International Journal of Research in Engineering, Science and Management
Volume-1, Issue-5, May 2018
www.ijresm.com

Using Tagged-Sub Optimal Code Compression for Improving Performance Enhancement of Web Services

Aakansha R. Shukla[1], Parag D. Thakare[2]
[1]*PG Student, Department of CSE, JCOET, Yavatmal, India*
[2]*Professor, Department of CSE, JCOET, Yavatmal, 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.

Index Terms—Huffman, Network, Performance, SOAP, TSC Web Services, XML

I. 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.

A) 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.

B) 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) 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.

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.

II. 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).

III. Proposed Work

The best candidate for hosting a middleware is the web server, the first component responsible for handling requests in a client-server model. Our approach is to implement a middleware that act as primary component for processing request and to run on the top of any web server such as IIS, Apache, etc.

By definition, Web services can be communicated with over a network using industry standard protocols, including SOAP. That is, using SOAP messages a client and a Web service can communicate with each other, 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.

In this analysis, done over the TSC with Huffman implements the compression technique that gets manipulated over the files of SOAP. Compression technique will get implemented over the message transfer through the network as well. Some SOAP are developed for the exchange of data with various users so that they get shared without compression so that it will increase the speed transmission. The proposed system will cross verify the execution with various algorithms and techniques.

Algorithm for Huffman:

Input: Text message
Step 1: Create sorted nodes on frequency / probablity
Step 2: Start loop
Step 3: Find and remove two smallest probability nodes
Step 4: Create new node, weight [node] = W (A) + W (B)
Step 5: Insert new node, back to sorted list
Step 6: Repeat the loop until the list consist of the only last node
Step 7: This algorithm terminates when there are no more

International Journal of Research in Engineering, Science and Management
Volume-1, Issue-5, May 2018
www.ijresm.com

missing values.

Algorithm for TSC:

Input: Text message

Step 1: Create a leaf node for each unique character and build a min heap of all leaf node. (Min Heap is used as a priority queue. The value of frequency field is used to compare four nodes in min heap. Initially, the least frequency character is at root.)

Step 2: Extract four nodes with the minimum frequency from the min heap.

Step 3: Create new internal node with the frequency equal to the sum of the four nodes frequencies. Make the first extracted node as its left child and the other extracted node as its right child and the rightmost extracted node will be of highest frequency among four of them. Add this node to the min heap.

Step 4: Repeat steps #2 and #3 until the heap contains only one node. The remaining node is the root node and the tree is complete.

Following Fig. 1. Showing the proposed implementation:

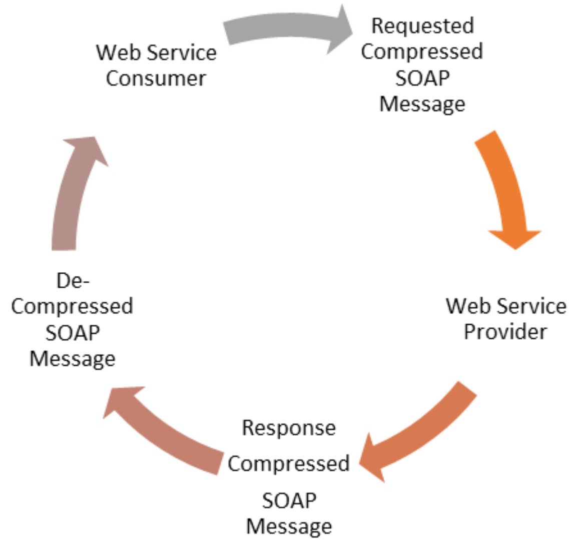

Fig. 1. Proposed Implementation

IV. Experimental Result

Following is the Table-I showing the experimental outcome of the 10 different files with different sizes which includes the result using the normal web services, Huffman coding and TSC coding.

In this experiment we found that TSC is the most efficient compressing technique with less time consumption. To summarize the outcome of the work, compression techniques are applied. It can be noticed that XML message size are reduced when they are compressed using Huffman or TSC compression techniques. However, the compression performance and ratios are different techniques.

When Huffman algorithm is applied to SOAP message, it achieved good compression ratio as compared to TSC. However, time encoding and decoding in web services using TSC is faster than using Huffman technique. As compared to Huffman algorithm and normal-web services, web-services using TSC achieved better performance as shown by the results.

