS of WS. Throughput and response time are the QoS of WS. In this paper, we have proposed parallel XML parser, by which we can parse UDDI, WSDL and SOAP XML files parallel by which it will improve the response time and throughput of WS.

Corresponding Author:

Rohitkumar R. Wagdarikar,
School of Computing Science & Engineering,
Vellore Institute of Technology,
Chennai, Tamil Nadu, India.
Email: rohit.wagadrikar@yahoo.com

1. INTRODUCTION

A web service provides the communication between heterogeneous systems. In this system consumer request for some services which is going to be provided by Service provider via service registry directory. Service provider first needs to register his services at service registry directory called UDDI, then services consumer search for the services at UDDI directory then the consumer will get the details to access these services from the service provider. While performing all these operation systems needs to focus on the quality of services of web service. The major requirements of Quality of web services are accessibility, availability, integrity, performance, reliability, regularity and security. These requirements will be fulfilled by SOAP, WSDL, and UDDI. In this paper, we are finding other QoS of web services which effects on quality and performance of the same. Web service selection can be done on a ranking based using data analysis also need to consider the QoS requirements regarding server and network performance. Web service Architecture as shown in Figure 1.

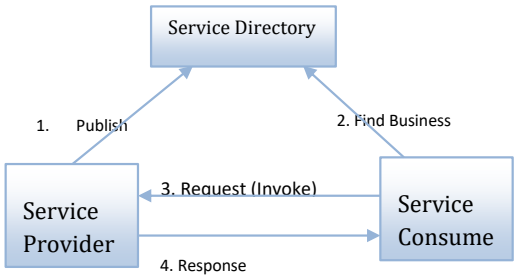


Figure 1. Web service Architecture

Following are the steps to invoke the service from the consumer by the provider. Service provider first publishes the details of business on UDDI directory, like firm name, service name, contact details etc.

a. Service Consumer search for the business at UUDI by SOAP request and response
b. In response from UDDI directory, the consumer will get the Business Key and Service Key
c. Consumer request for the service details at UDDI.
d. In response from UDDI, the consumer will get Access point and tModelkey details.
e. Consumer request for the tModel details that is the exact location of requested service interface details.
f. In response fromUDDI, the consumer will get the tModel details by Overview URL.
g. Then consumer invokes the service from service provider

Web service provides the feature of interoperability by using XML technology. By using XML technology different business process can communicate/ interact with each other. SOAP, WSDL, and UDDI are defined in XML only.

So, while performing these steps we need to focus on the different quality of services of web services which includes availability, accessibility, security, integrity, throughput, response time etc. to achieve these QoS parameters different authors are used different techniques and algorithms.

It should be noted that our proposed system address the throughput and response time of web services by performing parallel parsing of XML files, so proposed architecture concentrate on defining the architecture whose performs the parallel or concurrent execution of XML files. The remainder of this paper is outlined as follows. After discussion of existing system or literature survey, we generated comparison table which represents the different approaches used to represent the QoS of web service. In next section, we present the different multi-core architecture of system and then will represent different software which supports the development of parallel and concurrent programming, and then we present the architecture of our parallel approach for parsing of XML files of UDDI, SOAP, and WSDL. In the last section, we conclude with an outlook of future work.

2. LITERATURE SURVEY

As described by Anbazhangan Mani and Arun Nagarajan in [1] Understanding quality of service for Web services, improving the performance of your Web services, stated that Quality of services for the web services is important factor or properties like accessibility, availability, reliability, security, regulator, integrity etc. all these properties can be achieved by SOAP, WSDL, and UDDI standards. In this paper, Anbazhangan Mani and Arun Nagarajan mentioned the service proxy to find the response time of web service. This can be achieved by adding SOAP binding interface which can be used by the consumer to invoke the service.

Following are the steps to achieve this.

a. Generate the service proxy by WSDL.
b. Add code to clock time by using timer package and its start() method.
c. Recompile and modify proxy.
d. Implement client to invoke service.

