International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:2, 2014

Improvement of Data Transfer over Simple Object Access Protocol (SOAP)

Khaled Ahmed Kadouh, Kamal Ali Albashiri

Abstract**—**This paper presents a designed algorithm involves improvement of transferring data over Simple Object Access Protocol (SOAP). The aim of this work is to establish whether using SOAP in exchanging XML messages has any added advantages or not. The results showed that XML messages without SOAP take longer time and consume more memory, especially with binary data.

Keywords**—**JAX-WS, SMTP, SOAP, Web service, XML.

I. Introduction

SOAP is defined by World Wide Web Consortium (W3C) as a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralized, and distributed environment using eXtensible Markup Language (XML) on Web services. Its design makes it suitable for a wide variety of application messaging and integration patterns. In other words, SOAP is the envelope syntax for sending and receiving XML messages between client and services that have no prior knowledge of each other.

Web service is a technology that allows peoples, systems, and applications to communicate with each other. Basically, it works as an interface that describes a collection of operations in standard format. The operations are software application that can be discovered by Universal Description Discovery and Integration (UDDI) that allows application to discover information about Web services [2]. In order to describe the message syntax for Web services, the Web Services Description Language (WSDL) is used [8], [9]. The service is accessed by many protocols, such as Simple Mail Transfer Protocol (SMTP) and Hypertext Transport Protocol (HTTP). These protocols are transporting protocols used to transferring data such as XML document. SOAP is designed to be a new protocol for decentralized and distributed environments, used to enhance the power of internet and XML to pass typed information between the client and services [1].

SOAP is also a lightweight protocol with independent platform, independent transport, and independent operating system. This flexibility is due to its construct with time testing systems, like the HTTP protocol and text markup in XML [2].

A. Problem Definition

- XML processing can be slow and memory intensive, especially for binary data.
- When the size of file is big the response time is larger.
- XML specification is only for XML over HTTP and XML-RPC is only ASCII compliant.

K. A. Kadouh and K. A. Albashiri are with the Computer Science Department, Faculty of Science, Al-Jabal Al-Gharbi University, Gharian, Libya (e-mail: kaledgadou@yahoo.com, elbashiri0@yahoo.com).

B. Motivation and Objective

The motivation of this paper is to prove that using SOAP in Web service would ease the difficulties found in exchange XML documents, such as processing XML messages. In this paper, objectives can be summarized as follows:

- Review the XML and its success the Web service technology.
- Design and implement an algorithm for transferring SOAP document between Web services, and clients.
- Evaluate the algorithm with different examples and compare it with another approach.

The paper's organization is as follows. A brief review of some related work on Web services and SOAP is presented in Section II. The algorithm design and implementation with an overview of the mapping principle, is presented in Section III and the proposed approach evaluation is illustrated in Section IV using two scenarios, and finally some conclusions are presented in Section V.

II. Web Services and Related Technologies

Web service is an application service that is a combination of programming, data, and human resources. These are made available from a business Web server for Web users or other Web-connected programs. Web services are based on the specifications of data transfer, method invocation and publishing. Web service accepts requests, from other systems across the internet or intranet, mediated by lightweight, vendor-neutral communications technologies [1].

There are many definitions to Web service. W3C organization defines Web service as "a software system identified by a URI whose public interfaces and bindings are defined and described using XML". Its definition can be discovered by other software applications using XML based messages via internet based protocols.

These applications may then interact with the Web service in a manner prescribed by its definition, using XML-based messages conveyed by internet protocols [2]. In short, a business application sends a request to a service at a given URL using the transferring protocols over the internet. The service receives the request, processes it, and returns a response.

Web server is used in the internet and can be reached using HTTP or SMTP. HTTP is widely used protocol for data connections between Web browsers and servers. This protocol is the current standard for transferring HTML documents.

Although it is designed to be extensible to almost any document format like XML, it can also operate over TCP. After a successful connection, the client transmits a request message to the server, which sends a reply message back as XML document [3].

XML is the most important reason behind the success and popularity of Web services. In fact it is its backbone [6], [7].

III. Design and Implementation of an Algorithm for Mapping XML to SOAP

This section describes in detail the technology, and the tools used to design and implement the proposed algorithm. We present an algorithm to build the service, client, and transferring SOAP document, after mapping as a request, between the client and Web service.

The algorithm implementation can be described in three steps:

- Building Web service and adding application to it.
- Creating client application.
- Entering XML data and at the same time mapping it to a SOAP document, then transferring the document as a SOAP request and returning it as SOAP response.

A. Algorithm Design and Implementation

- The client application uses the JAX-WS programming model to invoke a Web service operation [7].
- The message sender component of the Web services client, runtime processes the request by sending a SOAP request message to the request queue identified by the endpoint location URI associated with the invocation.
- The message sender component uses the standard HTTP layer provided by the underlying messaging provider to send the request message.
- The SOAP request message is delivered to the request queue by the messaging provider.
- The message receiving component of the Web services server runtime uses the HTTP to receive the SOAP request message.
- The Web services server runtime uses the EJB container to dispatch the request to the Web service implementation class.
- After the Web service operation ends, the Web services server runtime uses the HTTP to send the SOAP response message to the reply queue.
- The Web services client runtime uses the HTTP to receive the reply message.
- The client then processes the reply to finish the invocation, and returns control back to the client application, as shown in Fig. 1.

