International Journal of Mechatronics, Electrical and Computer Technology

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

A Method for Performance Evaluation of SOAP Protocol

Hamidreza Zolfi

Islamic Azad University of Iran, Sari Beranch

*Corresponding Author's Email: *hamidrezazolfi2010@ymail.com*

Abstract

SOAP has revealed as practical protocol for web service to increase a cooperative capability. SOAP is based on XML thereby SOAP's big massage inherit some merits demerits of XML while different transactions apply for simultaneous operations from server using trade SOAP unicast to send response message can create large traffic. Also processing calculations of XML is time consuming. First we introduce HTTP/1.1 protocol as an underlying protocol. We use SMP protocol as a SOAP multicasting protocol for reducing network traffic also we use server side caching to decrease response time.

Keywords: SOAP, Message Multicasting, SOAP performance, SMP, Caching

1. Introduction

Technologies of web service like SOAP, WSDL, UUDDI are provided for web services integration that are used for different industrial, language and various platforms. RPC and RMI provide powerful models for distributed system programs. These program models including: one protocol for exchanging data, one language to describe application interface and a connection that protect the way of connecting computer inter program. SOAP and WSDL are protocol and interface language that provide the above cases for Web services and Web application in point to point systems [3] [4]. These technologies can support web service underlay that available services by WSDL described and discovered through services. One of the big problem of web services creating network heavy traffic. The factor of the problem can be web services based on SOAP and communicative protocol based on XML. SOAP provides underlying exchange of basic

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

communication for web service. SOAP overhead caused by XML usage. since SOAP and WSDL is based on XML and XML messages should be analysed in two side of client and server.XML analysed is perform runtime .therefore, this requires high processing time and as a results high response time. In [5] different limited cases for SOAP protocol investigated. SOAP transfer underlying is HTTP protocol. For HTTP has presented different models. One of the applied models is HTTP/1.1. HTTP/1.1 support characteristics that is efficient in performance improvement and transferring quality. Often for decreasing network traffic used multicasting protocols. Based on our researches, SMP is the first and the most efficient SOAP multicasting protocol.

This protocol able to obtain a 70 percent performance gain for large network. One of the most efficient method for decreasing response time, using cache method . the study showed that for SOAP protocol client side caching improving 800 percent and for server side caching improvement 250 percent about response time. in this research we use SOAP server side caching that in addition , we can decrease traffic and response time on SOAP.

1.1 Related work

In [5] a set of performance limitation SOAP investigating scientific calculations and discussing SOAP execution for systems with memory requirements and hard bandwidth. in [22] new approach presented in SOAP multicasting called SMP . This method increase web service performance using semantic similarity. In this method, instead of generating many messages with duplicated parts for different clients, the duplicated parts are reused for multiple clients, thus reducing network traffic. In [14] executed speeds SOAP as a limitation have been studied that SOAP request process sent toward server, inward client by cached client's movement has been improved. This method caused to improve 800 percent affiance, in [13] also to improve performance of protocol SOAP from method cached has been used. in this study, previous limitation of action [14] posed. Since

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

client's cases posed in server part, therefore, it used of method cached in server side. This method made improves 250 Percent of performance.

2. Backgrounds

In this part, we investigate the required cases of the research. The cases are as follows: SMP, HTTP/1.1 and server side caching. HTTP/1.1 is clarified as an underlying exchange of SOAP message. In channels that similar messages are sent, similar message unicast caused high traffic. The use of SMP multicasting can make improve redundancy. On the other hand, calculation arising of SMP message construction caused increase response time. The server side caching method can be efficient in showering down the calculated time.

2.1 HTTP/1.1

Different models introduced for HTTP are as follows: 0.9, 1.0, and 1.1. One of the studied [38] showed that prior to the introduction HTTP/1.1 modal 1.0 that concluded about 75%internet body. However, changes conducted in improved models 1.1 rather than previous models. That is divided into 9 sections: extensibility of server side catching, band width improvement, communication network management , message sending , protection of internet address , error declare, security, content exchange.therefore,HTTP1.1 protect characteristics about server side catching and band width improvement that presenting underlying case with efficient quality than previous models.