As described by M. Tian, A Gramm, T. Naumovicz, J. Schiller in [2] A concept of QOS integration in web services, it enables QOS integration and also a selection of web services based on requirements of server and network performance. They have also presented how business process mapped onto communication QOS to QOS aware network. They have also mentioned how to get real-time information about server performance of assured services by giving QoS feedback. In this paper they have mentioned, Web Service Offerings Language (WSOL), Web Service Level Agreement (WSLA), Web Service Management Language (WSML) this approach neither support mapping of the higher layer onto network

layer nor consider the server performance. In this paper, they have mentioned architecture which is used to measure the server performance by placing information in SOAP header. This architecture also has WSB (web service broker) which place in between consumer and provider. The consumer first needs place the request at WSB then WSB will process for the request at UDDI. This WSB also gives the details of cheapest web service details while giving a response to the request by the consumer. In this paper, they have mentioned two implementation strategies of WSB. First is a local object running within an application and second is remote object running. So finally they concluded that it gives instant information about server performance and by this integration, architecture user can allow for dynamic selection of web services.

As described by Tajudeen Adeyemi Ajao, Safaai Deris, Isiaka Adekunle Obasa in [3] QoS-based Web Service Selection Using Filtering, Ranking and Selection Algorithm, they have proposed an algorithm for selecting the best web services for the consumer. The authors allow the consumer to specify the constraints about web services, and based on this constraint web service directory broker starts filtering the web services which are relevant to consumers request. On the service directory many web services are available for the same problem statement, so after applying this constraints broker will display top five web services which meet the constraints. So, finally, they concluded that by this filtering and ranking algorithm consumer can select best web services.

As described by Chengyuan Yu, Linpeng Huang in [4] A Web service QoS prediction approach based on time and location-aware collaborative filtering, stated that location, network distance and internet connection between user make the impact on the performance of web services. To find the solution they have separated the user set and service set into many clusters according to location details, then he tries to find similar user and services with the smaller cluster. so by this technique consumer can make a search in a cluster instead of an entire database. If any user or service is added or deleted from cluster then it will not affect on other clusters, just need to update the modified cluster. they have also mentioned that Memory-based collaborative filtering algorithm and Model-based collaborative filtering algorithm has a scalability problem. So by combing strength of this algorithm, they are addressing challenges like the high quality of prediction, high scalability and easy to build and update.

As described by R. Sarala, P. Manisha, Mukku Vineesha, G. Indumathy in [5] A Consumption History and QoS based Web Service Ranking Technique, stated that web services are ranked by functional significance, user behavior, QoS requirement and service usage factor. Here consumer will request for the service by functional significance and QoS query. Once consumer will get the list then keyword frequency used to find the functional score. This keyword matches with directory repository against consumers function and QoS query. Then this keyword is going to be stored in log files. Then any user who has searched for the same web service in the history and has the same score as the current user, based on this theory they make the rank of web service.

As described by Huan Liu, Farong Zhong, Bang OuYang in [6] A Web Services Selection Approach Based on Personalized QoS Prediction, the objective is to select the web service by predicting the quality of unused web service, By using collaborative filtering approach. The basic idea is to find out the similarity between different consumers with different QoS data and analysis provided data and make a prediction before invoking web service. In these paper authors analyzing the experience of other consumers QoS data of unused web services. This analysis will be performed by collaborative filtering. In this approach for the data normalization fuzzy set and a guassian algorithm is used. Then it will use a collaborative algorithm to find the similarity between consumers QoS data, then will decide for the selection. Finally, they have proved that selection approach based on personalized QoS is better than COR and average prediction method.

In [7] QoS-Aware Web Service Recommendation by Collaborative Filtering paper, authors Zibin Zheng, Hao Ma, Michael R. Lyu and Irwin King observing the past history of consumers with QoS data making the prediction based on collaborative filtering. Here combining the User-based PCC (person correlation coefficient) and item based PCC approach to predict the QoS values. In user based PCC author mapping user and web service item and creating a user-item matrix. In this matrix, each cell stores the details about service user, and in item-based PCC it creates the same matrix and stores the details regarding service item. So based on by combining user based PCC and item based PCC using collaborative approach consumer can predict the best QoS of web service.

3. SPEEDUP

According to Amdahl's law [1], execution time is improved if independent code/ tasks are divided into multiple processors and executed parallel / concurrently. In this paper, we are we are parsing those XML tags who do not have any child element and assigning it to multiple cores to execute concurrently in order to improve execution time.

As mentioned in [7] Speed up is performance metric which is given as:

$$T_{parallel} = \sigma(n) + \frac{\varphi(n)}{p} + \kappa(n, p)$$

Where:
n is the size of the problem
p is number of processors
σ (n) is the program's serial part execution time,
ϕ(n) is the program's parallel part execution time, and
κ(n, p) is the communication time.

