A Combination Approach for Improvement Web Service Performance

Seyyed Hasan Aali and Roya Zareh Farkhady

***Abstract*— Web services are essentially application that can call, distribution and description through internet and intranet. In early future, the web will convert from set of information to set of distributed system and determine future of distributed system. Web service is suitable for distributed system because it use SOAP protocol. SOAP a simple and extensible for exchange of message, is the most widely used communication protocol in the web service. Beside of advantage, web services have basic problem and that is low performance. Some of approach introduced to improve performance but the problem of web service didn't solve perfectly and every solution solved part of problem.**

In this paper, we present a combining approach that is a combination of compression and caching. In this solution, with compression of SOAP message the transfer time reduce likewise with caching in server side, the response time reduce too. We implement and evaluate this approach. The result of evaluating this approach, shows the total time of SOAP message processing and transfer time in media is improved.

***Index Terms*— zipping, web services, performance, caching**

I. Introduction

Currently, a new approach have been created based on web service, with development and standardization of web service protocol such as SOAP, XML and WSDL. In a web service, describing, registering and searching of the service is based on XML and in order to use of XML, interoperable problem in heterogeneous distributed system has been solved.

Web services are independent from software and hardware platform and they are very interoperable in comparsion with other distributed models such as CORBA, DCOM and RMI. But low performance is a basic problem for the web services. The performance of distributed system depends on the format of data transferred in media. The traditional client/server communication paradigms such as RPC or RMI offer high performance, but the problem of the Remote Processing Call is the use of a certain protocol. For example in the RMI technology, client and server program must be implement java and these properties lead to creation tightly coupled system.

Manuscript received December 30, 2010; revised January 17, 2011.

Seyyed Hasan Aali Author is with the Computer Engineering Department, Islamic Azad University Of khodafarin, Iran, (e-mail: seyyed.hasan.aali@gmail.com)

Roya Zareh Farkhady Author is with the Computer Engineering Department, Islamic Azad University Of khodafarin, Iran, (e-mail: roya_farkhady@yahoo.com).

Because of web service use XML as a exchange of message, interoperability of heterogeneous system is increased But conversion of data to XML format leads to increase amount of data and decrease performance.

The reset of this paper is organized as follows: first, we survey the related works in section 2. In section 3 we describe our approach with implementation of details. In sections 4 the result of experiment is shown. At last in section 5 we have conclusions and future works.

II. Related works

A variety of approaches have been proposed for improving web service performance. Cache is a classic but effective approach. Especially in implementing web services, the use of cache mechanism results in reducing response time and increasing performance.

Two different architecture can be defined for caching web services. The first one has two levels: server and client. The second one has three levels that involve two types of architecture. First type has three level homogeneous caching architecture (HOCA). All three entities in the architecture use the same protocol [1][2]. For example, a web proxy, a web client and a web server is a good example of HOCA. Second type has three level heterogeneous caching architecture.(HECA). In this architecture, two different protocol are used. For example, consider a fat client invoking a remote component (using DCOM or EJB, for example). Let us suppose that the remote component invoke a web service; here we have there entity. The remote component and the web service communicate using a protocol different to the one used between the fat client and the remote component [3].

In [4] a new approach has been presented for optimizing performance. This approach uses file transfer protocol to optimize data transferring through network. According to accomplished evaluation, the use of this mechanism reduces the time of exchanging of data, more than 70%. In addition it can efficiently send data through network without losing standards of data transfer and traffic from http protocol is removed.

Another approach for improving performance is data bind. Data binding means the mapping between XML text and object, which can be divided into two types. Late binding uses java reflection mechanism to do the mapping, but java reflection is not efficient, for this reason, late binding is regarded as a bottleneck of web services.

Early data binding creates template class that record java object properties and method information before runtime,

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

and use these template classes for doing mapping. Early data binding can avoid java reflection at runtime, so it is more efficient than late data binding. In [1] with use of early data binding, the template classes embedded into SOAP processor and when SOAP message flow into it, SOAP processor uses these template classes to create java object. In these binding types it can't be added a new mapping between XML data and java data in runtime and it causes that flexibility in contrast to late data binding decreases.

In [5] another approach is presented with the name of dynamic early binding, extended by context free grammer and implemented by pushdown automation. According to accomplished evaluation this approach is better than early binding and late binding.

