

Chapter 9

High Performance Approach for Server Side SOAP Processing

Lei Li

University of Science & Technology of China, China & The Chinese Academy of Sciences, China

Chunlei Niu

The Chinese Academy of Sciences, China

Ningjiang Chen

Guangxi University, China

Jun Wei

The Chinese Academy of Sciences, China

Tao Huang

University of Science & Technology of China, China & The Chinese Academy of Sciences, China

ABSTRACT

Web services, with an emphasis on open standards and flexibility, can provide benefits over existing capital markets' integration practices. However, Web services must first meet certain technical requirements, including performance, security, and so on. Based on extensible markup language (XML), Web services inherit not only the advantages of XML, but its relatively poor performance, which makes it a poor choice for many high-performance applications. In this article, a new approach is proposed to improve Web services performance. Focusing on avoiding traditional XML parsing and Java reflection at runtime, this article presents a service-specific simple object access protocol (SOAP) processor to accelerate the execution of Web services. Moreover, the SOAP processor embeds several cache implementations and uses a novel adaptive cache mechanism, which can choose an optimized cache implementation dynamically. Through the experiments in this article, it is to be observed that the proposed approach can achieve a significant performance gain by incorporating the SOAP processor into the SOAP engine.

DOI: 10.4018/978-1-61350-104-7.ch009

INTRODUCTION

Currently, there has been a tremendous development in the area of Web services and the simple object access protocol (SOAP) is a dominant enabling technology in it. Over the last few years, a lot of works (e.g., Bustamante, 2000; Chiu, Govindaraju, & Bramley, 2002; Elfving, Paulsson, & Lundberg, 2002; Kohlhoff & Steele, 2003) have been carried out in comparing SOAP with binary protocols, such as Java RMI and CORBA. These researches show that there is a dramatic difference in the amount of encoding necessary for data transmission, when extensible markup language (XML) is compared with the binary encoding style followed in CORBA, and all these researches have proven that SOAP, because of its reliance on XML, is inefficient compared with its peers in distributed computing.

Although the performance of Web services has been adequate for many important purposes, processing speed unfortunately remains a problem in more performance-sensitive applications, especially when the SOAP engine is built in Java. Existing researches indicate that the performance of those systems built in Java cannot be compared to the system built in C/C++. This is because the C++ compiler translates source code into an executable file, which can be executed by a operation system directly, while Java source code is interpreted into bytecode by Java virtual machine. Eckel (2003) indicates that interpreted Java runs in the range of 20 times slower than C. Hence, the Java-based systems need to improve performance, particularly.

Many performance problems of Web services are inherent in core features of XML: it is text based, flexible in format, and carries redundant information. Let us simply describe the message processing flow in the SOAP engine in general. First, when the SOAP engine receives the message, it uses XML parser to parse the message. Second, it translates the parsed XML elements into Java objects by using deserializing operations. And

then, it invokes service business logic with those objects. After that, it gets the results and translates them into a response message by using serializing operations. Finally, the response message will be sent to the user. Davis and Parashar (2002) indicate that in the overall processing flow, the XML parsing and Java reflection at runtime is exactly the performance bottleneck of Web services. Obviously, if we can avoid the time-consuming operation in the runtime, the performance may be greatly improved. There are two intuitive ways to solve this problem, one is to use a cache mechanism to avoid time-consuming operations, and the other is to deal with them before runtime.

Aiming at the problems, this article proposes a novel approach to improve the performance of Web services by optimizing SOAP processing. The SOAP engine presented in the article maintains specific SOAP processors for each Web service. The SOAP processor is generated by using the information from WSDL and service's classes during system startup or service hot-deployment. In essence, the SOAP processor is a composition of SOAP parser and deserializer. It can identify the SOAP messages related to a specific Web service. Moreover, the SOAP processor embeds several cache implementations and uses an adaptive mechanism to choose the optimized caching strategy dynamically. When receiving a message, the SOAP processor first uses an adaptive caching mechanism to choose the best cache implementation to handle the message. If the cache fails, it uses XML a preparsing framework and predeserializing templates to complete the processing. Using a service-specific SOAP processor we can avoid traditional XML parsing and Java reflection at runtime. Moreover, because the SOAP processor scans a SOAP message only once, the performance is improved remarkably. Our method supports the WSDL document/literal style, which is the direction where the industry is headed; in addition, it is the most consistent model from a Web services architecture perspective (Curbera, Duftler, & Khalaf, 2005). The evaluation presented

25 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:

www.igi-global.com/chapter/high-performance-approach-server-side/59924?camid=4v1

This title is available in InfoSci-Books, InfoSci-Multimedia Technologies, Science, Engineering, and Information Technology, InfoSci-Select, InfoSci-Computer Science and Information Technology. Recommend this product to your librarian:

www.igi-global.com/e-resources/library-recommendation/?id=1

Related Content

Semantic Enrichment for Geospatial Information in a Tourism Recommender System

Joan de la Flor, Joan Borràs, David Isern, Aida Valls, Antonio Moreno, Antonio Russo, Yolanda Pérez and Salvador Anton-Clavé (2012). *Discovery of Geospatial Resources: Methodologies, Technologies, and Emergent Applications* (pp. 133-155).

www.igi-global.com/chapter/semantic-enrichment-geospatial-information-tourism/65112?camid=4v1a

Merkle Tree Authentication in UDDI Registries

Elisa Bertino, Barbara Carminati and Elena Ferrari (2004). *International Journal of Web Services Research* (pp. 37-57).

www.igi-global.com/article/merkle-tree-authentication-uddi-registries/3040?camid=4v1a

Adoption of Web Services in Digital Libraries: An Exploratory Study

Fatih Oguz (2009). *Managing Web Service Quality: Measuring Outcomes and Effectiveness* (pp. 307-320).

www.igi-global.com/chapter/adoption-web-services-digital-libraries/26085?camid=4v1a

Model-Driven Web Service Development

Roy Gronmo, David Skogan, Ida Solheim and Jon Oldevik (2004). *International Journal of Web Services Research* (pp. 1-13).

www.igi-global.com/article/model-driven-web-service-development/3047?camid=4v1a