



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 1, January 2014

A Study on Web Services Performance Measurements and Its Techniques

Mathavi parvathi¹, Pandimurugan², Dr.Paul rodrigues³

Assistant Professor, Hindustan University, Chennai, India ^{1,2}

Dean (Research), Velammal Engineering College, Chennai, India³

ABSTRACT: Web services can be viewed as a rapidly evolving application-integration technology that allows applications to interact or communicate with each other irrespective of the platform they are running on or the programming language they have been written in. In short, web services are both platform as well as programming language .The Internet and Web Technologies have changed the way organizations do business. Nowadays, businesses are willing to put their core business processes on the Internet as a collection of web services. In this paper we are analyzed web services performance factors and its techniques as a survey. Here we concentrate on the gap identify between the scalable factors of web services based on that we try to improve the performance.

Keywords: Web service, composition, XML, web service factors.

I INTRODUCTION

The Internet and Web Technologies have changed the way organizations do business. Nowadays, businesses are willing to put their core business processes on the Internet as a collection of web services [1]. A web service can be defined as "an interface that describes a collection of operations that are network-accessible through standardized XML messaging". They are quickly becoming a significant technology in the evolution of the Web and distributed computing. Web services can be viewed as a rapidly evolving application-integration technology that allows applications to interact or communicate with each other irrespective of the platform they are running on or the programming language they have been written in. In short, web services are both platform as well as programming language independent. One of the critical factors for the success of web services is that they leverage well-established technologies such as HTTP and XML. They allow developers to create loosely coupled applications based on XML messaging protocols very easily by leveraging the data independence of XML to solve enterprise integration problems, both inside and outside the firewall. Web service interfaces also act like wrappers that can map to any type of software program, middleware system, database management system, or packaged application. They provide a layer of abstraction necessary for bringing together highly distributed and heterogeneous systems. Individual web services are self-contained, network-accessible applications. An individual web service is limited in its capability because it provides only a specific functionality. Because of their strong potential as a business solution to enterprise application integration, more and more businesses will make their core processes available as web services and will also be ready to adopt web services for their own business needs. As new web services are developed all the time, multiple services exist that provide similar functionality. Selecting an appropriate service with the right capability is very important. In such a scenario, Quality of Service becomes a significant factor in differentiating between various service providers. The Quality of Service offered by an individual web service can decide the usage and popularity of that service. Quality 2 of service refers to the non-functional attributes of a web service such as performance, reliability, availability and security. Web services mainly used for the purpose of communicating with each other and clients. Web services can transfer the data to others behind firewall. Web service shares business logic, data and processes through interface over network. Now developers can add this service to their GUI to offer that service.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 1, January 2014

Web services permits numerous applications from several resources to communicate. Web services are not tied to any operating system or any programming language. Meeting Performance Requirements is an important issue towards acceptability of given software? The performance of the SOA is primarily studied as web service performance, since the web service is one of the main supporting technologies of SOA.

II WEB SERVICE COMPOSITION

It provides an open, standards-based approach for connecting web services together to create higher-level business processes.[4] Standards are designed to reduce the complexity required to compose web services, hence reducing time and costs, and increase overall efficiency in businesses.

