Efficacious Transmission Technique for XML Data on Networks

Xu Huang, Alexander Ridgewell, and Dharmendra Sharma

School of Information Sciences and Engineering, University of Canberra, ACT Canberra, 2617, Australia

Summary
XML is increasingly being used to transmit data on networks but is a verbose format and needs an efficient encoding to send relatively large amounts of data efficiently. It is a common technical challenge for researchers in XML-driven networks to have good performance. One may employ a middleware to enhance performance by minimizing the impact of transmission time [1, 2]. Normally, to reduce the amount of data sent the XML documents are converted to a binary format using a compression routine such as XMill [3]. However while this would reduce the amount of data, it results in an increase in the CPU time as the XML document must be compressed before being sent and uncompressed when it is received. We first present a technique, called multi-threshold method to decide if it would be transmitting the XML document as a compressed document or not depending on a threshold that we first establish. We compare this technique to a widely known technique proposed by Ghandeharizadeh et al [1] called Network Adaptable Middleware (NAM). Experimental results show that for an established threshold size at a discussed situation, our method is superior to the NAM method [1]. The simulation results shows that for an example of a 4.5 MB XML file in our method will make the CPU time decreasing 22.69% and total transition time will save 4.61% in comparison with the method described in [1]. The final simulations show the multi-threshold does work for XML data transmissions.

Key words:
Web server, XML, transmission efficiency, document size compressing, XMill

1. Introduction

XML has become an increasingly important data standard for use in organizations as a way to transmit data [4, 5, 6, 7,11]. Additionally it is being used to enable web services and similar, often custom, RPC functionality to allow greater access to data across multiple systems within an organization and allowing the possibility of future systems to be created from collections of such RPC functionality.

XML is a verbose, text based format with strict requirements on structure and is often criticized for its large space requirements. This large size can be particularly problematic for use in transmission across a network, where network bandwidth restrictions can cause significant delays in receiving the transmission.

One solution to this problem is to look at reducing the size of these transmissions by rendering them in a binary format, such as by using XMill to compress an XML document. However such methods can take longer as compressing and decompressing may take more time than what is saved transmitting the smaller XML document.

One solution to this problem may be the Network Adaptable Middleware (NAM) raised by Ghandeharizadeh et al [8], even though there are some ways to directly compress, such as column-wise compression and row-wise compression for large message sizes [9]. This solution estimates the time it will take to compress, transmit in binary format and decompress a document compared to an estimate of how long it would take to transmit the document as uncompressed text. The estimates are based on a persistent collection of information on how the system has performed in the past and provides an accurate estimate on whether it would be faster too compress the document before transmission or not.

We have looked at another way of determining when to compress an XML document before transmitting it in our *One Pass Technique* (OPT). In this technique we determine a threshold size value for the network. Any XML document size smaller than this threshold it will be sent uncompressed while any XML document size larger it will be compressed before it is sent.

2. Establishing Threshold and One Pass Technique (OPT)

In contrast to the five network factors that contribute to the latency time of delivering a query output [1] based on the analysis of the one gigabyte TPC-H benchmark [10], our method presented here is utilizing an established " threshold" for the current working status and then to

Manuscript received March 28, 2006.

have "one-pass" transmission. We defined a threshold value for the network such that the transmitted time, for XML documents size are compressed (such as via XMill) and uncompressed, will be comparable. To determine what this value could be, we first need to determine the networks characteristics. As the networks characteristics will evolve with time the threshold value needs to dynamically change with the network.

Before OPT can be used on a network we need to determine the threshold value by making a number of XML transfers of different sizes across the network. The transmissions need to be made both with the document compressed, using XMill as an example, (and decompressed where it is received) and by transmitting the document without compression. An estimate of how long it takes to transmit a document of a given size can then be determined by curve fitting to these results. The threshold value is set to be the size when the estimated time to transmit it without compression is equal to the estimated time to transmit it with compression. In some situations this may result in a threshold value that will require compression of all documents or one that will never require compression of a document.

There are a number of factors that can prevent OPT from yielding the best result for all cases. The threshold value will only be valid for the network bandwidth it is calculated for, so if that bandwidth changes a threshold value will give an inaccurate result and a new threshold value will need to be determined.

The compression and decompression times are dependent on the CPU load. If the load on a CPU is heavier (or lighter) than it was when calculating the threshold value it may not make the appropriate decision on whether or not to use compression on the XML document. Similarly the technique works best with a homogenous set of CPUs. Different CPUs will take different time periods to compress and decompress the XML documents. The compression/decompression time of two low end CPUs on a network will be different to the compression/decompression time of two high end CPUs on the same network using the same threshold value. This can also lead to the OPT making a wrong decision on whether or not to compress the document.

