Improve XML Web Services' Performance Using SOAP Compression

Iehab Al Rassan and Haifa Alyahya
King Saud University, College of Computer and Information Sciences
Department of Computer Science

***Abstract*—Applications are part of our daily life that many people use it more than anything else. Nowadays, applications are connected via the internet and the users want to use fast and efficient services offered through the web services. Web services are the core of modern application architectures that will be used for many years. Regardless of what platform or language that developers are using, the critical skill understands how web services work. "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" [1]. Compressing and reducing the size of SOAP messages traveling over the network, improves the web services performance, which consumer invoke.**

This paper proposes and discusses a methodology that compresses the SOAP messages using compression techniques that have special features to improve the performance of SOAP messages. These compression techniques are Tagged Sub-optimal Code (TSC), Huffman Encoding Algorithm, Byte Pair Encoding (BPE) and J-bit encoding (JBE) Algorithm.

***Index Terms*—Web Services, SOAP, Performance, XML, Network, TSC, BPE, Huffman, JBE.**

I. Introduction

A **SOAP** message is an XML-based protocol for exchanging information between computers. It enables client applications to easily connect to remote services and invoke remote methods. The main challenge of **SOAP** performance is when exchanging messages over a network is the size of the **SOAP** messages which is large and effecting the transmission time. Creating Web Services without considering **SOAP** performance could reduce overall system performance. There are many ways to improve the performance of sending **SOAP** messages. In our proposed project we will apply **SOAP** message compression and evaluate the performance of Web Services using it to see which technique is better.

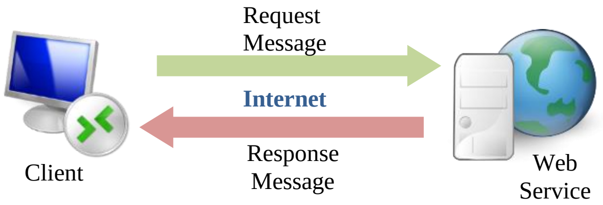

Fig. 1. Message Formats

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.

II. Related Work

In (**2008**) AlRassan, I., Assiri, A.[3], 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**). [1], 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. [4], 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**).[5], 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. [2] 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 solution

We will aim to improve the Web Services performance **SOAP** Compression using tagged sub-optimal code (**TSC**). We will discuss the benefits of using it to enhance the SOAP performance and compare it with the other techniques that used in compression such that Huffman encoding and Byte Pair Encoding (**BPE**). We will focus on **SOAP** performance and propose our approach to **SOAP** message compression of the Web Services.

In our search for the aim of project, a new algorithm of data compression, called j-bit encoding (**JBE**). "J-bit encoding (**JBE**) works by manipulate bits of data to reduce the size and optimize input for other algorithm. The main idea of this algorithm is to split the input data into two data where the first data will contain original nonzero byte and the second data will contain bit value explaining position of nonzero and zero bytes. Both data then can be compress separately with other data compression algorithm to achieve maximum compression ratio. "[2] JBE will be used to compare it with other algorithms such as Huffman encoding, Byte Pair Encoding (**BPE**) and tagged sub-optimal code (**TSC**) to improve the performance of **SOAP** message.

IV. Compression Algorithms

1) Byte Pair Encoding (BPE):

Byte pair encoding is one of the data compression techniques that replaced most common pair of consecutive bytes of data with a byte that does not occur within that data. Rebuild the original data by changing pair using table of the replacements. The **BPE** encoding algorithm is a multi-pass and requires that all the data must be stored in memory. The algorithm was first described publicly by Philip Gage in a February **1994** article "**A New Algorithm for Data Compression**" in the C Users Journal.

2) Huffman encoding:

Huffman coding is an entropy encoding algorithm used for lossless data compression. It is one of the simple compressing encoding schemes and can be implemented easily and efficiently. The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It was developed by David A. Huffman, and published in the **1952** paper "**A Method for the Construction of Minimum-Redundancy Codes**".

3) Tagged Sub-optimal code (TSC):

TSC is a suboptimal coding technique that supports minimal prefix property. It generates code-words from a tree traverse that can be represented by a quad tree (degree 4). The code-words are generated in every level and its sizes depend upon the number of represented codes (2 to 14 bits long for 254 codes). Its boundaries are instantaneously detected that why it is "tagged". The algorithm was first described publicly by Bellaachia and Al Rassan in **2004**.

4) J-bit encoding (JBE):

J-bit encoding (**JBE**) is a new lossless compression algorithm that used to minimize the size without losing any data after decoding by manipulates each bit of data inside file. This algorithm will optimize the compression ratio rather than other data compression algorithms. To measure the performance of this algorithm you need to comparing combination of different data compression algorithms. The algorithm was first described publicly by I Made Agus Dwi Suarjaya on 5 Sep 2012 article "**A New Algorithm for Data Compression Optimization**" in the (IJACSA) International Journal of Advanced Computer Science and Applications.

V. Methodology

In order to test the performance of compression algorithms, the tagged sub-optimal code (**TSC**) Algorithm, Huffman Encoding Algorithm, Byte Pair Encoding (**BPE**) and J-bit encoding (**JBE**) Algorithm will be implemented and compared with a set of text files. Performances will be evaluated and tested by comparing the speed of invoking the webservices over a network for all these compression techniques.. The main goal of the experiment is to see which algorithm is effective for **SOAP** compression to improve the performance of **XML** web services. **JBE** compression techniques will be implemented and tested also to compare it with other algorithms to find out which algorithm achieves better performance and compression ratio..

- The development environment for the implementation will be **C#** on the **.NET** framework.