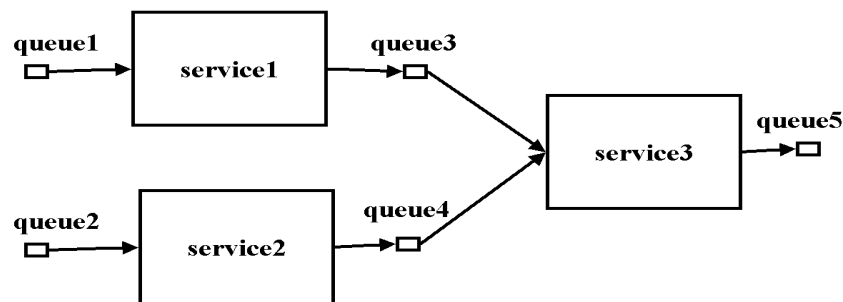


Figure 1: Web Service composition

As composition involves two or more different vendor's services to be combined, the composition becomes very complex task. There are lot many composition techniques available for web service composition. In this paper, we have discussed some of them. To understand the web service composition, we need to understand the issues involved in web service composition. The issues are as follows. 1. Co-ordination: In the process of web service composition, in order to combine two or more web services, we need to make some co-ordination amongst the web services. 2. Transaction: It is very important to add transaction protocol in co-ordination to provide atomic transaction. Sometimes we need to add the Atomicity, Consistency, Integrity and Durability (ACID) properties to ensure the safe transaction. 3. Context: It is very important issue which needs to take care. Context can be defined as the information used by the web service during composition to provide output and execution results. 4. Selection: The selection process of a proper web service plays a vital role in web service composition. There are lot many web services available with the same functionality and same characteristics. Based on QoS (Quality of Service), how we can select a proper web service is also a major issue. Whenever we talk about web service composition, we look for the word web service selection as without web service selection, we cannot go for composition.

III WEB SERVICES TECHNOLOGIES

The following technologies have been standardized to implement web services. [3] Each technology targets one specific layer in the web services architecture stack. These technologies are: XML (Extensible Markup Language), WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol), and UDDI (Universal Description, Discovery and Integration). Other technologies may also be used.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 1, January 2014

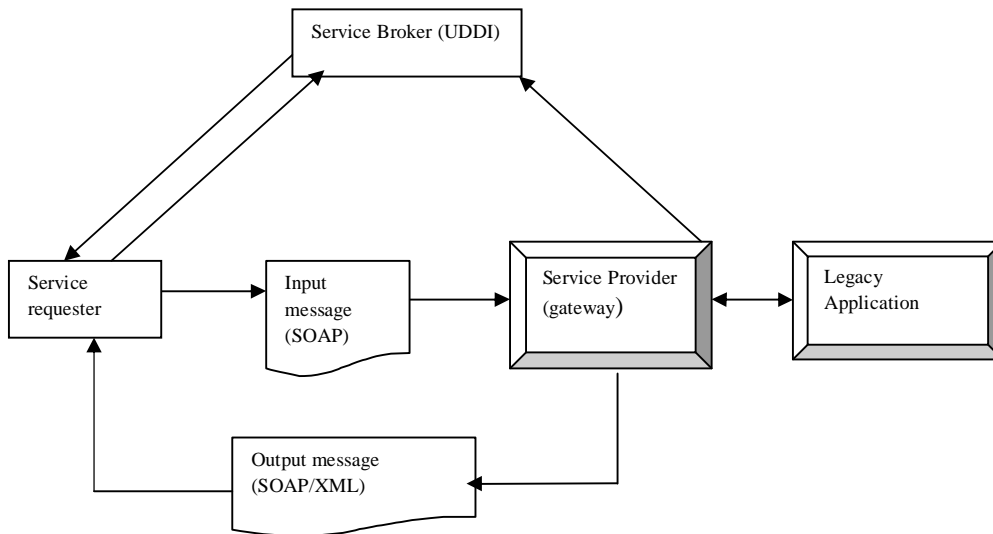


Figure 2: Web Service architecture

IV XML: EXTENSIBLE MARKUP LANGUAGE

Extensible Markup Language or XML is a universal text-based standard for representing and exchanging documents on the Internet or on Intranets. XML is a meta-language that can be used to create other specialized markup languages. XML allows users to create their own customized tags and document structures, thereby enabling the definition, transmission, validation, and interpretation of documents and data between different applications and between organizations. Since XML data is stored in plain text format, it provides a software- and hardware-independent way of sharing data across different platforms, operating systems and languages.[6]An XML Parser is required to read and manipulate XML documents. The parser extracts the actual data out of the XML document and creates new data structures from them. Parsers also check whether documents conform to the XML standard and have a correct structure. This is an essential for the automatic processing of XML documents. The correct document structure can be defined by using XML Schemas. XML Parsers are available in different languages. XML is also a core technology in the development of Web services.