2.2 SMP

SOAP messages that sent from one server to few clients have high similarity. SOAP messages created with the same execution has similar message structure. So, sending similar messages with a similar structure in network creates redundancy. One of ways of decreasing traffic is using multicasting protocol. The first and the most efficient protocol of SOAP similarity based multicasting, is the SMP. In networks that sent similar response

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

messages, the performance of this protocol is higher. similarity of messages is based on semantic structure of XML and recognized, then similar messages is sent in a framework of SMP aggregate message for following clients, the structure of SMP message in figure 1 has been shown[2].clients addresses saved titled as SMP string that in a body of SOAP message encapsulated. SMP's body inserted in the SOAP's body. The inner part of SMP has two sections. The common part involves common parts of SOAP messages and distinctive part involves specific part and each one involves SOAP response messages. In figure 2, a model of inner components of SMP protocol presented underlying mode HTTP1.1 SMP. SMP detector recognized that received message is a common SOAP message or a SMP message.

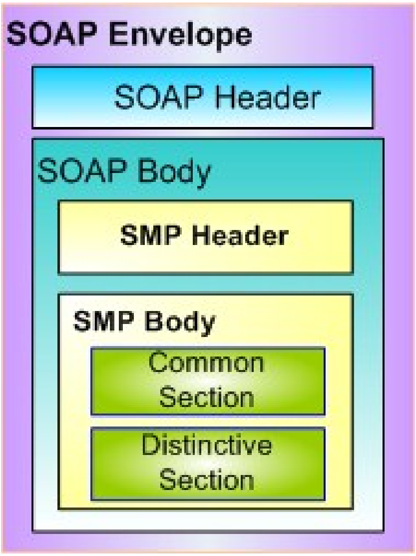

Figure 1: SMP message structure

If one message is SMP message, it should send for controller of message, otherwise is sent application layer and failed from SMP layer. When one message received by message handler, is analysed by splitting algorithm until it is necessary for next router to change sending. Routing adaptor is the responsible for routing table searchers that it finds another hop for present clients of SMP.

