A Review On Performance Enhancement Of Web Services Using Tagged-Sub Optimal Code(TSC) Compression

Aakansha R. Shukla[1], Prof. Parag D. Thakare[2]

[1]PG Student, Department of CSE, JCOET Yavatmal, Maharashtra, India
[2]Professor, Department of CSE, JCOET Yavatmal, Maharashtra, India

---***---

Abstract - *The most straightforward way to improve performance of any system is to define the bottlenecks and think of ways to remove them. Web services are the inseparable part of any web application, as a result enhancing performance of web services will have a great effect on the overall performance of the system. The most widely used communication protocol in the web services model, SOAP, is a simple protocol for the exchange of messages. The serialization of large SOAP responses is a major performance bottleneck in a SOAP message exchange.*

Compression can be used to reduce the size of files and speeding up the transmission time over networks. However, not all compression techniques have the same features and capabilities to improve the performance of transmission over networks. This paper shows a comparison between different compression algorithms in order to improve the performance of web-services over the Internet. Nowadays, Service Oriented Architecture (SOA) being used heavily between applications as interaction between loosely coupled services, which are function independently. Therefore, fast and efficient services offered through the web services are needed. Enhancing performance of web services, would improve overall system's performance. As a result, compressing and reducing the size of SOAP messages traveling over the network, improves the webs-service performance.

This paper compares the performance of web-services by compressing SOAP messages using Tagged Sub-optimal Code (TSC) and Huffman Encoding Algorithms. Experimental results show that web-services compressed using TSC speeds up the performance of web-services compared to normal web-services and web-services compressed using Huffman encoding.

Key Words: **Web Services, SOAP, Performance, XML, Network, TSC, Huffman**

1. INTRODUCTION

Web service is a widely-used technology for exchanging data between applications and its scope of usage has widened even more in recent years.

A SOAP message is an XML-based protocol, which can be used for exchanging information between computers. It allows applications automatically to connect to remote services and invoke remote methods. The main challenge of SOAP performance is when exchanging large SOAP messages over a network, which increases the transmission time and causing performance delay [1]. Therefore, creating web services without considering the performance of SOAP messages, could reduce overall system's performance. In this work, different compression algorithms will be used to compress the SOAP messages in order to measure the performance of web services using compressed techniques.

1.1 Performance of Web Services

Performance in web services is an important issue, especially if a large amount of data has to be exchanged, so to solve this problem you need to find a way to lower the size of the data which is passed between the clients and the servers. Compressing text in Web service is most important challenge to improve its performance because message sizes in Web services are larger than in traditional web technologies. Compression means that the size of message reduced up to 80% and the data needs less time to be transferred over a network, which will affect a higher performance for client-server applications that communicate with text, like XML Web services.

1.2 Compression Techniques

"Data compression refers to reducing the amount of space needed to store data or reducing the amount of time needed to transmit data. The size of data is reduced by removing the excessive information. The goal of data compression is to represent a source in digital form with as few bits as possible while meeting the minimum requirement of reconstruction of the original."[2]. There are two general types of compression algorithms:

1.2.1 Lossless compression

Lossless compression compresses the data in such a way that when data is decompressed it is exactly the same as it was before compression i.e. there is no loss of data. Lossless compression basically rewrites the data of the original file in a more efficient way.

1.2.2 Lossy compression

Lossy compression is the one that does not promise that the data received is exactly the same as data send i.e. the data may be lost. So, Lossy file compression results in lost data

and quality from the original version. They are typically achieving much better compression ratios than the lossless algorithms.

Tagged Sub-optimal Code (TSC) [2] is a variable-length sub-optimal code, which will be used in this work compared to Huffman encoding. Figure 1 shows the Quad tree for TSC. In previous work [3], we introduced a methodology to compress a SOAP message and compare it to different compression algorithms. This works shows the results of the methodology we have proposed.

2. EXISTING WORK

In this, problem is addressed on the sender's side, by avoiding serializing entire messages. The sender side of our SOAP implementation, called bSOAP, saves copies of outgoing message buffers, and tracks when the client code makes changes to the data sent in the messages. Only the changes are reconverted and rewritten into the outgoing message buffer template. The rest of the template remains unchanged from the previous send, avoiding serialization for that portion of the message. Our performance study indicates that this technique, called differential serialization (DS).