Another approaches are the differential serialization and differential deserialization. The idea in differential serialization is avoided complete serialization. The serialization is performed as a complete certain structure for a message, then the structure of this message is saved. For subsequent message with the same structure and almost the same content can be reused the saved templates. This approach can be implemented in client and server sides[6].

The main idea in the differential deserialization is that it should be deserialized only just area that has not been processed before, and reused the object that has been deserialized in the previous stage. Therefore the set of processes for complete de serialization are reduced [7][8].

The performance of deferential deserialization is better than deferential serialization because differential serialization work only if the same client send a stream of similar message but deferential deserialization, on the other hand can accomplish deserialization of similar message sent by multiple different clients.

III. Proposed approach and implementation

The abbreviation nation for proposed approach is CSMC that is a combination of two different solutions. According to accomplished evaluation, it improves performance. This solution is a combination of classic caching approach and compression of SOAP message in order to decrease amount of sent data.

For a web service to be implemented CSMC approach, following facts must be true:

1) The number of request from web service should be large enough.
2) There is at least one method with the name of cache method that response to the repeated requests in certain time without any changing in SOAP response.
3) This approach will have better efficiency if the size of SOAP message is large.

CSMC approach can be used for different applications in real life. Some of examples of these applications are:

1) A web service that receives the important world events from valid news agency and displays them.
2) A web service for obtaining price and information of share in capital markets. Although this is a typical real time application, and clients should obtain real price in every request, but running of the service can be considered when the stock market has been closed.

If the size of SOAP message is large, the sending of data will be difficult because of limited bandwidth. In presented approach before sending SOAP message in media, message should be zipped in order to decrease transfer time. An important point in zipping of SOAP message is that a whole of SOAP message don't zip and only the body of SOAP message zips because the header of SOAP message is required for correct function.

The main stages for creation of SOAP message in client side and its sending to other side to continue the procedure of work in proposed approach are follows:

1) Serializing in client and creating SOAP message.
2) Detecting body of SOAP message and zipping it and replacing with original body.
3) Transferring Zipped SOAP message on media.
4) Unzipping zipped message in server side after receiving the zipped SOAP message.
5) Deserializing requested SOAP message to obtain objects and parameters.
6) If requested result exists in the cache, after zipping the response SOAP message it will be sent to client via media else after executing web service, first the response SOAP message is created and then is saved in cache and after zipping it is sent to client via media.
7) unzipping in client side after receiving the zipped SOAP message
8) Deserializing the unzipped message and using of results in client side

The architecture proposed approach that is based on presented plan in [4] is shown in figure 1.

IV. Experimental results

According to accomplished evaluations it could be claimed the combining of zipping and caching is best combination because in client and server communication there is not problem in view of processing speed and the basic problem is transferring speed in media. Thereby the use of zipping message the size of message can be decreased about 90%. For example a message with size of 100KB, after zipping has 10KB that requires less time to transferring in media. The use of caching mechanism in server side shows appropriate usage of central processing unit. With caching mechanism there isn't required extra processing and SOAP message creation and can be used the SOAP message has been created in previous stage.

NetMon software is used for evaluating and analyzing proposed approach. With the use of this software, the difference between the time of last packet of requested SOAP message and the time of first packet of response SOAP message is the time taken for processing of web service and the time difference between first and last packet of response SOAP message is the time taken for transferring on media.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