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

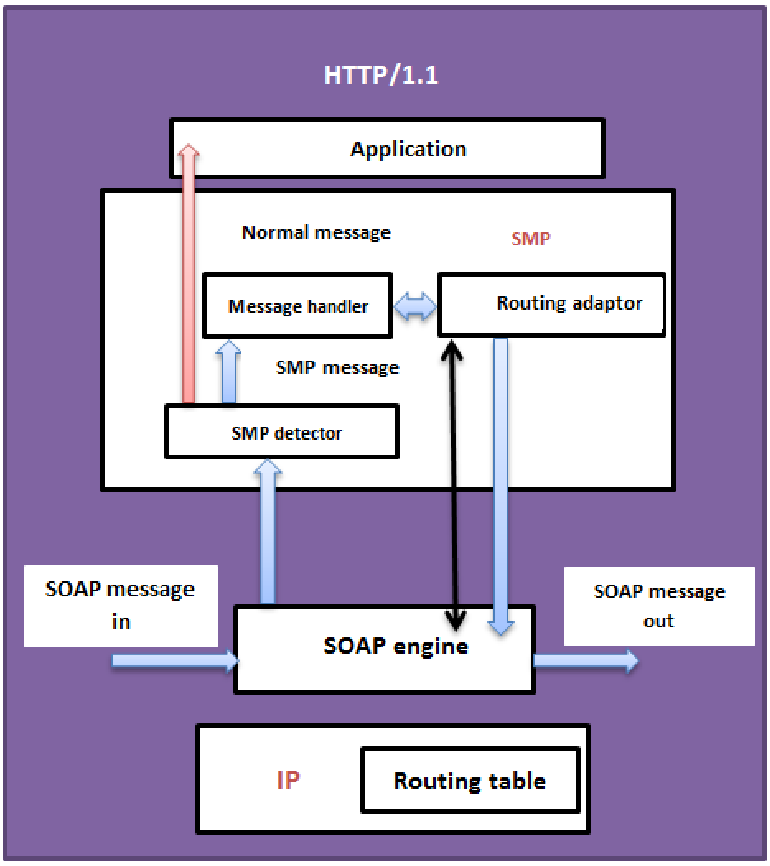

Figure 2: SMP component

2.3 Server side caching

One of the most expensive parts in the process of receiving application and sending SOAP response is the execution of processes such as coding XML and organizing messages. On the other hand, if different server's receiver of similar server from apply server, response server creates similar response messages and it should process for each one undoubtedly, performing additional processes causes to decrease calculated operations and increasing response time. To decrease extra work, it can use one cached server until client presented similar responses of server side: complete caching, body caching, body and packet caching (template).

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

3. CSMP Solutions

A lot of studies have been conducted for solving traffic problem and the response time of networks based on SOAP. According to studies, the first similarity based SOAP multicasting is SMP. The protocol has been represented for decreasing the general traffic of network [2]. To decrease response time cached methods has been represented so that we used server side caching in this paper [13]. In the proposal Cached Similarity based Multicasting Protocol (CSMP), to decrease response time resulting calculation creating SMP message, used SMP server side caching. In figure 3 we presented general design of CSMP structure. The basis of CSMP protocol is message similarity. Therefore, the protocol in networks that similar response messages send for client, represented performance. A set of client sent SOAP application for the following server. Messages, after finding route, delivered to server (it is assumed that server sends application in the same time). When application SOAP message received, a set of object entitled SOAP existed. It can be used the objects as message or client ID.

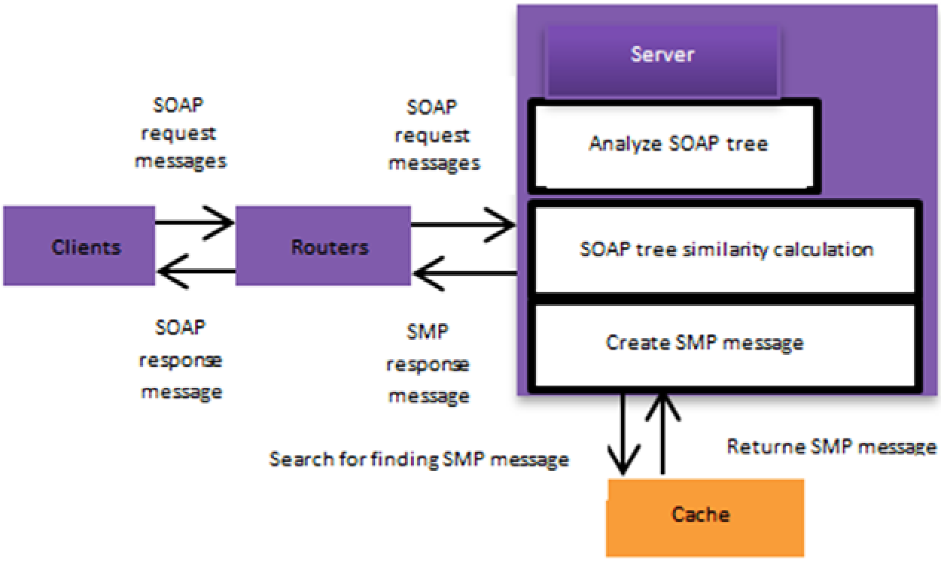

Figure 3: CSMP structure

The ID should be for send messages from client individually. To be individually clients, we use IP address and for being specific sent messages from client we use sent time as

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

ID. When in a time frame work t_{wait} , received application by clients, before creating SMP response, investigated cached contents. In case of in case of response message presence. The SMP message sent for clients needed the response and corrected its title. When there is not SMP message in cache, one SMP response is created. Then, the message for following usage cached used. In the caching method, after the available data mass cached reached to a certain amount, it should use available constituent policies for new data substituent data with available data cached. In CSMP protocol, we use LRU policy for substituting new messages. In the method, SMP message has been used the last recently and selected for deleting and changing a new SMP message.