Efficiency is the ratio of speed up obtained to the number of processors used [7]. It measures processors utilization. Parallel system efficiency of solving an n-size problem on P processors is given by

$$0 \le \varepsilon(n, p) \le \frac{\psi(n, p)}{p} \le 1$$

Our algorithm is designed in such a way that operations are performed in parallel/concurrent manner and there is no communication required among cores and no sequential execution.

Based on [8-10], parallel/concurrent execution is more efficient than sequential execution. According to the theory described by this authors, that is mentioned in [8-10] we have observed that performing concurrent/parallel execution is more efficient than sequential. In this paper, we have proposed parallel XML parser, by which we can parse the UDDI, WSDL, and SOAP XML files parallel by which it will improve the response time and throughput of web services.

4. PROPOSED WORK

As we have seen in literature survey section and in the comparison table, there will be many QoS those make the impact on the selection of web service and on execution web service. In this section, we are proposing the solution which makes an impact on throughput and response time of web service by performing parallelism in the parsing of XML files. As shown in Figure 2 Architecture of parallelism in web services.

This architecture is equivalent to web service architecture. In our proposed system we have introduced parallel XML parser, which performs the parallel parsing of the XML file. As we have already seen, three main components of web services are UDDI Registry, Service Requester and Service Provider, all these three components communicate with each other via XML file.

UDDI and service requester communicate with each other via SOAP XML file and UDDI registry and service provider communicate with each other via WSDL XML file, and Service requester and service provider communicate with each other via SOAP XML file. While interacting with each other, these three components parsing these file i.e reading these files sequentially. In our proposed system we have introduced parallel XML parser which performs the parallel parsing or reading of SOAP and WSDL files. This can be achieved by multithreading programming which is supported by OpenMP, JAVA, CUDA and many other programming languages.

Nowadays we have dual core, quad core, octa core etc many high-end configuration system, and all these systems support multithreading programming. Even NVIDIA has introduced GPGPU system. These systems are very high configured heterogeneous systems, which has millions of threads. So by using these high configured systems, we can perform the parallel and concurrent reading of XML files, which improves the performance of the web services. This improves throughput and response time of web service over the internet. In next section will see how UDDI XML file is going to be parsing. As describe authors in [11], UDDI is Universal Description, Discovery, and Integration discovery tool. It's a centralized system which provides the systematic way to find the services over the UDDI. It provides two specifications first are providing the information about service and how to encode it? And the second one is to how to access and update the information? It encodes three type of information, first, white pages which include name and contact details, second is, yellow pages, information about, business and service types, and third is, green pages, which includes, technical information. As shown in below Figure 3 the structure of UDDI XML file, which represents these three types of information. <businessEntity> tag represents white page details, <businessService> tag represents yellow pages and <bindingTemplate> <tmodelInstance> tag represents green page details. Tree representation of UDDI XML File as shown in Figure 4.

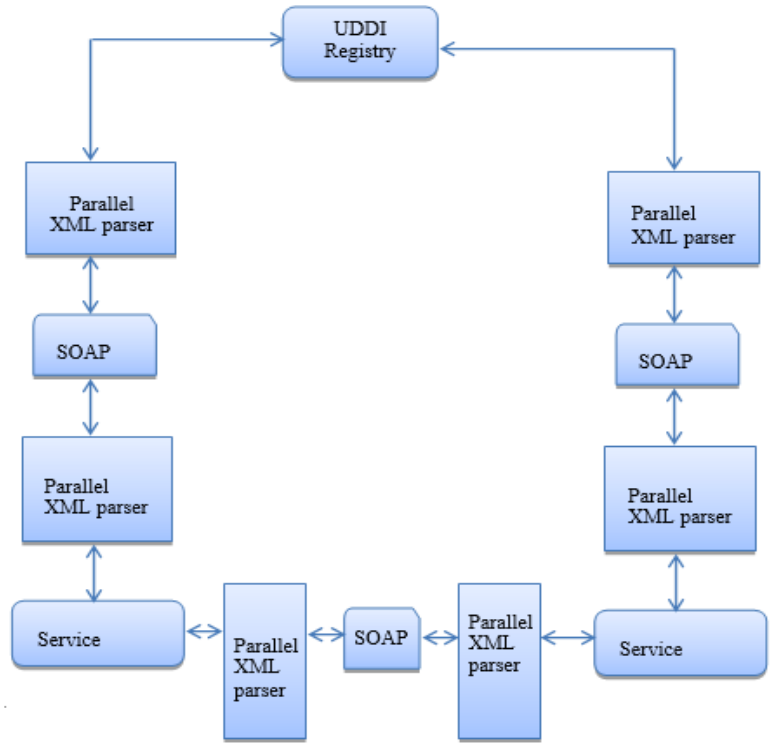
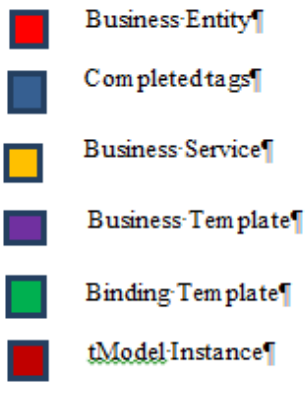


