

AN OVERVIEW OF METHODS FOR MONITORING WEB SERVICES BASED ON THE QUALITY OF SERVICES

Sara Zirak¹, Naser Nematbakhsh², Kamran Zamanifar³

¹Sara Zirak, Department of Computer, Najaf abad University, Isfahan, Iran

²Naser Nematbakhsh, Department of Computer, Najaf abad University, Isfahan, Iran

³Kamran zamanifar, Department of Computer, Najaf abad University, Isfahan, Iran

Abstract

Service-oriented architecture is an approach for making distributed systems that provide their services in the form of software applications. These services are called by other software and are used to make new services as well. Therefore, service quality is one of important issues in service-oriented architecture, whereas this issue has a considerable impact in selecting and composition of services to be presented to the users. In this area and in order to update data obtained from web service operations, continuous monitoring of non-functional properties is of particular importance. In this paper, the existing techniques in monitoring process are classified and discussed in detail.

Keywords: Monitoring, Quality parameters, Quality of service, Web service

1. INTRODUCTION

Nowadays, service-oriented architecture is taken into consideration as a suitable approach to increase the efficiency of systems. The approach, as a preferred approach in the field of architecture, helps enable organizations to adapt quickly their business needs with IT, and respond effectively to the demands of the market. With service-oriented approach, it can provide solutions which are not limited to the boundaries of organization, company or department. This architecture is implemented by the Web services. In fact, the web services allow exchange between different computers in a heterogeneous environment consists of various systems.

Here, an important problem is the quality of web services in order to increase their efficiency and quality. The issue of monitoring of web service quality parameters arises to achieve this important goal. Monitoring of quality parameters is a way to collect data related to the quality parameters that is achieved through a variety of mechanisms aimed to provide the best quality. In section 2, the process of monitoring of web services will be described. Then, in Section 3, various monitoring techniques will be classified into groups and discussed in detail. Finally in the conclusion section (Section 4), a number of advantages of monitoring process will be reviewed.

2. MONITORING OF WEB SERVICES

The term monitoring is widely used in the design and engineering of service-oriented systems. In the life cycle of a system, there is a monitoring process which plays the role of collecting information. Monitoring may be defined as follows:

"A process of collecting and reporting relevant information about the implementation and development of services in the service oriented architecture" [1]

The aims of monitoring can be mentioned as follows:

- Correcting the system in runtime: the system implementation and its parameters should be checked in order to identify the right or wrong positions. Hence, it is able to analyze whether the information at the run time correspond with the values expected or not? In addition, this information is needed to analyze the service level agreement requirements and to compare system's specification
- Completing detection and fixing certain errors: monitoring is necessary to complete detection and fixing the specified errors. For example, if an error is detected and an action is taken to eliminate the error, monitoring should be performed to ensure the operation
- In commercial activities: commercial activities are seen greatly in service-oriented architecture. By this process, all the quality and the performance parameters of commercial activities derived from combinations of services could be monitored and real data could be presented. The data obtained from monitoring may lead to the conclusion that the yield of a service is not acceptable and it is replaced with another alternative to improve the business activity performance.

Furthermore, the service applicants, by using the information obtained from monitoring, will check the values with their

expected values, which are the same values as entered in service level agreement. The result will determine exactly the level of customer satisfaction from the service. This information is very useful for the service provider as well, because with this information he can adapt himself to the new conditions and cover the weaknesses resulted from the monitoring.

Finally, the monitoring can be used to detect problems and service provider could, if necessary, reconsider the allocation of its resources to the services by observing the data collected from monitoring. Hence, a different service implementation could be provided. In the following section we will describe different methods existed for monitoring [2].

3. EXISTING METHODES FOR MONITORING OF QUALITY PARAMETERS

A number of methods has been proposed for quality parameters that are different in place and level of implementation. All methods of monitoring of parameters are implemented at three levels of service, communication and coordination. Monitoring of quality parameters may be performed on the client side, server side, or on third-party side like the bass. In this section, we examine the monitoring action levels. All the existing methods of monitoring of quality parameters will be discussed in one of the three mentioned sides and will be presented by one of the three approaches presented.

3.1 Approaches to Monitoring

3.1.1 Service-level Monitoring Approach

Service-level monitoring approach is the most basic approach is the monitoring of quality parameters. This approach is one of the first solutions to get the values of quality parameters of web hosting service. In this approach, the monitor code is entered into the client code and the web service provider .For example, putting a timer on the client code to register a request time and its receiving time, or by encapsulating in the operational functions of the client/web service provider that are inspired by aspect-oriented architecture.

Monitoring of quality parameters in the service level is among the most common methods in monitoring of quality parameters which is possible by coding on the client side or the server side or both of them. To further explain this matter, we will pay attention to measure the response time parameter using service level monitoring approach. A simple solution for measuring characteristics of a web service is to develop the proxy service functions in addition to their respective codes. Webproxies service contains codes that the binding conditions are specified in the service interface. Thereby, the communication network complexities and their details remain concealed from the customer. For example, if SOAP is used as

binding, the proxy service will include special SOAP codes that will be used for a service call. Here, the example given as the response time measurements is explained:

To measure the response time, the following steps should be performed on the proxy service:

1. Generating a proxy service by WSDL service definition file
2. Correcting the generated code for the proxy server by adding the code related to the added timer
3. Recompiling to correct possible errors
4. Developing the client's program to create an object from the service proxy and calling the appropriate methods.