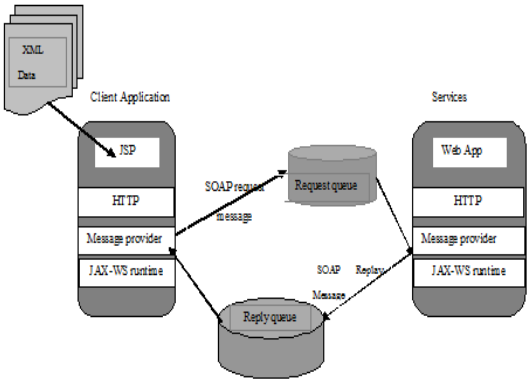

Fig. 1 Algorithm Components

B. Mapping XML to SOAP

This section discusses how an XML document becomes SOAP document by focusing on the document exchange model. As shown in Fig. 2, the work is achieved as follows: (i) we put the XML document inside a SOAP body. (ii) we wrap the SOAP body within the SOAP envelope, namespace declarations and encoding style directives for the serialization of data. (iii) we bind the document to a protocol such as HTTP. (iv) the sender class reads XML document, parses it into a DOM tree, wraps it in a SOAP envelope, and sends it to a URL destination. The destination is a simple HTTP servlet that takes the contents of the message and dumps it to the service.

Various mechanisms such as Java Message Service JMS or JAXM are used for sending a message [7]. We have used JMS in this work. The JMS is a portable, standard messaging API that enables Java programs to communicate by exchanging messages. JMS is also a component of the Java EE specification [4], [5].

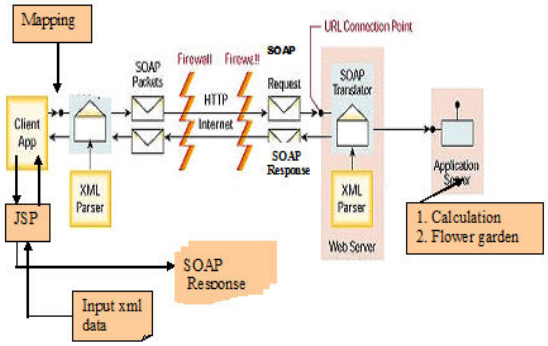

Fig. 2 Components Work of the Algorithm

C. Required Software

The proposed algorithm is implemented using the software tools shown in Table I. Many examples will be used as input of the algorithm. The input to the algorithm is an XML data. The algorithm shows how to build and deploy Web service and the client receives the data as XML document as forms found in a client application transferring it as SOAP message.

TABLE I
IMPLEMENTATION REQUIRED SOFTWARE

Software	Version required
NetBeans IDE	Java tools bundle (6.5.1)
Java2EE or application server NetBeans IDE	GlassFish application server V2.1 or Tomcat server
Java development Kit (JDK)	Version 5 or version 6

IV. EVALUATION OF THE ALGORITHM

To evaluate the algorithm, we used many examples as input data for operations located in JSP, which are created in advance.

Another evaluation of the algorithm is to compare between deploying the application to GlassFish server and deploying it to Tomcat server, and to make another comparison between the results of the proposed algorithm and those of other approaches.

A. Evaluation by Using Many Examples

As mentioned in pervious section, the algorithm is willing to accept XML data, dynamically maps it to SOAP document, and transfers it as a SOAP request.

The first part of evaluation will shows how we can compare the results of the proposed algorithm when transferring the XML document with and without SOAP WS from client to two of our applications (calculator WS, and flower image). The calculator application is used to respond to calculation requests such as summation, subtraction, multiplication or division. The flower image application is used to process image requests.

When one of the operations (i.e. *sum operation*) is chosen, the two numbers are calculated by the application that returns the result through method invocation.

One example is the sum of two numbers. Fig. 3 shows the constructed SOAP request.

```
<S:Body>
    <ns2:add xmlns:ns2="http://calculator.me.kh/">
      <i>7644</i>
      <j>8615</j>
    </ns2:add>
  </S:Body>
</S:Envelope>
```

Fig. 3 Soap Request Example

The size of file and response time of results of each sending file is measured using SOAP-UI. This tool is used to test messages over SOAP.

To evaluate the algorithm, a number of experiments were conducted to analyze the effect of: (i) the size of the data (in terms of file size), and (ii) the response time.

Table II shows measured results of applying a number of messages using a regular protocol and a SOAP WS.

TABLE II
FILE SIZE AND RESPONSE TIME

Application	File Size (byte)	Operation	Response time Without SOAP	Response time With SOAP
Calculator WS	314	Add	10707 ms	10503 ms
	331	Subtract	10770 ms	10650 ms
	370	Division	19870 ms	19750 ms
	400	Multiplication	19088 ms	19070 ms
Flower WS	1030	Get Image	sec132	121sec
	1334	Get Image	221 sec	sec212
	1668	Get Image	sec256	249 sec
	1993	Get Image	sec285	sec276

B. Comparing the Proposed Algorithm with another Approach

The second part of evaluation shows the results of the proposed algorithm compared with other approach.