Figure 2. Architecture of Parallelism Web Services

4.1. UDDI XML File

Figure 5 parallel parsing of UDDI XML file with threads, in this figure we have proposed a diagram which shows the thread assignment to each node and its thread number. On multi-core machine we have N number of threads; we can utilize these threads for the parsing purpose.

```
<businessEntity..>
<name>.. </name>
<description>.. </description>
<Contacts>.. </contacts>
<businessservice>
<name>.. </name>
<description>.. </description>
<businesstemplate>
<bindingtemplate>
<description>.. </description>
<accesspoint>..</accesspoint>
<tmodelintstance>
<name>.. </name>
<description>.. </description>
<overviewdoc>
<description>.. </description>
<overviewurl></overviewurl>
</overviewdoc>
</tmodelintstance>
</bindingtemplate>
</businesstemplate>
<category>.. </category>
 </businessservice>
<category>.. </category>
</businessEntity>
```



Figrue 3. UDDI XML File Structure

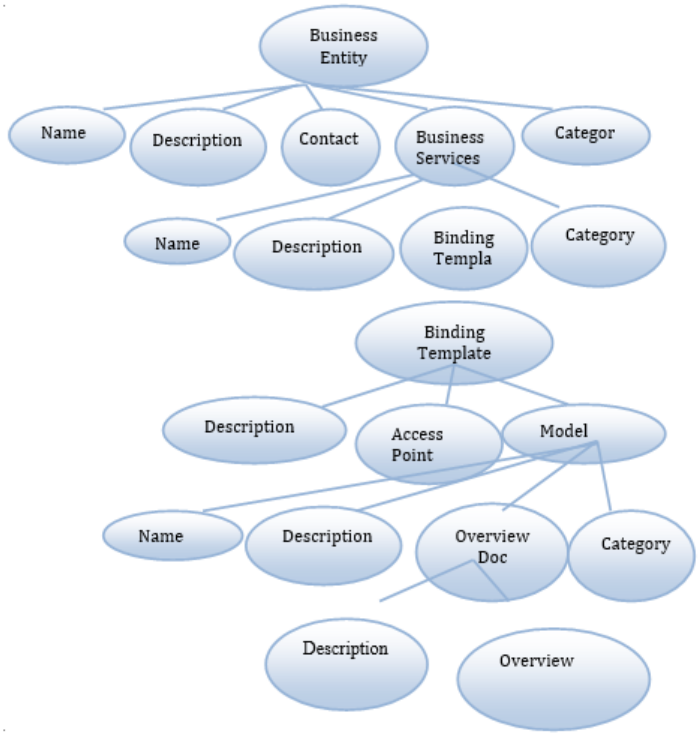


Figure 4. Tree representation of UDDI XML File

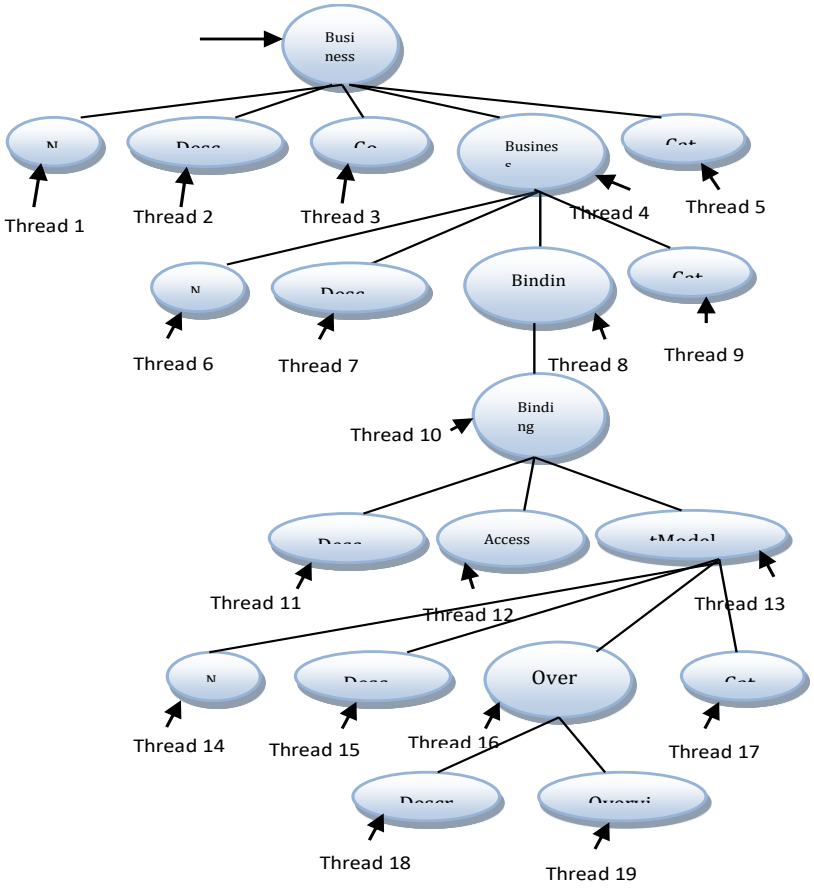


Figure 5. Parallel parsing of UDDI XML file with threads

4.2. WSDL XML File

SOAP is used for basic communication, whereas WSDL provides web service interface and contact details. As described in [11] WSDL describes two things, first, application level service interface and second is access protocol details which requester must follow to access the service. In first application level service interface, it defines three main components, 1 the vocabulary 2 the message 3 the interaction. As shown in Figure 6 WSDL XML file structure.

```
<definitions>
<types>
 data type definitions.....
</types>
<message>
 definition of the data being communicated....
</message>
<portType>
 set of operations.....
</portType>
<binding> protocol and data format specification...
</binding>
<Service>
Specifies the port address of the binding

