Using XML-Based Multicasting to Improve Web Service Scalability

Joe Tekli, Antonine University, Lebanon
Ernesto Damiani, Università degli Studi di Milano, Italy
Richard Chbeir, CNRS University of Bourgogne, France

ABSTRACT

Web services’ (WS) emphasis on open standards provides substantial benefits over previous application integration techniques. A major WS feature is SOAP, a simple, robust and extensible XML-based protocol for the exchange of messages. For this reason, SOAP WS on virtual hosts are now widely used to provide shared functionalities on clouds. Unfortunately, SOAP has two major performance-related drawbacks: i) verbosity, related to XML, that leads to increased network traffic, and ii) high computational burden of XML parsing and processing, that leads to high latency. In this paper, we address these two issues and present new results regarding our framework for Differential SOAP Multicasting (DSM). The main idea behind our framework is identifying the common pattern and differences between SOAP messages, modeled as trees, so as to multicast similar messages together. Our method is based on the well known concept of Tree Edit Distance, built upon a novel filter-differencing architecture to reduce message aggregation time, identifying only those messages which are relevant (i.e., similar enough) for similarity evaluation. In this paper we focus on recent improvements to the filter-differencing architecture, including a dedicated differencing output format designed to carry the minimum amount of diff information, in the multicast message, so as to minimize the multicast message size, and therefore reduce the network traffic. Simulation experiments highlight the relevance of our method in comparison with traditional and dedicated multicasting techniques.

Keywords: Differential Processing, Message Multicasting, SOAP, SOAP Performance, Web Service Communications, XML

1. INTRODUCTION

Web Services – WS – have been proposed as a key technology for systematic and flexible application-to-application integration. Today, WS provide a comprehensive solution for representing, discovering and invoking services in a wide variety of virtualized architectures.

DOI: 10.4018/jwsr.2012010101

Based on Google’s web service SOAP request and response WSDLs1, using an adaptation of IBM’s XML document generator2

Here we focus on XML Web services, i.e. the ones that utilize message formats based on XML (Bray T., Paoli J. et al. 2008). This technology builds on two XML schemata: WSDL – Web Service Description Language (Chinnici R., Moreau J.J. et al. 2007) supporting the machine-readable description of a service’s interface, and SOAP – Simple Object Access
Protocol (W3 Consortium 2007) dictating the messages’ format, with bindings to existing protocols (e.g., HTTP, FTP, SMTP, etc.) for SOAP message negotiation and transmission. As a result, WS can rely on existing XML parsers for automatic validation of messages. Also, the easy extensibility of XML schemata allows integration mechanisms to evolve, as markets require new functionalities, without causing incompatibilities and fragmentation of protocols.

Today, many SaaS—Software-as-a-Service—applications are based on SOAP Web services. Client applications invoke WS executed in virtualized infrastructures; this way, underlying physical servers’ capacity can be dynamically assigned to services, enabling innovative pay-as-you-go revenue schemes. In a multi-tenant, virtualized setting, WS receive message flows coming from multiple applications at the same time; this makes non-functional requirements even more stringent. These requirements include WS performance, security and reliability, which are closely related to XML processing. In particular, WS have inherited a major XML drawback, verbosity, which strongly affects WS performance. SOAP message exchanges are quite elaborate; the client program has to build the skeleton of the XML message, put the right values in it (serialization), and then send it to the remote service. In turn, the remote service parses the message, digs out the data it needs (de-serialization), and then goes through the same procedure to generate an XML reply. No wonder, then, that SOAP message processing produces considerable network traffic and causes higher latency than competing technologies (Kohlhoff C. and Steele R. 2003; Suzumura T., Takase T. et al. 2005). High latency becomes more critical when handling large volumes of SOAP-based communications such as with cloud-based e-science (Gannon D., Krishnan S. et al. 2004) and e-business (Singh G., Bharathi S. et al. 2003) applications.

In the context of multi-tenant virtualized applications, however, the same WS are invoked by a high number of clients executing different applications. Intuition suggests that this scenario will increase the likelihood of WS receiving large numbers of similar SOAP messages.

Hence, similarity and differential encoding have been often proposed to enhance SOAP performance, aiming to: (i) reduce processing time—in parsing (Makino S., Tatsubori M. et al. 2005; Takeuchi Y., Okamoto T. et al. 2005; Teraguchi M., Makino S. et al. 2006), in serialization (Devaram K. and Andersen D. 2002; Abu-Ghazaleh N., Lewis M.J. et al. 2004), and in de-serialization (Abu-Ghazaleh N. and Lewis M.J. 2005; Suzumura T., Takase T. et al. 2005), and to (ii) reduce network traffic—via compression (Werner C., Buschmann C. et al. 2005) and multicasting (Phan K.A., Tari Z. et al. 2008; Azzini A., Marrara S. et al. 2009; Phan K.A., Bertok P. et al. 2009). This is based on the observation that SOAP exchanges often involve highly similar messages since those created by the same implementation usually have the same structure, and those sent from a server to multiple clients tend to show similarities in structure and content (e.g., stock quote services (Phan K.A., Tari Z. et al. 2008), online booking and meteorological broadcast services (Azzini A., Marrara S. et al. 2009), etc.). In this paper, we focus on SOAP multicasting, as a technique to save network bandwidth by delivering SOAP messages to a group of destinations simultaneously (Zhang B., Jamin S. et al. 2002).

To our knowledge, the only approach to SOAP multicasting was described in (Phan K.A., Tari Z. et al. 2008), where the authors introduce SMP (Similarity-based Multicasting Protocol), identifying, indexing and routing similar SOAP messages together (cf. Section 2). SMP’s main contribution consists in grouping and transmitting together similar SOAP messages, in comparison with identical-only message aggregation of traditional network-layer (e.g., IP) multicasting (Zhang B., Jamin S. et al. 2002). Nonetheless, careful analysis of (Phan K.A., Tari Z. et al. 2008) led us to pinpoint certain aspects which limit both the effectiveness and efficiency of SMP multicasting. On one hand, while SMP considers the common and distinctive parts of SOAP messages in multicast message encoding, it does not always generate
Enhancing the Testability of Web Services
Daniel Brenner, Barbara Paech, Matthias Merdes and Rainer Malaka (2009).
*Managing Web Service Quality: Measuring Outcomes and Effectiveness* (pp. 223-244).
[www.igi-global.com/chapter/enhancing-testability-web-services/26081?camid=4v1a](www.igi-global.com/chapter/enhancing-testability-web-services/26081?camid=4v1a)

Dynamically Adaptable Web Services Based on the Simple Object Access Protocol
[www.igi-global.com/chapter/dynamically-adaptable-web-services-based/26913?camid=4v1a](www.igi-global.com/chapter/dynamically-adaptable-web-services-based/26913?camid=4v1a)