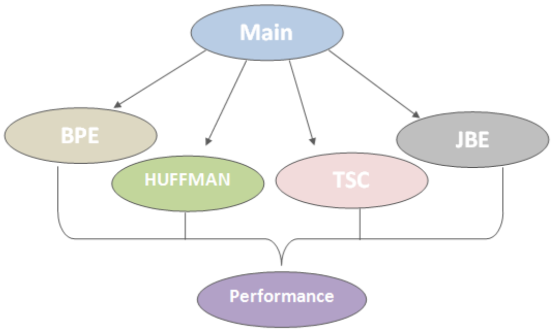

Fig. 2. Application's Class Hierarchy

A. *Proposed Implementation*

1) The web service consumer will request by send a **SOAP** message.
2) The **SOAP** message will compress before sending it by compression algorithm
3) The compressed **SOAP** message will be sent via the Internet.
4) The **SOAP** message will be de-compressed before arriving.
5) The web services provider will be receiving the request **SOAP** message.
6) The web services provider will be responding by sending the respond **SOAP** message.
7) The **SOAP** message will be compressed before sending it by compression algorithm.
8) The compressed **SOAP** message will be sent via the Internet.
9) The **SOAP** message will be de-compressed before arriving.
10) The web services consumer will receiving the response of the request.

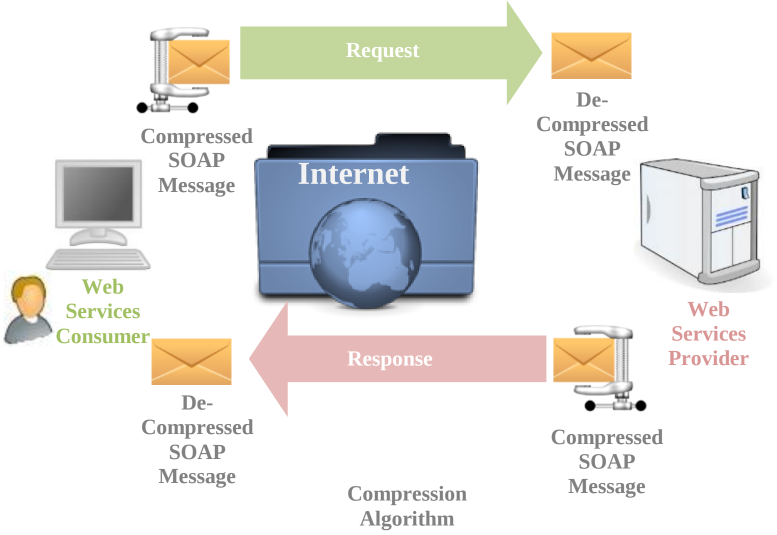

Fig. 3. Proposed Implementation

B. *Measuring Compression Performances*

The performance depends on the type and the structure of the input source. The compression behavior depends on the redundancy of symbols in the source file, so it is difficult to measure performance of a compression algorithm in general. However, the overall time needed to compress, decompress , and invoke the web services over a network will be measured and compared among all compression algorithms in order to find out which one is the best that can improve the web services performance.

The category of the compression algorithm will affect the result. The space efficiency and time efficiency would be high if we are using a lossy compression algorithm to compress a particular source file, but this would not be accepted for webservices We should use different measurements to evaluate the performances of the compression algorithms.

Compression Ratio is the ratio between the size of the compressed file and the size of the source file.

$$compression\ \ Ratio = \frac{size\ \ after\ \ compression}{size\ \ before\ \ compression} \qquad (1)$$

Compression Factor is the inverse of the compression ratio. That is the ratio between the size of the source file and the size of the compressed file.

$$compression\ \ Ratio = \frac{size\ \ before\ \ compression}{size\ \ after\ \ compression} \qquad (2)$$

Saving Percentage calculates the shrinkage of the source file as a percentage. (3)

$$saving\ \ percentage = \frac{size\ \ before\ \ compression - size\ \ after\ \ compression}{size\ \ before\ \ compression}\%$$

VI. Conclusion

Transferring large message size is one of the web services issues, so the best way to improve performance by using data compression. The compression algorithms comes as a solution to reduce the message size and improve the performance of web services by a number of researchers, using different concepts and strategies. In our project, we will implement some algorithm of compression such as the tagged sub-optimal code (**TSC**) Algorithm, Huffman Encoding Algorithm, Byte Pair Encoding (**BPE**) and J-bit encoding (**JBE**) Algorithm, so we can figure out which one is the best to use. Some factors are considered for comparison in order to identify the best solution. The main challenge is to achieve high compression ratio and at the same time speeding up the overall time needed for invocation process over the network.

In the future, we will develop a programming interface with a set of text files to measuring the compression performances of each Algorithm. After the implementation is done an experiment will be conducted to evaluate which algorithm is good for improving the **SOAP** message compression.

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] 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.

[3] AlRassan, I., Assiri, A.(2009). Optimizing Web Services for Mobile Devices using Tagged Sub Optimal Code (TSC). MoMM '09 Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia.

[4] 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.

[5] Al-Shammary, D., Ibrahim, K. (2012). Redundancy-aware SOAP messages compression and aggregation for enhanced performance, School of Computer Science & IT, RMIT University, Melbourne, Australia.

[6] 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.

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

[8] Girish Tere, G., Jadhav, B. (2011). Improving Performance of XML Web Services. Technology Systems and Management, Communications in Computer and Information Science.

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

[10] Rosu, M. (2012). Adaptive parsing and compression of SOAP messages. IEEE International Conference on Web Services (ICWS 2007).