V WSDL: WEB SERVICES DESCRIPTION LANGUAGE

Web Services Description Language or WSDL is an XML document used to describe web services. The WSDL document describes the following critical pieces of information about a given web service: all the publicly available functions of the web service, the data types used in the message requests and responses, the address where the service can be accessed and the protocol binding information. [2] Using the information in the WSDL document a client can locate and invoke the functions of a web service. [11]WSDL document can also contain other elements, like extension elements and a service element that makes it possible to group together the definitions of several web services in one single WSDL document.

VI SOAP: SIMPLE OBJECT ACCESS PROTOCOL

Simple Object Access Protocol or SOAP is the industry standard protocol for XML-based messaging which provides users with a transmission framework for inter-application or inter-service communication via HTTP.[9] The SOAP

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 1, January 2014

specification establishes a standard message format that consists of an XML document capable of hosting RPC and document-centric data. A SOAP message is actually an HTTP POST message with an XML envelope as a payload. It is a lightweight protocol based on XML and can be used to exchange structured messages between different applications that are distributed over a network.

A SOAP message consists of the following parts:

- An envelope that identifies the document as a SOAP message,
- An optional Header element that contains header information,
- A required Body element which contains a set of encoding rules and a convention to represent RPCs and responses,
- An optional Fault element used to provide information about errors encountered while processing the message.

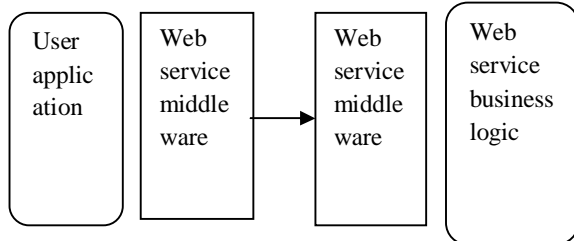


Figure 3: SOAP Messaging

The SOAP envelope is used for describing what the message actually contains. [8] SOAP also defines data encoding rules. The set of encoding rules are used to express data types defined by applications. SOAP data encodings specify rules for object serialization; that is, mechanisms for marshaling and unmarshaling data streams across the net. SOAP can be used in combination with a variety of network protocols like HTTP, SMTP, FTP and RMI over IIOP.

VII UDDI: UNIVERSAAL DESCRIPTION ,DISCOVERY AND INTEGRATION

UDDI defines a way to publish and discover information about web services. The UDDI business registry is used to store information about web services. [7] The business registry is actually an XML file that describes a business and its web services. Service providers register themselves and the services that they provide in a UDDI registry by publishing the service descriptions. The UDDI business registry classifies the services under different categories. Service requestors then search a registry for the type of service that they are looking for. Once they find a service, they use the service descriptions to contact the provider of that service and bind with or invoke that service.

A: A UDDI business registration consists of the following components:

- Business Description or White Pages – Basic contact information about a company such as address, contact information, urls, and other known identifiers; This information allows clients to discover a web service based on business identification.
- Business Service or Yellow Pages - Industrial categorizations based on standard taxonomies; this information allows clients to identify web services based upon its category.
- Binding Template or Green Pages – This gives technical information about services exposed by the business such as service entry point and other implementation details.
- tModel – A tModel describes the web service interfaces and provides pointers to specifications that describe their implementation.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 1, January 2014

B: Factors affecting the performances of Web services

Web services is remote computing, so its performance is affected by many factors, including latency and bottlenecks in the network; intermediaries (if present); and your service provider.[5] Some of these factors are out of our control, especially network servers not under our control.

- **TRANSMISSION:** The longer the SOAP message, the longer the transmission time.
- **PROCESSING:** The parsing, binding, validation, and transformation of XML/SOAP messages will add to the response time.
- **PERSISTENCE:** RDBMS/XML databases— the databases referenced by XML—must stay “resident” for the life of the SOAP message. This might imply that the XML message might have to “live” beyond the life of a process.
- **PROACTIVE STRATEGY:** Involves training engineering and QA staff on writing optimized XML Web services
- **DEFINITIVE STRATEGY:** Follow irrespective of the stringent performance requirements
- **REACTIVE STRATEGY:** Defining and addressing performance-oriented requirements right in the design phase; i.e., software performance engineering (SPE). SPE is the discipline that identifies best practices and necessary hardware and software for leveraging performance. The reactive strategy involves analyzing bottlenecks and weak links that may be causing performance problems. Once issues are identified, they are resolved by both analyzing and tweaking the code, or by changing the design.