OPT can also be affected by changes in the networks traffic density. If the network is under a heavier load than it was when the threshold value was calculated the technique is more likely to transmit an uncompressed XML document when a compressed document would have been faster, and with a lighter network load compressed XML transmissions are more likely to occur when an uncompressed transmission would have been faster. OPT is best used in a homogenous environment where the network bandwidth is well known and network traffic is reasonably stable.

3 Experimental Results: Some Examples

A number of XML documents were gathered to test using a time based threshold to decide on when to compress a document and when not to. These files were of different sizes. An application program was written to transmit these documents a number of times across a network using a threshold value. Any XML document with a size greater than the threshold value is transmitted compressed while all other XML documents are sent uncompressed. The algorithm used is:
If $\text{Size}_{\text{Document}} > \text{Size}_{\text{Threshold}}$ *Then* transmit_compressed, *Else* transmit_uncompressed

A similar application was set up to transfer the documents using the NAM methodology (Ghandehazrizadeh, 2003). NAM uses measured network and computer characteristics to compare estimates on how long it would take to transmit an uncompressed document against an estimate of how long it would take to transmit a compressed document. The algorithm used is:

If $\text{Time}_{\text{Uncompressed Transmission}} > \text{Time}_{\text{Document Compression}} + \text{Time}_{\text{Compressed Transmission}} + \text{Time}_{\text{Document Decompression}}$ *Then* transmit_compressed, *Else* transmit_uncompressed.

The experiment was conducted using a client PC (754pin Athlon64 3200+@2.05GHz with 1GB RAM), one Server PC (Celeron D 2.8@2.79GHz with 512MB RAM) connected by a Router (Billion BIPAC 7402G) over a 100MBit Ethernet connection.

A set of twenty-seven runs were carried out to determine the characteristics of the network before the applications were run against it, solving the quadratic equations used to get the time and size estimates NAM uses in it decision algorithm and determining the threshold value for the current network traffic load for the OPT. The threshold value was found to be 425KB.

Figure 1 shows the comparison between the time it took NAM to decide whether to send the XML document compressed or uncompressed and the time it took the STT to do the same.

Figure 2 shows the total transmission time in each run using NAM with the total transmission time in the same run using the OPT.

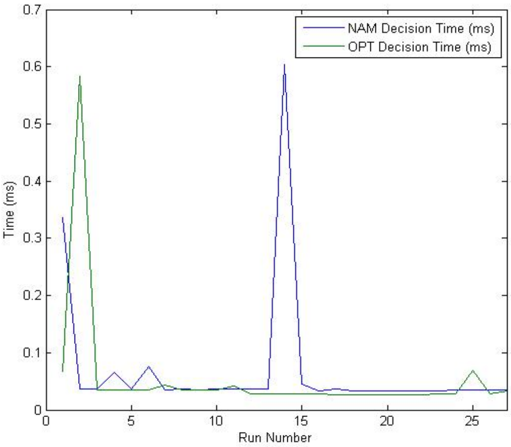

Figure 1: NAM decision time vs. OPT decision time

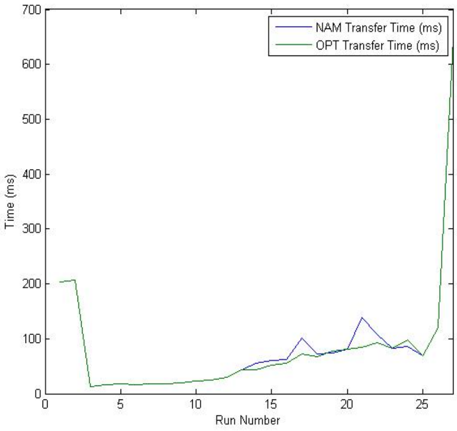

Figure 2: Total NAM transmission time vs. total OPT transmission time

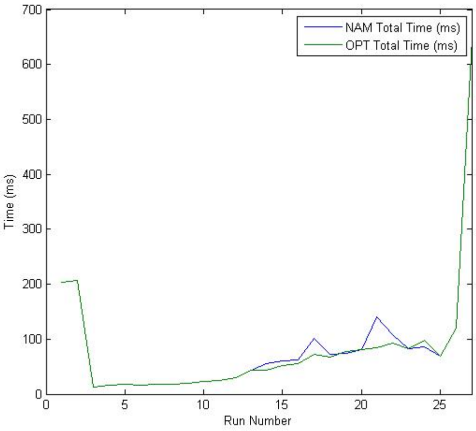

Figure 3: NAM total time vs. OPT total time

Figure 3 shows the combined total transmission and decision times for NAM with the total transmission and decision times for the OPT.

In these experiments we see the OPT performing slightly better than NAM, completing the 27 runs 149.33734 ms faster than with NAM, 148.90656 ms in the total transmission time and 0.43078 ms faster than NAM in the decision making time.