4. CSMP performance evaluation

In network systems, often the performance investigated in three categories: response time, network traffic and throughput [15]. Response time (delay or end to end response time) is the time perceived by a client to obtain a reply for a request for a web service [16]. The total network traffic for a communication scheme or session consists of the total size of all session-related messages sent over the network for the duration of the communication. The throughput, which is defined as the number of requests per unit of time. In recent years, different operation conducted on web service performance [17-20] that often concentrated on processing and substituting SOAP messages as an effective agent on performance.

4.1 Performance measurement

Network traffic

It is assumed that SMP routing tree (H) is a spanning tree whit L level. S is a root of H and we have N client. If K is the number of messages, S_k is the common part of SOAP messages and $1-S_k$ is the distinctive parts of each messages. It can calculate SMP aggregated message size as follows:

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

$$aggregate\_size\{k\} = (S_k + k(1 - S_k)) \times msg\_size = (k - (k-1)S_k) \times msg\_size \quad (1)$$

We assume that message size coordinated between $optSize_{min}$ and $optSize_{max}$. The mean size of the response messages as the following formula calculated:

$$msg\_size = \frac{w_1 \times optSize_{min} + w_2 \times optSize_{max}}{w_1 + w_2} \quad (2)$$

the average of client in a SMP message is $\left\lfloor \frac{N}{n(d)} \right\rfloor$ and $n(d)$ is the number of d level nodes inn H tree. Sum of sent traffic of a line in d level from SMP tree calculated as follows:

$$aggregate\_size\left\{\frac{N}{n(d)}\right\} = \left(\left\lfloor \frac{N}{n(d)} \right\rfloor - \left(\left\lfloor \frac{N}{n(d)} \right\rfloor - 1\right) S_{\frac{N}{n(d)}}\right) \times msg\_size \quad (3)$$

Therefore, general network traffic calculated in a routing tree as follows:

$$Traffic\{CSMP\} = \begin{cases} \sum_{d=1}^{l} n(d) \times aggregate\_size\left\{\frac{N}{n(d)}\right\}, & if\ n(d) < N \\ \sum_{d=1}^{l} \sum_{i=1}^{k} N_i \times size(opt_i), & if\ \ n(d) = N \end{cases} \quad (4)$$

Response time

The response time experienced by an individual client is composed of the delay to convey the response message in the network plus any overhead at the server and at intermediate routers if they exist. Overall delay also including sum of transferring delay and propagation delay in source route to client. The transmission time, denoted by t_{trans}, of a message passing through a link is computed by dividing the size of the message, size(opt_i), by the link bandwidth. It is assumed that the bandwidth on each link in the network is the same and is equal to blink. The required response time of each client calculated by this formula:

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

$$response(c_n) = t_{overhead}(c_n) + delay(c_n) = t_{overhead}(c_n) + \sum_{e \in p_{s,c_n}} (t_{prop} + t_{trans}(e)) = t_{overhead(c_n)} + \sum_{e \in p_{s,c_n}} \left(t_{prop} + \frac{size(opt_i)}{b_{link}}\right) \quad (5)$$

In CSMP, there are overheads such as server processing overhead, cache overhead and intermediary routers overhead:

$$t_{overhead}(c_n)\{CSMP\} = t_{procServer}(CSMP) + t_{cache} + L \times t_{procRouter}(CSMP) \quad (6)$$

That we have:

$$t_{cache} = t_{cacheLT} + t_{explore} + t_{call} \quad (7)$$

$t_{cacheLT}$ is the required time to lookup marshaling data for assumed response object. $t_{explore}$ is the required time for search response in cache and t_{call} is required time for calling cache and servers. Overall transmission time of CSMP is calculated by overall transfer traffic division by network bandwidth:

$$t_{trans}\{CSMP\} = \frac{1}{N \times b_{link}} \begin{cases} \sum_{d=1}^{l} \left\lfloor \frac{N}{n(d)} \right\rfloor \times n(d) \times aggregate\_size\left\{\frac{N}{n(d)}\right\}, & if\ n(d) < N \\ \sum_{d=1}^{l} \sum_{i=1}^{k} N_i \times size(opt_i), & if\ \ n(d) = N \end{cases} \quad (8)$$