 </Service></definitions>
```

Definitions¶
Types¶
Message¶
PortType¶
Binding¶
Service¶

Figrue 6. WSDL XML file structure

As shown in Figure 7 tree representation of WSDL XML file, this diagram represents the WSDL XML file in tree form for a better understanding of independent node in the XML file. Each node represents the XML tag. To perform parallelism or concurrent parsing of XML file, first need to find the independent node and here in Figure 7 by representing this file in tree format we can identify that <types>, <message>, <portType>, <binding> and <servcie> are the independent node, means we can parse this tag by concurrent or parallel manner. In this manner, we can find the independent node and we can perform parallel parsing of these files.

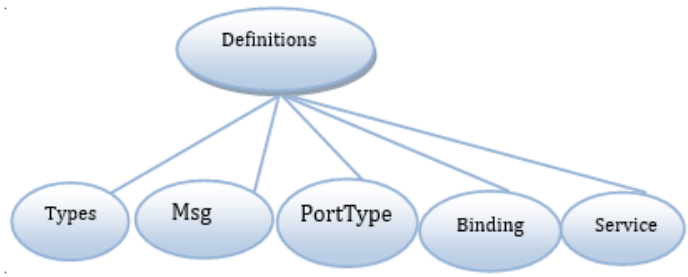


Figure 7. Tree representation of WSDL XML files structure

As shown in Figure 8 parallel parsing of WSDL XML file with threads, in this figure we have proposed a diagram which shows the thread assignment to each node and its thread number. On multi-core machine we have N number of threads; we can utilize these threads for the parsing purpose.

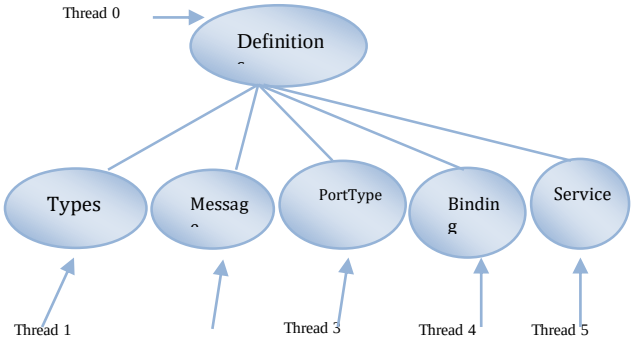


Figure 8. Parallel parsing of WSDL XML file with threads

4.3. SOAP XML File

SOAP is a simple Object Access Protocol; which provides the communication between web service provider, web service requestor and service directory. As describes in [12], SOAP has simple structure has three parts the first Envelope defines the framework for describing what about the message and how to process it. Second is set of encoding rules to define the data types and third is to represent remote procedure call and response. As shown in Figure 9, it will represent the SOAP XML file in a tree structure. In this figure all the node are dependent, so we cannot perform parallelism on SOAP XML file.