IX PERFORMANCE AND SCALABILITY ISSUES

The main issues that can adversely affect the performance and scalability of our Web services are summarized in the following list.

- **Incorrect communication mechanism.** Currently, there are three main technologies for remoting a method call: Enterprise Services, .NET remoting, and ASP.NET Web services. The best choice depends upon various factors, including the source and target platforms, whether you need to communicate across an intranet or the Internet, whether you require additional services such as distributed transactions, your security requirements, deployment considerations (such as whether your communication must pass through a firewall), other port limitations, and so on.
 - **Web services.** Use Web services to build your services.
 - **Enterprise Services.** If you use Web services to build your services, you may still need to use Enterprise Services within your service implementation. For example, you may need it to support distributed transactions or if you want to use object pooling.
 - **.NET remoting.** Use remoting for same-process, cross-application domain communication or for remote communication if you need to integrate with a legacy protocol. If you use remoting, avoid custom proxies, custom sinks, and using contexts. This helps to avoid compatibility issues with future communication technologies.
- **Chatty calls.** Network round trips to and from a Web service can be expensive. This issue is magnified if clients need to issue multiple requests to a Web service to complete a single logical operation.
- **Improper choice of parameters.** Your choice of parameters depends upon various factors, such as interoperability, the varying platforms used by the clients, maintainability, the type of encoding format used, and so on.[12] Improper choice of parameters can lead to a number of issues, including increased serialization costs and potential versioning problems for the Web service (for example where a custom type is updated). Where possible, you should use primitive types. If interoperability is an issue, consider using the **XmlElement** and **XmlDocument** types and choose types specific to your application, such as an **Employee** or **Person** class.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 1, January 2014

- **Serialization.** Serializing large amounts of data and passing it to and from Web services can cause performance-related issues, including network congestion and excessive memory and processor overhead.

Other issues that affect the amount of data passed across the wire include improper data transfer strategies for large amounts of data. Selecting an appropriate data transfer strategy — such as using a SOAP extension that performs compression and decompression or offloading data transfer to other services — is critical to the performance of our Web services solution.

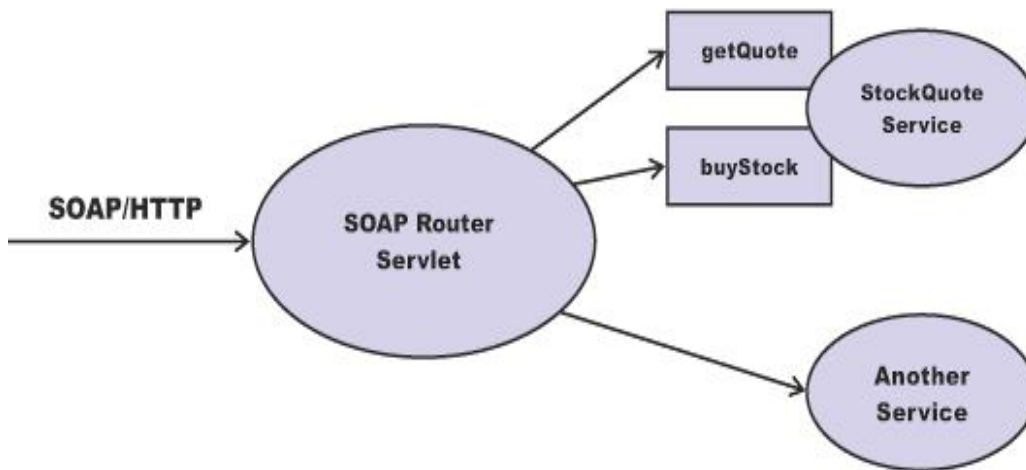


Figure 4: Web services cache

Improper choice of encoding format. You can use either literal or SOAP encoding. SOAP encoding involves more SOAP-processing overhead as compared to literal encoding.