In order to confirm our results, we take the file size of ±10% and ±20% of the threshold value to see how the file size to affect the transition time. The results are shown in Figure 4. It is clearly shown that the lower size files are constantly taking more time in compassion with large size files. This shows that those files that the sizes are above the threshold should be compressed before any transmission and those files that the sizes are below the threshold should be uncompressed. The results of the example show that the worst case could make the time taken as large as > 10 times longer than it normally takes.

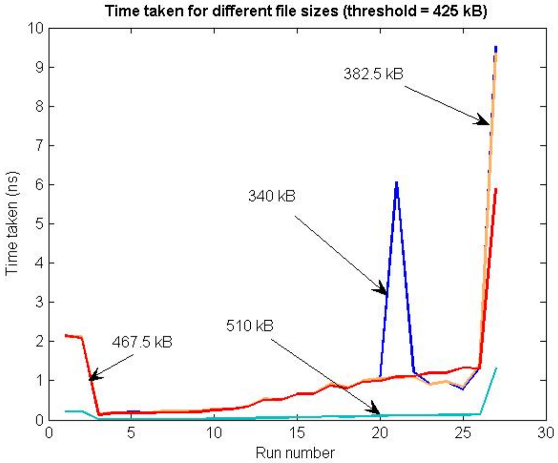

Figure 4: Times taken for the files changed to the threshold values changed ±10% and ±20% and then (using five parameters) transmitting.

In order to check how the multi-threshold affect the solutions, we took three different thresholds, namely 300 kB, 552.5 kB and 595 kB. The first one is less than 425 kB that we discussed above, the rest two are higher than 425 kB and a reasonable close. The results are put in Figure 5, showing the times are taken for different XML file sizes. Figure 6 is shown those data in 3D, namely in "run number" and "XML file size". It is clearly shown that the multi-threshold method is working, when net work's environments changed as well as the thresholds, but the method is same. Larger than threshold should be compressed wile less than the threshold should not be compressed. However, if the environment does change but the threshold does change, the network will sit on mismatched state, which leads transmissions very costly.

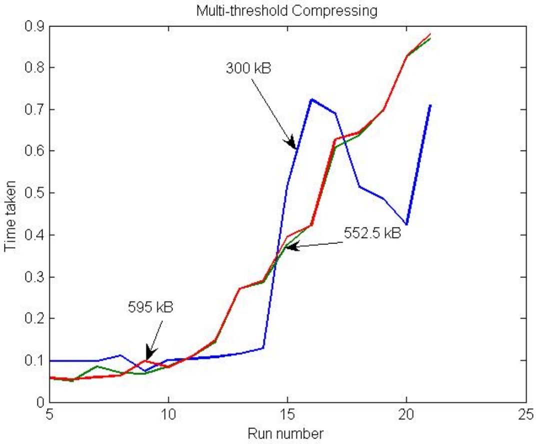

Figure 5: The time take for different runs (different sizes)

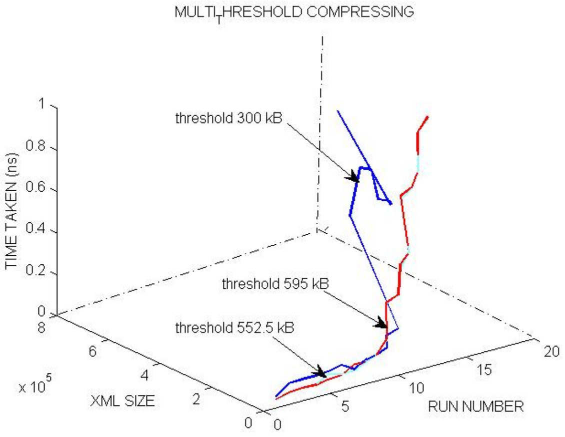

Figure 6: 3-D plotting. Shows the multi-threshold method

It is seen that the threshold does make different curves but the threshold 595 kB and 552.5 kB are very close hence, those two curves are almost the same even there are some differences there.

4 Conclusion

We have examined the possibility of using the OPT to control when an XML document should be compressed before being transmitted over a network. We compared this technique to another control technique, the Network Adaptable Middleware (NAM), and found that for a stable network of known characteristics and a random selection of XML documents the OPT is able to out perform the NAM technique. While we suspect that the NAM technique would be able to match the transmission times of the OPT after enough data has been collected to refine it's estimates the lower CPU decision making time required for the OPT means that it is a better choice for situations where a network is relatively stable in bandwidth, CPU load and network traffic density.

The experimental results demonstrate that our OPT method is superior to the method offered by [1] for fixed size, for example for 4.5 MB XML file our method will make the CPU time decreasing 22.69% and total transition time will save 4.61% in comparison with method [1].