TABLE I
EXPERIMENTAL RESULT USING TSC AND HUFFMAN CODING

S. No.	File Size	Normal Web Services	Webservice Using Services	Webservice Using TSC	Proposed
1	252833.0 kb	2.52833E7 Ms	2.022664E7 Ms	1.769831 E7 Ms	1.516998 E7 Ms
2	17723.0 kb	11772300.0 Ms	1417840.0 Ms	1240610.0 Ms	1063380.0 Ms
3	51770.0 kb	5177000.0 Ms	4141600.0 Ms	3623900.0 Ms	3106200.6 Ms
4	5031.0 kb	503100.0 Ms	402480.0 Ms	352170.0 Ms	301860.0 Ms
5	806.0 kb	80600.0 Ms	64480.0 Ms	56420.0 Ms	48360.0 Ms
6	5800.0 kb	580000.0 Ms	464000.0 Ms	406000.0 Ms	348000.0 Ms
7	57106.0 kb	5710600.0 Ms	4568480.0 Ms	3997420.0 Ms	3426360.0 Ms
8	1272801.0 kb`	1.272801 E8 Ms	1.0182408 E8 Ms	8.909607 E7 Ms	7.636806 E7 Ms
9	272300.0 kb	2.723E7 Ms	2.178437 Ms	1.9061E7 Ms	1.6338E7 Ms
10	98025.0 kb	9802500.0 Ms	7842000.0 Ms	6861750.0 Ms	5881500.0 Ms
Average	**158981.675 kb**	**1.58981675E7 Ms**	**1.2718534 E7 Ms**	**1.112871725E7Ms**	**9538900.5 Ms**

Also, following is the performance graph showing the result of the above table that used different compression techniques TSC, Huffman and normal web services.

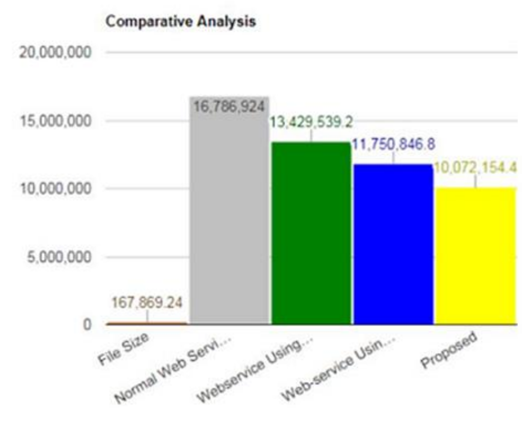

Fig. 2. Graph showing result of different web services

V. Conclusion

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

REFERENCES

[1] B. Minaei and P. Saadat, "SOAP Serialization Performance Enhancement Design and implementation of a middleware," *International Journal of Computer Science and Information Security*, vol. 6, no. 1, pp. 105-110, October 2009.

[2] A. Bellaachia and I. A. L. Rassan, "Fast searching over compressed text using a new coding technique: tagged sub-optimal code (TSC)," *Data Compression Conference, 2004. Proceedings. DCC 2004*, 2004, pp. 526.

[3] I. Alrassan and H. 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 and M. J. Lewis, "Differential Deserialization for Optimized SOAP Performance," *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, 2005, pp. 21-21.

[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*, 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. J. Lewis and M. Govindaraju, "Differential serialization for optimized SOAP performance," *Proceedings. 13th IEEE International Symposium on High performance Distributed Computing, 2004.*, 2004, pp. 55-64.

[8] Miranda and Claudio, "Tools and Tips to Diagnose Performance Issues," *The International Conference on JAVA Technology*, 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*, vol. 35, no. 6, pp. 37-46, June 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] I. Suarjaya, "A New Algorithm for Data Compression Optimization," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 8, pp. 14-17, 2012.

[14] I. AlRassan and A. Assiri, "Optimizing web services for mobile Devices using Tagged Sub Optimal Code (TSC)," *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, 2009, pp. 356-359.

[15] D. Al-Shammary and I. Khalil, "SOAP Web Services Compression Using Variable and Fixed Length Coding," *2010 Ninth IEEE International Symposium on Network Computing and Applications*, Cambridge, MA, 2010, pp. 84-91.

[16] V. Swarnkar and K. Satao, "An Implementation of Efficient Text Data Compression," *International Journal of Engineering and Advanced Technology*, vol. 3, no. 2, December 2013.

[17] S. Shanmugasundaram and R. Lourdusamy, "A Comparative Study of Text Compression Algorithms," *International Journal of Wisdom Based Computing*, vol. 1, no. 3, pp. 68-76, December 2011.