Using SOAP with Attachments for e-Science

Ying Ying, Yan Huang, and David W. Walker
School of Computer Science, Cardiff University
PO Box 916, Cardiff CF24 3XF
{Y.Ying, Yan.Huang, David.W.Walker}@cs.cardiff.ac.uk

Abstract

This paper presents an experimental evaluation of the performance of standard SOAP compared with SOAP with Attachments (SwA) in communicating floating-point matrices of different sizes. Several key factors, including response time, size of SOAP message, and XML parsing time are addressed. In addition, these performance-related factors are also compared for two different formats of attachment – MIME (Multipurpose Internet Mail Extension) and DIME (Direct Internet Message Encapsulation). The objective of this paper is to identify how the use of SwA to deal with complex datatypes and large amounts of data can improve performance by reducing XML parsing. The results show that sending and receiving a SOAP message that carries same data takes much longer using standard SOAP than using SwA. Furthermore, the size of the SOAP message built by standard SOAP is several times larger than when using SwA.

1. Introduction

e-Science is based on the integration of distributed resources for solving challenging problems across a broad spectrum of scientific disciplines, and often involves the communication of large complex data objects over the network. The speed with which such data objects can be communicated is an important factor in determining how effective a Grid-based approach is for solving a particular problem. Grid computing is rapidly transitioned to a service-oriented architecture (SOA) based on a Web services framework, which offers the power of resource and service sharing, and a high degree of usability and interoperability. Thus, the underlying communication with and between services plays a critical role in e-Science.

SOAP (Simple Object Access Protocol) is commonly used for interprocess communication in distributed applications, in particular between Web services. Standard SOAP represents the invocation parameters of a Web service in XML, hence it is widely perceived that service invocation can suffer severe performance penalties by adopting the XML-based SOAP protocol for transporting large volumes of data[1]. Several studies have been done on the performance of using SOAP in Web services for scientific computing. These include the investigation of the performance improvement gained through message chunking, compression, routing and streaming[1]; a comparison of the latency of SOAP implementations[2]; an evaluation of the limitations for SOAP in scientific computing[3]; and an evaluation of the multi-protocol XSOAP in improving SOAP performance[4]. However, these studies did not address the extent to which performance could be simply and effectively improved by reducing XML parsing. This can be achieved by reducing the payload of a SOAP message through the means of transmitting most of the associated data as a SOAP attachment.

2. SOAP with Attachments (SwA)

SOAP with Attachments (SwA) is an abstract model for SOAP attachments defined by W3C in SOAP 1.2 Attachment Features. It provides the basis for the creation of SOAP bindings that transmit attachments along with a SOAP envelope, and provides a reference to an attachment from the envelope. SOAP attachments are described using the notion of a compound document structure consisting of a primary SOAP message part and zero or more secondary parts known as attachments[5]. The primary SOAP message provides the processing context for the compound SOAP structure as a whole including the secondary parts. A secondary part is a resource that has identity and is identified by a URI. The representation of the resource can be of any type and size. There are two main implementations of SwA currently, which are based on: MIME (Multipurpose Internet Mail Extension) and DIME (Direct Internet Message Encapsulation).

2.1 SwA using MIME

MIME was originally developed for emailing with attachments. Typically, e-mail messages

with attachments are sent over the Internet using Simple Mail Transfer Protocol (SMTP) and MIME. SMTP is limited to 7-bit ASCII text with a maximum line length of a thousand characters which results in the inability to send attachments. MIME addresses these limitations by specifying message header fields and allowing different related objects such as attachments to be included in the message body in the form of a MIME multipart. RFC 2387 specifies the Multipart/Related Content-Type to provide a mechanism for representing an object that consists of related MIME body parts. In SOAP Messages with Attachments[6], this MIME multipart mechanism is used for encapsulation of compound documents to bundle attachments related to the SOAP message.

A typical SOAP message with attachments using MIME is structured as follows:

Content-Type: multipart/related;
type="text/xml";
start="<main>"
boundary="-----=Part_MIME"
….
-----=Part_MIME
Content-Type:text/xml; charset=UTF-8
Content-Transfer-Encoding: binary
Content-id=<main>

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelop ….>
…
  <in0 href="cid:attachment"/>
…
</soapenv:Envelope>
-----=Part_MIME
Content-Type:application/octet-stream
Content-Transfer-Encoding: binary
Content-id=<attachment>
…

In each multipart header, Content-Type, and Content-Transfer-Encoding declare the type and encoding style of this part, and Content-id identifies and references this part.

2.2 SwA using DIME

DIME is a new specification from Microsoft that provides a method for sending and receiving SOAP messages along with additional attachments, like binary files, XML fragments, and even other SOAP messages[7].