- **Lack of caching or inefficient caching.** In many situations, application or perimeter caching can improve Web services performance. Caching-related issues that can significantly affect Web services performance include failure to use caching for Web methods, caching too much data, caching inappropriate data, and using inappropriate expiration settings.
- **Inefficient state management.** Inefficient state management design in Web services can lead to scalability bottlenecks because the server becomes overloaded with state information that it must maintain on a per-user basis. Common pitfalls for Web services state management include using stateful Web methods, using cookie container-based state management, and choosing an inappropriate state store. The most scalable Web services maintain no state.
- **Misuse of threads.** It is easy to misuse threads. For example, you might create threads on a per-request basis or you might write code that misuses the thread pool. Also, unnecessarily implementing a Web method asynchronously can cause more worker threads to be used and blocked, which affects the performance of the Web server.

On the client side, consumers of Web services have the option of calling Web services asynchronously or synchronously. Your code should call a Web service asynchronously only when you want to avoid blocking the client while a Web service call is in progress. If you are not careful, you can use a greater number of worker and I/O threads,



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 1, January 2014

which negatively affects performance. It is also slower to call a service asynchronously; therefore, you should avoid doing so unless your client application needs to do something else while the service is invoked.

- **Inefficient Web method processing.** A common example of inefficient processing is not using a schema to validate input upfront. This issue can be significant because the Web method may de-serialize the incoming message and then throw exceptions later on while processing the input data.

X CONCLUSION

In this paper we have surveyed the some technologies of web services and its uses. Special attention has been devoted to identify gap in the web services by using scalable metrics, based on that we will try to improve the performance of web services. We learned some difficulties and drawbacks from previous systems, so we will do a brief research in scalable metrics related to web services and how we can provide a quality based service for improving web performance. Finally we decided to concentrate on scalable factors and metrics types and its measurements for improving performance in the web field. We are persuaded that this class of approaches will get an even greater importance and attention within the research community in the next months.

REFERNCES

- [1] Anura Guruge, (2004) "Web Services: Theory and Practice", Digital Press.
- [2] Booth D and Liu C.K, (2007) "Web Services Description Language (WSDL)", Version 2.0 Part 0: Primer. W3C Recommendation, <http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626>.
- [3] K. Gottschalk, S. Graham, H. Kreger, and J. Snell "Introduction to Web services architecture", IBM Systems Journal, Volume 41, Number 2, 2002, New Developments in Web Services and E-commerce, pp. 2-3.
- [4] Mark Carman, Luciano Serafini, and Paolo Traverso, "Web Service Composition as Planning," presented at Proceedings of ICAPS 2003 Workshop on Planning for Web Services, Trento, Italy, 2004.
- [5] Jinghai Rao and Xiaomeng Su, "A Survey of Automated Web Service Composition Methods," Semantic Web Services and Web Process Composition: First International Workshop, SWSWPC 2004, vol. 3387 / 2005 no. pp. 43, 2004.
- [6] Tim Bray, et al. Extensible Markup Language (XML) 1.0 (Second Edition). October 2000. <<http://www.w3.org/TR/REC-xml>>.
- [7] Naveen Srinivasan, Massimo Paolucci and Katia Sycara: Adding OWL-S to UDDI, implementation and throughput. In Proc. 1st Intl. Workshop on Semantic Web Services and Web Process Composition (2004) 6–9.
- [8] D. Box et al. Simple Object Access Protocol (SOAP) 1.1. Online:<http://www.w3.org/TR/SOAP/>, 2001.
- [9] M. P. Papazoglou and J. jacques Dubray, "A survey of web service technologies," tech. rep., University of Trento, Jun. 2004.
- [10] SemanticWeb Services Language (SWSL), [Online] <http://www.daml.org/services/swsl/>.
- [11] Web Service Semantics (WSDL-S), [Online] <http://www.w3.org/Submission/WSDL-S/>.
- [12] S B Patil, Sachin Chavan, Preeti Patil; "High Quality Design And Methodology Aspects To Enhance Large Scale Web Services", International Journal of Advances in Engineering & Technology, 2012, ISSN: 2231-1963.