The approach describes the design and implementation of differential serialization's analogue on the server side, called differential deserialization (DDS). The idea is to avoid fully deserializing each message in an incoming stream of similar messages. Differential deserialization gets its name because the server-side parser deserializes the differences between an incoming message and a previous one.

SEM and DDS are completely separated and independent ideas and implementations; neither depends on the other for any portion of the performance enhancements; the two techniques represents very different realizations of the same high level idea; DS for sending SOAP data, and DDS for receiving it. SEM is a combination of DS and DDS, that is, it shares the idea behind both techniques. On the other serialization process will be improved but with a completely different implementation.

In general, SEM is more promising optimization technique than DS, because it is more applicable. DS only works if the same client sends a stream of similar messages. DDS can avoid deserialization of similar messages sent by multiple different clients while SEM does both.

Implementing a new compression method called "tagged suboptimal code (TSC)" to compress XML content replace with its corresponding code word representation, then transmitted the resulted comprise file (or message).

Minaei, B., Saadat, P. in (2009). , try to improve performance of any system by enhancing performance of web services. The main idea of their studies is to avoid the redundant serialization stage of SOAP responses for request

which have the same call parameters. They provide solution to take advantage of similar Soap requests on a web server for a particular web service by running a Middleware on top of web server called Serialization Enhancement Middleware (SEM).

In (2010) Al-Shammary, D., Khalil, I., try to enhancing the performance of Web services by compressing SOAP messages. They are proposing two innovative techniques capable of reducing small as well as very large messages. Their research shows that high Compression Ratio is up to 7.8 and around 13.5 for large and very large messages respectively.

Al-Shammary, D., Ibrahim, K. (2012)., proposed two new redundancy-aware SOAP Web message aggregation models - Two-bit and One-bit XML status tree reduce the required bandwidth, latency, and improve the overall performance of Web services by enable the Web servers to aggregate SOAP responses and send them back as one compact aggregated message. Their experiments achieving compression ratios as high as 25 for aggregated SOAP messages in significant performance for both aggregation techniques.

In (2012) a new algorithm for data compression is proposed by Suarjaya, I. called j-bit encoding (JBE). It is classified to lossless compression, because it will manipulates each bit of data inside file to minimize the size without losing any data after decoding which. To measure the performance of this algorithm you need to comparing combination of different data compression algorithms. In this experiment the algorithm gives better compression ratio when inserted between move to front transform (MTF) and arithmetic coding (ARI).

3. PROPOSED WORK

The first component responsible for handling requests in a Client-Server Model is the Web Server; therefore it would be the best candidate for hosting a middleware. Our approach is to implement a middleware to run on top of any web server (IIS,Apache,..) and act as the primary component for processing request.

By definition, Web services can be communicated with over a network using industry standard protocols, including SOAP. That is, a client and a Web service communicate using SOAP messages, which encapsulate the in and out parameters as XML. Fortunately, for Web service clients, the proxy class handles the work of mapping parameters to XML elements and then sending the SOAP message over the network [8].This means that the SOAP message can be reached before and after Serialization/Deserialization process.

As the calling of web service methods has a unique signature, the probability of receiving requests with completely the same parameters for a service is so high. The idea behind this paper is to avoid the redundant serialization stage of SOAP

responses for request which have completely the same parameters.

The approach will be even more efficient if a constraint is put on the method signature. Our researches show that the best case is the situation in which the method parameters are all string and the response is a result set.
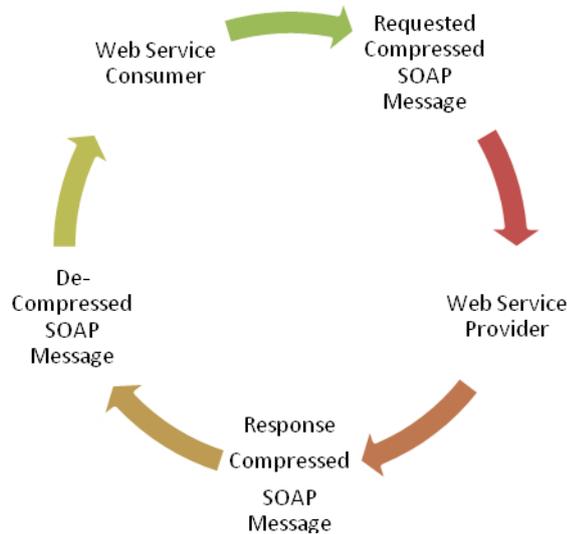

Figure:- Proposed Implementation

4. CONCLUSIONS

Web services with large SOAP messages in size encounter transmission delay over the Internet, which is one of the web services issues. As a result, using compression algorithm that has features and capabilities to improve performance in transmission over a network is a good choice. TSC is one of those algorithms, which has these capabilities and shows increasing the performance of web services when it compared to Huffman encoding and normal web services.