There are many techniques used to send SOAP document from client to services as mentioned in Section I. However, most of these approaches are not publically available.

Fortunately, there was one approach available and similar to ours. It is called Building a Generic SOAP Framework over Binary XML [10]. This approach measures the message response time for small sized messages in which the model size of the data set range is from 0 to 1000 byte, constructs the data over BXSA and transmit over TCP.

The comparison is conducted by applying a number of operations to several files, as number data and binary data construct to SOAP and binding to HTTP, and measuring the response time of the two approaches, as shown in Fig. 4.

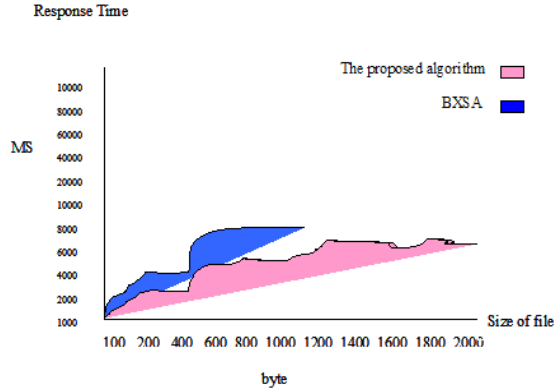

Fig. 4 Comparing the Proposed Algorithm with other Approach

The results shown in Fig. 4 indicate that our proposed algorithm offers positive advantages in that it was more computationally efficient than the other algorithm. The results of the analysis also indicated that our approach coped best with a large number of data set.

V. Conclusion

SOAP based on web services is designed with a common XML-based protocol. The goal is to allow for a machine readable document to be passed over any multiple connection protocols to create a decentralized, distributed system.

SOAP is fundamentally a stateless, one-way message exchanging paradigm between SOAP nodes, from a SOAP sender to a SOAP receiver by combining one-way exchanges with features provided by the underlying transport protocol and/or application specific information. SOAP can be used to create more complex interactions such as SOAP request and SOAP response.

Web service applications should be able to use a SOAP to make delivery mechanism improve its reliability, scalability, and performance. Since a document message is usually a self-contained XML file, it is better suited for asynchronous processing and can be placed directly into the queue. SOAP message is XML documents consisting of envelope, header, body, and fault. The envelope contains a body, a mandatory, and an optional header. The header contains information relevant to how the message is processed, and how to be sent to different paths. This includes meta-data entries that help describe or augment the main content of the SOAP message. The body is the mandatory element within a SOAP envelope. It contains the actual message, and the message that is included in an XML format.

This work introduced a design and implementation of an algorithm for transferring the SOAP document from client to services. We used two applications, calculate WS, which contains many operations, such as summation, multiplication, division and subtraction, as well as another application, named flower images WS. The evaluation of the algorithm was through comparison its work with several operations and results to the work of another approach.

The first part of evaluation showed how we were able to compare the results of the proposed algorithm with transfer the XML document from client using two applications (calculator WS, and flower image) with/without SOAP.

The size of file and response time of results of each sending file is evaluated using SOAP-UI. This tool is used to test messages over SOAP.

The experiments results showed that the algorithm, designed in this paper, can be tested by using many examples as input data for operations located in JSP. The results also showed that deploying the application to GlassFish server is faster than deploying it to Tomcat server. In addition, the results showed that the proposed algorithm scaled well when compared to other approach in terms of data size and response time.

References

[1] Nan-Chao Huang, “Cross Platform Web Service Implementation Using SOAP", 2003, Knowledge Systems Institute.
[2] Stephanie Bodoff, Debbie Carson, Maydene Fisher, "The Java™ Web Services Tutorial", 2004, California 94303 U.S.A.
[3] Brian Suda, "SOAP Web Services", 2002, School of Informatics University of Edinburgh.
[4] Carlos M. Gutierrez, Secretary," Guide to Secure Web Services", 2007, Information Technology Laboratory National Institute of Standards and Technology.
[5] Phil Adams and Zina Mostafia, "Develop a SOAP Web services application with WebSphere Application Server V7 and Rational Application Developer V7.5", 2009, IBM Corporation.
[6] Christian Platzer, "Trust-based Security in Web Services", 2004, Technical University of Vienna.
[7] Santa Clara, "The Java EE 5 for Sun Java System Application Server 9.1", 2008, SunMicrosystems, Inc. 4150Network Circle.
[8] Irmen de Jong, "Web Services SOAP and CORBA", 2004, Knowledge Systems Institute.
[9] Jack Koftikian," Simple Object Access Protocol (SOAP)", 2001, Technical University Hamburg-Hamburg.
[10] Wei Lu, Kenneth Chiu, Dennis Gannon," Building a Generic SOAP Framework over Binary XML", 2005, Department of Computer Science, State University of New York (SUNY) at Binghamton.