We also show the results that the time taken for the cases that if the file's size changed ±10% and ±20% of the threshold value. The final results strongly support our method. If we use five-parameter method (or NAM) to transmit those files, the results show that the worst case could make the time taken as large as > 10 times what it normally takes.

We also check how the different thresholds affect this method, multi-threshold method.

Acknowledgments

The author wishes to show his acknowledgments to the supports from the University Research Grants for this project.

References

[1] S. Ghandeharizadeh, C. Papadopoulos, M. Cai, and K. K. Chintalapudi, Performance of Networked XML-Driven Cooperative Applications", In Proceedings of the Second International Workshop on Cooperative Internet Computing Hong Kong, China, August 2002.

[2] Alexander Ridgewell, Xu Huang, and Dharmendra Sharma, "Evaluating the Size of the SOAP for Integration in B2B", the Ninth International Conference on Knowledge-Based Intelligent Information & Engineering Systems Melbourne, Australia, September, 2005. Part IV, pp.29.

[3] H. Liefke and D. Suciu. XMill: An efficient Compressor for XMLL Data. Technical Report MSCIS-99-26, University of Pennsylvania, 1999.

[4] Curbera, F. Duftler, M. Khalaf, R. Nagy, W. Mukhi, N and Weerawarana, S.: Unraveling the web services web: An introduction to SOAP, WSDL, UDDI. IEEE Internet Computing, 6(2): 86-93, March-April 2002.

[5] Fan, M. Stallaert, J. and Whinston, A. B.: The internet and the future of financial markets, Communications of the ACM, 43(11):83-88, November 2000.

[6] Rabhi, F.A. and Benatallah, B.: An integrated service architecture for managing capital market systems. IEEE Network, 16(1):15-19, 2002.

[7] Kohloff, Christopher and Steele, Robert: Evaluating SOAP for High Performance Business Applications: Real-Time Trading Systems, 2003, http://www2003.org/cdrom/papers/alternate/P872/p872\kohlhoff.html, accessed 22 March 2005.

[8] S. Ghandeharizadeh, C. Papadopoulos, M. Cai, R. Zhou, P. Pol, NAM: A Network Adaptive Middleware to Enhance Response Time of Web Services, 2003, MASCOTS 2003: 136.

[9] R.R. Iyer and D. Wilhite. " Data Compression Support in Databases." In Proceedings of the 20th International Conference on Very Large Dasta Bases, 1994

[10] M. Poess and C. Floyd. "New TPC Benchmarks for Decision Support and Web Commerece." ACM SIGMOD Record, 29(4), Dec 2000.

[11] Valter Crescenzi, "Automatic Information Extraxtion from Large Websites", Journal of the ACM, Vol. 51 No.5 Sep. 2004 00731-779.

Dr Xu Huang Received the B.E. and M.E. degrees and Ph.D. in Electrical Engineering and Optical Engineering from the Hauzhong University of Sciences and Technology, P.R. China prior to 1989 and Ph.D. in Experimental Physics in the University of New South Wales, Australia in 1992. He has been working on the areas of the telecommunications, optical communications, and wireless communications more than 25 years. Currently he is the Head of the Networking Engineering at the School of Information Sciences and Engineering, University of Canberra, Australia. He has been a senior member of IEEE in Electronics and in Computer Society since 1989 and a Member of Institution of Engineering Australian (IEAust), Chartered Professional Engineering (CPEng), a Member of Australian Institute of Physics. He is a member of the Executive Committee of the Australian and New Zealand Association for Engineering Education, a member of Committee of the Institution of Engineering Australia at Canberra Branch.

Alexander Ridgewell received the B.Sci. degree from the Australian National University in 1998. He has been working as a graduate student towards a M.E. at the University of Canberra since 2004. His research interests include means to increase the efficiency of XML transports.

A/Prof Sharma is the Head of the School of Information Sciences and Engineering at the University of Canberra. He is an established researcher in dynamic planning systems, fuzzy reasoning, distributed artificial intelligence and intelligent multiagent systems. He is currently leading teaching and research in software engineering, artificial intelligence and multiagent systems at the University of Canberra. He has published widely in these areas and currently has eleven postgraduate research students. He has won several research grants for his work. A/Prof's current research interest on collective intelligence from distributed agents is very relevant to the proposed project. Smart digital image processing and image transmission require an articial intelligence approach within a multiagent architecture for the needed functionality. The proposed work is expected to be carried out within the framework of Multi Agent Reasoning System Environment (MARSE) developed by AProf Sharma's team. MARSE is a framework models a multiagent architecture, interaction language, inter-agent communication protocol etc and provides a rich framework to capture the digital image analysis and communication problem. The framework has been successfully applied to medical decision support, dynamic planning, and thin client computing. The results are encouraging and beneficial for the proposed work