Combining Equations 5, 6 and 7, the formula for computing SMP's average response time is given by the following:

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

$$Average\ Response\ Time\{CSMP\} = t_{procServer}(CSMP) + t_{cache} + L\big(t_{procRouter}(csmp) + t_{prop}\big) + t_{trans}(CSMP) \quad (9)$$

4-2 Experimental result

We use a simple model of sending / receiving that messages sent as the group from server, routed whit router and received by client. Therefore, network traffic by calculation of all SOAP messages size on line obtained. To test the mentioned data, a set of 500 SOAP messages with average size 0.5 Kbyte based on SOAP web server of Google created. The messages created by IBM XML document. The result of comparative showed through network traffic in figure 4. As shown message in unicast method caused to create high network traffic mass due to repetitive similar message sending. CSMP and SMP methods as a result of similar messages omission, decreased the network traffic considerably. In high number of message, the traffic of CSMP rather than SMP decreased a little due to omission of traffic in inner server part.

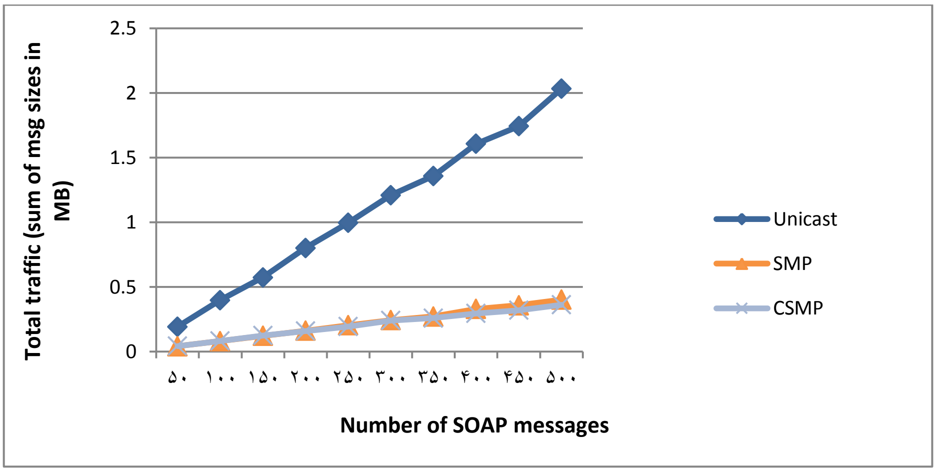

Figure 4: comparative network traffic diagram

We performed response time test on system with processor of Intel Xeon 2.66GHz with 4GB RAM. The comparative result of time shown in figure 5. SMP creates the most

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

response time. The suggested CSMP model prevent repetitive similar response messages create due to use of present response messages in cache. Therefore, calculated time for creating SMP response messages required is omitted. Whit spent the time, the calculated time between two SMP and CSMP protocol increased and this is due to different copies cached.