A DIME message consists of a series of one or more DIME records. Each record is self-describing – that is, the record header contains binary information used by a parser to interpret the message. Figure 1 shows the layout of fields in a DIME record.

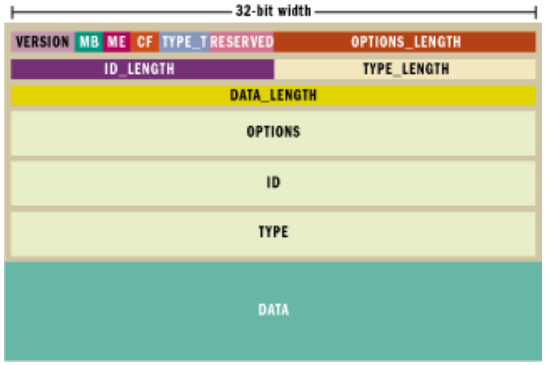

Figure 1: DIME Record [8]

The most significant fields in a DIME record are MB and ME, which specify whether this record is the first or the last record of the message. This feature is an important difference between MIME and DIME message processing. When parsing a MIME message, all of the data in the message must be read and interpreted to determine simple things like the number of attachments included in the message. However, when using DIME, a parser can simply use the data in the record headers to quickly walk through and count the number of records in the message without having to read any record data[8]. This feature enables DIME to process messages faster and more efficiently.

A typical SOAP message with attachments using DIME is structured as follows:

Content-Type: application/dime
…
00001 1 0 0 0010 0000000000000000000
0000000000000000 0000000000101000
00000000000000000000000110110101
…
<soapenv:Envelop…>
…
  <in0 href="uuid:attachment"/>
…
</soapenv:Envelop>
00001 0 0 1 0001 0000000000000000000
0000000000101001 0000000000001010
00000000000101011010101011100000
uuid:attachment
application/octet-stream
…

In DIME records, UUID (Universal Unique Identifier) or URI (Uniform Resource Identifier) are used as identifiers in referencing DIME records from the primary SOAP message.

3. Experimental Evaluation

3.1 Experimental Design

The tests described below are designed to evaluate the performance of standard SOAP compared with SOAP with Attachments in communicating floating-point matrices of different sizes. Both the roundtrip time of communicating a SOAP message and size of the SOAP message are examined. In addition, the performance indicators are compared for two different formats of attachment – MIME and DIME.

For each SOAP message format tested, we implement a remotely accessible server with the following interface:

- float[][] getMatrix (float[][] f)

or

- javax.activation.DataHandler getMatrix(javax.activation.DataHandler dh)

Tests are performed by sending floating-point matrices of size 10x10, 50x50, 100x100, 500x500 and 1000x1000. For standard SOAP messages, the matrix is sending directly as a float[][] object. For MIME and DIME, the float[][] object representing the matrix is serialized to a byte[] object, and this byte[] object is used to build a DataHandler object as attachment to send with primary SOAP message.

Each client implementation makes a call to one of the above method. The client prints the time required for these calls excluding the setup time to create the float[][] matrix object, but the serialisation and deserialisation time of converting between the float[][] object and the DataHandler object is counted. At the same time, the client also prints the size of the SOAP message.

The SOAP implementation used in our tests is Apache Axis 1.1. Axis is chosen because it is the only one that could carry both an array of arrays and an attachment in comparing with Apache SOAP, XSOAP and MS SOAP.

Both server and client are written in Java and complied and run with Sun's JDK 1.4.2 for Microsoft Windows. Tomcat 5.0 provides the web server; xerces and xalan are included with Tomcat 5.0 for XML parsing.

In the results presented here both client and server are on the same host. The host machine is a laptop with a Pentium-III 900 MHz processor and 128 MB RAM, running Windows XP Professional. We have done more extensive testing with client and server being on different hosts on the same LAN, and on different continents. However, these results, which cannot be presented here due to lack of space, will be presented in a subsequent paper.

3.2 Experimental Results

Table 1 shows the roundtrip time of a SOAP message for attached matrices of different size when getMatrix() method is called.

Matrix Size	Standard SOAP (ms)	SwA using MIME (ms)	SwA using DIME (ms)
10 x 10	399.1	635.9	296.1
50 x 50	6782.3	1398	908.5
100 x 100	25802.6	3905.5	3386.4
500 x 500		85542	84225
1000 x 1000		356418	347415

Table 1: SOAP Message Roundtrip Time

Table 2 shows the size of a SOAP message for attached matrices of different size when getMatrix() method is called.