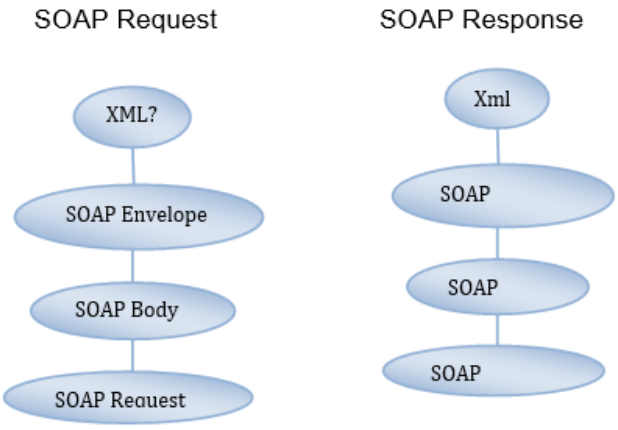


Figure 9. Tree representation of SOAP XML files structure

5. CONCLUSION

As we have seen communication between heterogeneous applications is possible by XML file in web services. As we have seen many authors have worked on the bandwidth of internet, network distance and Requirement of Server to improve the performance.

But still data from XML file is reading in a sequential manner, wherein nowadays all the application are executing on a multi-core machine which has N number of threads, so we can read this XML file in a parallel or concurrent manner by this threads, which improves the Response time and Throughput of web services.

REFERENCES

[1] Anbazhangan Mani, Arun Nagarajan. "Understanding quality of service for Web services". IBM on January 01, 2002.
[2] M. Tian, A Gramm, T, Naumovicz, J Schiller. "A concept of QOS integration in web services". IEEE,19 April 2004, ISBN: 0-7695-2103-7.
[3] Tajudeen Adeyemi Ajao, Safaai Deris, Isiaka Adekunle Obasa. "QoS-based Web Service Selection Using Filtering, Ranking and Selection Algorithm". *IJSCER*, Volume 4, Issue 7, July-2013.

[4] Chengyuan Yu, Linpeng Huang. "A Web service QoS prediction approach based on time and location-aware collaborative filtering". *Springer*, June 2016, Volume 10, Issue 2.
[5] R Sarala, P Manisha, Mukku Vineesha, G Indumathy. "A Consumption History and QoS based Web Service Ranking Technique". *ICEIET*-2017, Issue ICEIET'17, Vol:2.
[6] Huan Liu, Farong Zhong, Bang OuYang. "A Web Services Selection Approach Based on Personalized QoS Prediction". *IEEE*, 22 December 2011, ISSN: 2379-5352.
[7] Zibin Zheng, Hao Ma, Michael R Lyu, Irwin King. "QoS-Aware Web Service Recommendation by Collaborative Filtering", *IEEE*, VOL. 4, NO. 2, APRIL-JUNE 2011.
[8] Michael J. Quinn, "Parallel Programming in C with MPI and Open MP", McGraw-Hill Education (2003) ISBN 10: 0070582017 ISBN 13: 9780070582019
[9] Paul Edmon, "Introduction to OpenMP: HARVARD Faculty of Arts and Sciences", ITC Research computing Associate. Section
[10] Mark D. Hill, Michael R. Marty, "Amdahl's Law in the Multicore Era", *IEEE Computer Society* July 2008
[11] Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi, Sanjiva Weerawarana IBM T.J.Watson Research Center. Unraveling the Web Services Web An Introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing* (Volume: 6, Issue: 2, March-April 2002).
[12] Don Box, DevelopMentor, David Ehnebuske, IBM, Gopal Kakivaya, Microsoft, Andrew Layman, Microsoft, Noah Mendelsohn, Lotus Development Corp., Henrik Frystyk Nielsen, Microsoft, Satish Thatte, Microsoft, Dave Winer, UserLand Software, *Inc. Simple Object Access Protocol (SOAP)* 1.1. W3C Note 08 May 2000.
[13] Erik Christensen, Microsoft, Francisco Curbera, IBM Research, Greg Meredith, Microsoft, Sanjiva Weerawarana, IBM Research, "Web Services Description Language (WSDL) 1.1", W3C Note 15 March 2001'