REFERENCES

[1]   Minaei, B., Saadat, P. (2009). SOAP Serialization Performance Enhancement Design and implementation of a middleware. International Journal of Computer Science and Information Security, IJCSIS, Vol. 6, No. 1, pp. 105-110, October 2009, USA.

[2]   Bellaachia, I. AL Rassan . Fast Searching Over Compressed Text Using a New Coding Technique: Tagged Sub-optimal Code (TSC). DCC 2004: IEEE Data Compression Conference, Snowbird, Utah, March 23 - 25, 2004.

[3]   Iehab Alrassan and Haifa Alyahya. Improve XML Web Services' Performance Using SOAP Compression. The 2014 International Conference on Semantic Web and Web Services (SWWS'14v), Nevada, 2104.

[4]   N.Abu-Ghazaleh, M. J. Lewis.-Differential Deserialization for optimized SOAP Performance. 2005 ACM/IEEE conference on Supercomputing, pp. 21-31, Seattle WA, Novomber 2005.

[5]   N.Abu-Ghazaleh, M.Govindaraju, and M. J. Lewis.-Optimizing Performance of Web Services with Chunk-Overlaying and Pipelined-Send. Proceedings of the International Conference on Internet Computing (ICIC), June 2004.

[6]   N.Abu-Ghazaleh, M.Govindaraju, and M. J. Lewis – Performance of Dynamic Resizing of Message Fields for Differential Serialization of SOAP Messages. Proceedings of the International Symposium on Web Services and applications, June 2004.

[7]   N.Abu-Ghazaleh, M.Govindaraju, and M. J. Lewis.- Differential Serialization for Optimized SOAP Performance. Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC-13), June 2004, Honolulu, Hawaii.

[8]   Miranda, Claudio:- Tools and Tips to Diagnose Performance Issues. The International Conference on JAVA Technology, Zurich.(2008)

[9]   K. Chiu and W. Lu.- A Compiler-Based Approach to Schema-Specific Parsing. In First International Workshop on High Performance XML Processing,2004.

[10]   K Devaram and D. Andresen. – SOAP Optimization via Parameterized Client-Side Caching. In Proceedings of PDCS 2003, November 3-5,2003.

[11]   I. Foster, C. Kesselman, J. Nick, and S. Tuecke. Grid Services for Distributed System Integration. Computer 35(6), 2002.

[12]   M. R. Head, M. Govindaraju, A. Slominski, P. Liu.N. Abu Ghazaleh, R. van Engelen, K. Chiu, and M. J. Lewis. –A Benchmark Suite for SOAP-based Communication in Grid Web Services. SC-05 : Supercomputing '05, Seattle WA, November 2005.

[13]   Suarjaya, I. (2012). A New Algorithm For Data Compression Optimization. Information Technology Department, Udayana University, Bali, Indonesia. (IJACSA) International Journal of Advanced Computer Science and Applications.

[14]   AlRassan, I., Assiri, A.(2009) . Optimizing web services for mobile Devices using Tagged Sub Optimal Code (TSC). MoMM '09 Proceedings of the &th International Conference on Advances in Mobile Computing And Multimedia.

[15]   Al-Shammary, D., Khalil, I. (2010). SOAP Web Services Compression Using Variable and Fixed Length Coding. Network Computing and Applications (NCA), 2010 9th

IEEE International Symposium on Date of Conference: 15-17 July 2010.

[16] Swarnkar, V., Satao, K.(2013). An Implementation of Efficient Text Data Compression. International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-3, Issue-2, December 2013.

[17] Altarawneh, M., Altarawneh, H. (2011). Data Compression Techniques on Text Files: A Comparison Study. International Journal of Computer Applications.

[18] Shanmugasundaram, S., Lourdusamy, R. (2011). A Comparative Study of Text Compression Algorithms. International Journal of Wisdom Based Computing, Vol. 1 (3).