Matrix Size	Standard SOAP (byte)	SwA using MIME (byte)	SwA using DIME (byte)
10 x 10	4250	1813	1497
50 x 50	75212	11810	11498
100 x 100	292867	42314	41998
500 x 500		1006312	1006000
1000 x 1000		4011316	4011000

Table 2: SOAP Message Size

When carrying a matrix containing 10 x 10 floating-point numbers of total size 0.39 Kbytes, we cannot see too much difference between standard SOAP and SwA for both roundtrip time and message size. But when the matrix is increased in size to 50 x 50 (9.77 Kbytes) standard SOAP takes 4.85 times longer than SwA using MIME and 7.47 times longer than SwA using DIME. Meanwhile, the size of the message grows to be about 6.46 times larger for standard SOAP compared with SwA. The result is nearly the same when the size of matrix is increased to 100 x 100. But when the matrix is further increased to 500 x 500, standard SOAP could not complete the task because of an out-of-memory error. It is obvious to see there are no problems for both SwA formats even when the matrix is increased to 1000 x 1000, for which the total size is already 3.81 Mbytes. At

the same time, we notice that the difference in message size between SwA using MIME and SwA using DIME is always around 310 bytes no matter how large the attached matrix. For the roundtrip time, SwA using DIME is just slightly faster than SwA using MIME. It should be noted that the above data are illustrative, but might be different with a faster processor and larger memory.

It is not surprising to see that both SwA SOAP message formats have much better performance than standard SOAP messaging when the size of the attached matrix becomes large. Because our tests are run on the same host, the network delays are not significant. Under this condition, XML parsing and formatting are the most important factors in the performance. The large amount of XML parsing and formatting in standard SOAP messaging gives it a higher memory requirement. From our analysis of the above results, it is clear that reducing the amount of XML parsing and formatting can improve SOAP performance by a large amount. Both SwA SOAP message formats avoid large amounts of XML parsing and formatting for the float[][] object, which means that SwA has a significant impact on performance and message size. In addition, the difference between MIME and DIME should make DIME faster and more efficient in message processing (see Section 2), and this is supported by our data.

4. Conclusion

Our tests indicate that SOAP performance of complex datatypes and large amounts of data could be improved by simply using SwA. XML parsing is both a time-consuming and memory-consuming task that can be reduced to a minimal level. Furthermore, SwA can be used as an alternative method for transferring large files.

As a new approach to sending SOAP messages, SwA using MIME is already supported by Web Service Definition Language (WSDL), and an extension to WSDL to support SwA using DIME has been proposed by Microsoft. This would allow the original type of data object to be properly defined in processing both message formats. Support in WSDL ensures the precision of converting data objects during transactions. But using SwA is still restricted in that both client and server must have the same serialisation and deserialisation mechanism when converting between data objects and binary attachments.

5. Future Work

The performance advantages of using SwA when dealing with complex datatypes and large amounts of data are clear. However, current implementations of Grid middleware, such as the Globus Toolkit, do not support SwA yet, so future work will be carried out on its use in a Grid environment. At the same time, the use of SwA as an alternative to FTP and GridFTP in performing large file transfers will be investigated.

References

[1] Robert A. van Engelen, Pushing the SOAP Envelope with Web Services for Scientific Computing, In proceedings of the International Conference on Web Services (ICWS), 2003, pages 346-354.

[2] Dan Davis and Manish Parashar, Latency Performance of SOAP implementation, In 2nd IEEE International Symposium on Cluster Computing and the Grid, 2002

[3] Kenneth Chiu, Madhusudhan Govindaraju, and Randall Bramley Investigating the Limits of SOAP Performance for Scientific Computing, In proceeding of the 11th IEEE International Symposium on High-Performance Distributed Computing, 2002

[4] Madhusudhan Govindaraju, Aleksander Slominski, Venkatesh Choppella, Randall Bramley, Dennis Gannon, Requirement Requirements for and Evaluation of RMI Protocols for Scientific Computing, [www] http://www.extreme.indiana.edu/xgws/ papers /sc00_paper/

[5] W3C, SOAP 1.2 Attachment Features, [www] http://www.w3.org/TR/soap12-af/

[6] W3C, SOAP Messages with Attachments, [www] http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211

[7] Microsoft, Direct Internet Message Encapsulation (DIME) Draft, [www] http://msdn.microsoft.com/library/en-us/dnglobspec/html/draft-nielsen-dime-02.txt

[8] Jeannine Hall Gailey, Sending Files, Attachments, and SOAP Messages Via Direct Internet Message Encapsulation, MSDN Magazine, [www] http://msdn.microsoft.com/msdnmag/issues/02/12/DIME/default.aspx