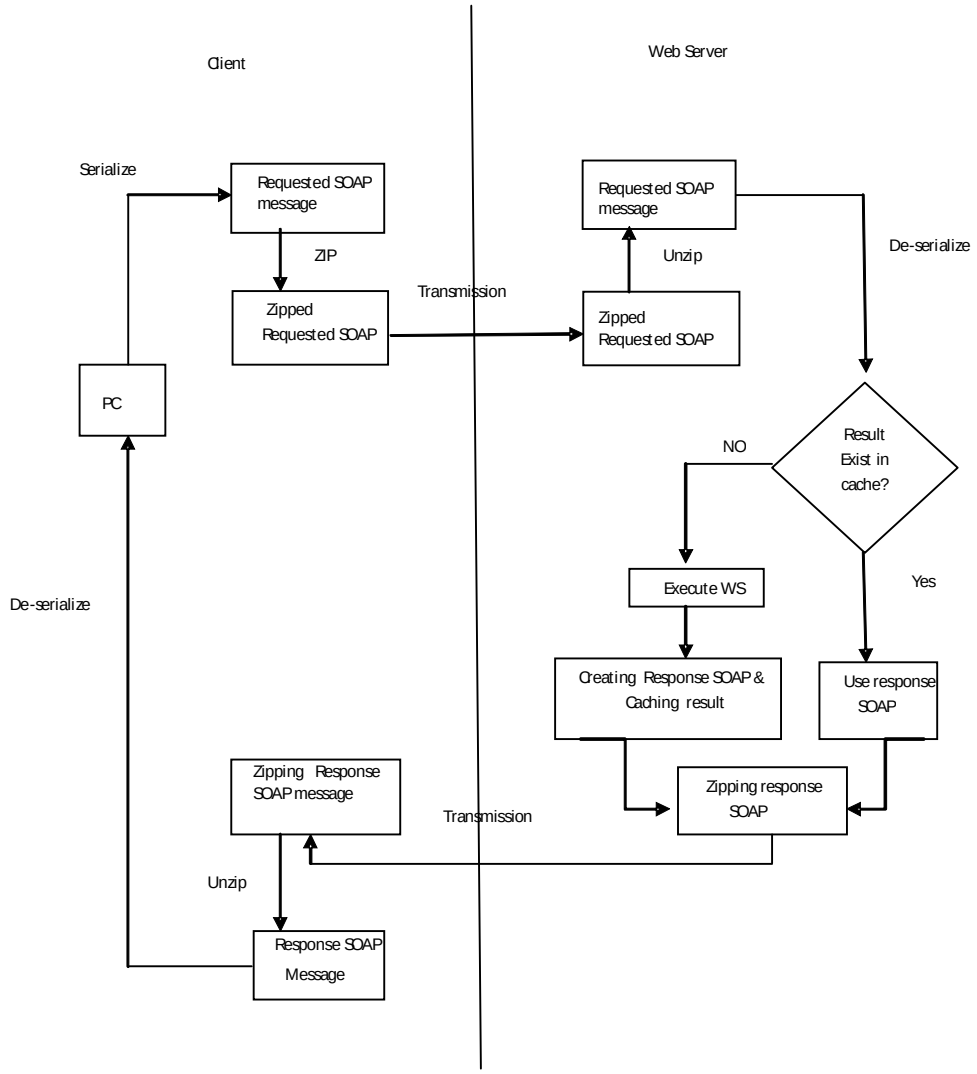

Fig. 1.Architectue of proposed approach

The experiment is done by two computers with 3.2GHz CPU and 1CB Ram. Two computer are connected to each other with a hub that supported 10 mbps. Windows server 2000 and IIS have been installed on the first computer that it play web server rule. Then, NetMon software is installed and run on web server. In order to be analyzed the network traffic. Now, we consume web service through second computer with XP operating system.. In this time a one SOAP message is sent from client to server. When client received response of SOAP message, we should stop running of NetMon software.

The web service returns records of pubs database. The experimental is done many times and the results is registered to processing later.

Table I shows the time needed for transferring of data on media in two modes: Common mode, CSMC mode.

According to figure2 it is obvious that by CSMC approach, time needed for transferring of data on media is decreased.

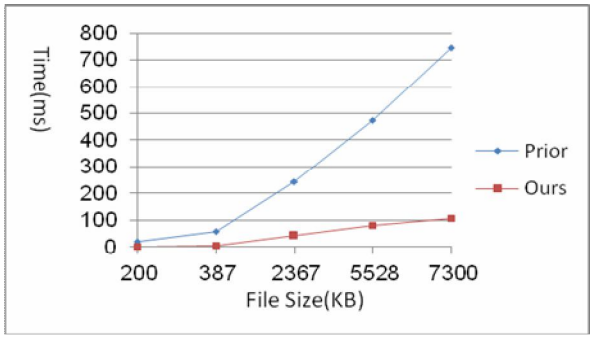

Fig. 2.Transferring time SOAP message in media

TABLE I
TRANSFERRING TIME SOAP MESSAGE IN MEDIA

File size	Transfer Time(ms)	
	WS	CSMC
200	20	2
387	57	6
2367	242	83
5528	473	83
7300	747	107