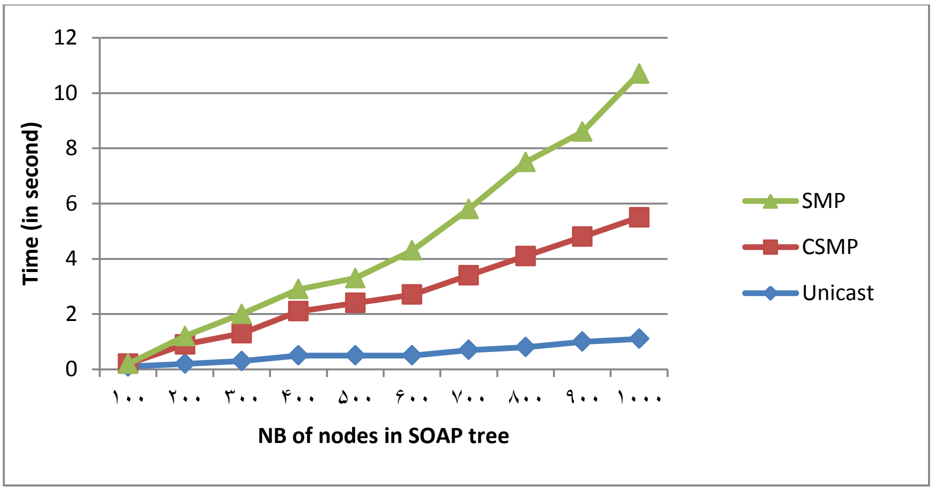

Figure 5: comparative time diagram

Conclusion and future work

In this paper, we investigated SOAP performance improvement methods. Two important limitation of SOAP, high traffic and high response time investigated. To decrease network traffic SMP selected but, calculation of SMP messages reconstruction made increase response time. To solve the problem, we used server side caching. Finally, we presented CSMP protocol. The results showed that the protocol, in addition to improving traffic to Unicast, caused to improve response time rather than SMP. The method decreases sent message size and compound it with other methods such as multicasting can present an acceptable result. To decrease response time, we used server side caching. In the future, we can use client side caching or two side caching.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 4(10), Jan, 2014, pp. 86-97, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

References

[1] D. Box, D.Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, “Simple Object Access Protocol (SOAP)1.1,” World Wide Web Consortium (W3C), May 2000.

[2] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, “Web Services Description Language (WSDL) 1.1,” World Wide Web Consortium (W3C), Mar.2001.

[3] K. Chiu, M. Govindaraju, R. Bramley,” Investigating the Limits of SOAP Performance for Scientific Computing”, Department of Computer Science Bloomington, Indiana University,IEEE,2002.

[4] K. Anh Phan, Z. Tari, P. Bertok,” Optimizing Web Services Performance by Using Similarity-based Multicast Protocol”, RMIT University,IEEE,2006.

[5] K. Devaram and D. Andresen,” SOAP Optimization via Parameterized Client-side Caching”, Department of Computing and Information Sciences 234 Nichols Hall, Kansas State University,2003.

[6] D. Andresen, D. Sexton, K. Devaram, V. P. Ranganath,” LYE: a high-perfomance caching SOAP implementation”, Department of Computing and Information Sciences Kansas State University,IEEE,2004.

[7] K. Thompson, G.J. Miller and R. Wilder, Wide-area internet traffic patterns and characteristics, IEEE Network 11 (6) (1997) 19–23.

[8] K. Anh, T. Phan,” Enhanced SOAP Performance for Low Bandwidth Environments”, A thesis submitted in fulfilment of the requirements for the degree of Masters of Applied Science, RMIT University, August 2007.

[9] J. Tekli, E. Damiani, R. Chbeir, and G. Gianini,” Similarity-based SOAP Processing Performance and Enhancement”, published in "IEEE Transactions on Services Computing 99, PrePrints (2011) 1939-1374".

[10] D.A. Menascé and V.A.F. Almeida, Capacity Planning for Web Services – Metrics, Models and Methods. 2002. p.556, Prentice Hall.

[11] A. Ghazaleh N. Govindaraju and L. M.J., Optimizing Performance of Web Services with Chunk-Overlaying and Pipelined Send. . Proceedings of the International Conference on Internet Computing (ICIC), 2004. pp. 482-485.

[12] R. Elfwing, U. Paulsson and L. Lundberg, Performance of SOAP in Web Service Environment Compared to CORBA. Proceedings of the 9th Asia-Pacific Software Engineering Conference (APSEC'02), 2002. pp. 84-94.

[13] S. Makino, M. Tatsubori, K. Tamura and Y. Nakamura, Improving WS-Security Performance with a Template-Based Approach. Proceedings of the IEEE INternational Conference on Web Services (ICWS'05), 2005. pp. 581-588.

[14] T. Suzumura, T. Takase and M. Tatsubori, Optimizing Web Services Performance by Differential Deserialization Proceedings of the IEEE International Conference on Web Services (ICWS'05), 2005. Vol. 1, pp.185- 192.