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE II
PROCESSING TIME OF SOAP MESSAGE

File size	Process Time(ms)	
	WS	CSMC
200	109	129
387	198	227
2367	778	813
5528	1441	1217
7300	1762	1400

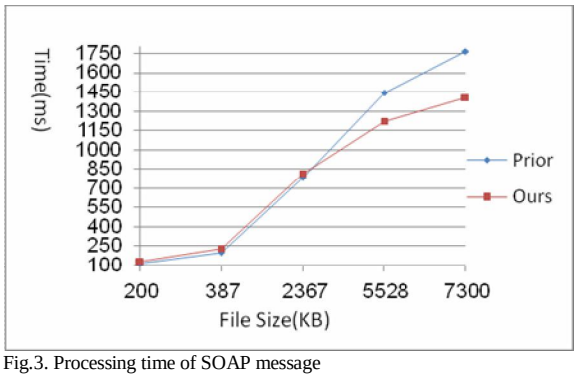

Fig.3. Processing time of SOAP message

TABLE III
PERCENTAGE OF INCREASING PERFORMANCE

File size	Total time =Process Time + Transfer Time (ms)		Performance
	WS	CSMC	%
200	129	131	-0.01
387	255	233	0.09
2367	1020	854	0.19
5528	1914	1300	0.32
7300	2509	1507	0.39

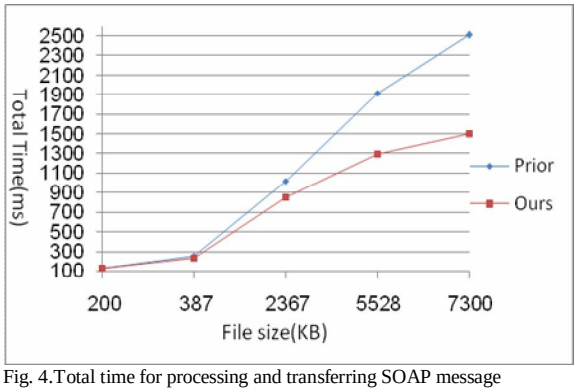

Fig. 4.Total time for processing and transferring SOAP message

Figure 3 and table II show the processing time in common mode and CSMC mode. Finally figure 4 and table III show the amount of increasing web service performance by CSMC approach.

V. CONCLUSION AND FUTURE WORKS

As mentioned, with this approach performance is increased when the size of SOAP message is large, while performance is decreased when the size of SOAP message is small. Also, in this approach, if the different clients frequently invoke a web method with the same parameter, it will be avoided to be created of SOAP message because of use of caching mechanism, and this result in increasing of performance.

REFERENCES

[1] J. Fernandez, A. Fernandez and J.Pazos, "High availability wit clusters of Web services", Proceedings of the 16th APWeb, April 2004. pp. 644-653

[2] J. Fernandez, A. Fernandez and J. Pazos, "An architecture for building Web services with Quality-of-service features", Proceedings of the 15th WAIM, July 2004.

[3] J. Fernandez, A. Fernandez and J. Pazos, "Optimizing web services Performance using Caching", Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'05), 2005.

[4] T. Wichaiwong, C. Jaruskulchia, "A Simple Approach to Optimize Web Services' Performance", Proceedings of the 3rd International Conference on Next Generation Web Services Practices,2007.

[5] W. Jun, H. Lie and N. Chunlei, "Speed-up SOAP processing by Data Mapping Template", Proceedings of the IW-SOSE'06, 2006.

[6] N. Abu-Ghazaleh, M. J. Lewis, "Differential Serialization for Optimized SOAP Performance", Proceedings of the 13th IEEE International Symposium on High-Performance Distributed Computing (HPDC'04), 2004.

[7] N. Abu-Ghazaleh, M. J. Lewis, "Differential Deserialization for Optimized SOAP Performance", Proceedings of the 2005 ACM/IEEE SC'05 Conference (SC'05), 2005.

[8] T. Suzumura, T. Takase, and M. Tatsubori, "Optimizing Web Service Performance by Differential Deseralization", Proceedings of the IEEE International Conference on Web Services, 2005

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)