The above-mentioned steps will be explained in detail as follows:

1) Proxy service is not coded by the programmer, but it can be easily generated by WSDL definition file. The code defined by the WSDL can be seen in the following figure.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EchoService"

targetNamespace="http://www.echoserviceser
vice.com/EchoService-interface"
xmlns="http://schemas.xmlsoap.org/wsdl/"

xmlns:soap="http://schemas.xmlsoap.org/wsd
l/soap/"

xmlns:tns="http://www.echoserviceservice.c
om/EchoService"

xmlns:xsd="http://www.w3.org/1999/XMLSchem
a">
<message name="InechoRequest">
<part name="meth1_inType1"
type="xsd:string"/>
</message>
<message name="OutechoResponse">
<part name="meth1_outType"
type="xsd:string"/>
</message>
<portType name="EchoService">
<operation name="echo">
<input message="InechoRequest"/>
<output message="OutechoResponse"/>
</operation>
</portType>
<binding name="EchoServiceBinding"
type="EchoService">
<soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap
/http"/>
<operation name="echo">
<soap:operation
soapAction="urn:echoservice-service"/>
```

```

<input>
<soap:body

encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
namespace="urn:echoservice-service"
use="encoded"/>
</input>
<output>
<soap:body

encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
namespace="urn:echoservice-service"
use="encoded"/>
</output>
</operation>
</binding>
<service
name="EchoService">
<documentation>IBM WSTK 2.0 generated
service definition file</documentation>
<port binding="EchoServiceBinding"
name="EchoServicePort">
<soap:address
location="http://localhost:8080/soap/servlet/rpcrouter"/>
</port>
</service>
</definitions>
2) The generated code is for public use and we should add a
time to measure the response time. In the following code, a
timer is added for measuring response time.
import java.net.*;
import java.util.*;
import org.apache.soap.*;
import org.apache.soap.encoding.*;
import org.apache.soap.rpc.*;
import org.apache.soap.util.xml.*;
import mytimer.Timer;
public class EchoServiceProxy
{
private Call call = new Call();
private URL url = null;
private String SOAPActionURI = "";
private SOAPMappingRegistry smr =
call.getSOAPMappingRegistry();
public EchoServiceProxy() throws
MalformedURLException
{
call.setTargetObjectURI("urn:echoservice-
service");
call.setEncodingStyleURI("http://schemas.x
mlsoap.org/soap/encoding/");
this.url = new
URL("http://localhost:8080/soap/servlet/rp
crouter");

```

```

this.SOAPActionURI = "urn:echoservice-
service";
}
public synchronized void setEndPoint(URL
url)
{
this.url = url;
}
public synchronized URL getEndPoint()
{
return url;
}
public synchronized java.lang.String echo
(java.lang.String meth1_inType1)
throws SOAPException
{
if (url == null)
{
throw new
SOAPException(Constants.FAULT_CODE_CLIENT,
"A URL must be specified via " +
"EchoServiceProxy.setEndPoint(URL).");
}
call.setMethodName("echo");
Vector params = new Vector();
Parameter meth1_inType1Param = new
Parameter("meth1_inType1",
java.lang.String.class,
meth1_inType1, null);
params.addElement(meth1_inType1Param);
call.setParams(params);

// Start a Timer
Timer timer = new Timer();
timer.start();

Response resp = call.invoke(url,
SOAPActionURI);

// Stop the Timer
timer.stop();
// Print the response time by
calculating the difference
System.out.println("Response Time = " +
timer.getDifference());

// Check the response.
if (resp.generatedFault())
{
Fault fault = resp.getFault();
throw new
SOAPException(fault.getFaultCode(),
fault.getFaultString());
}
else
{

```

```

    Parameter retValue =
resp.getReturnValue();
return
(java.lang.String)retValue.getValue();
}
}
}

```

- 3) Compiling the code to detect and fix potential errors.
 - 4) Developing the client code to call the Web service.
- By implementing these four steps, we can obtain the value of the response time qualitative parameter.

As it is seen, this approach is inspired from the aspect-oriented architecture [3].

In monitoring of quality parameters in the service level, using aspect-oriented architecture has the first letter. In this method, different objects are used to measure the quality parameters. Other methods by calling these objects begin to record and measure the service quality parameters [4].

3.1.2 The Communication Level Monitoring Approach

In this approach, the monitoring is performed by interpreting the messages exchanged between customers and web service providers. Messages are exchanged in web services by SOAP, HTTP, TCP/IP protocols. In these methods, by analyzing the communication layer messages exchanged, it is attempted to collect data on quality parameters. Thereby, the values of quality parameters of a web service could be obtained [5.6]. In reference [7], monitoring of the quality parameters through analyzing the HTTP + TCP / IP communication protocol is performed.

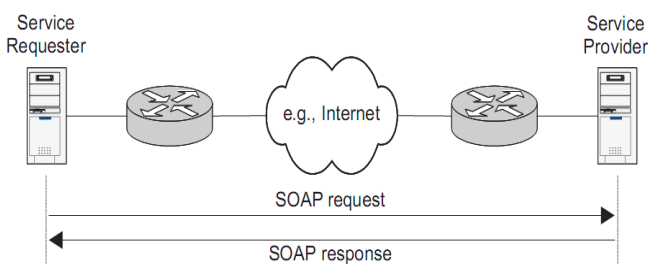


Fig -1: Web service interactions

Figure 1 shows the communication layers existed in a simple relationship between the server and client. Service applicator generates a SOAP request message and sends it to service provider to receive service using the HTTP. This message is being passed by several systems between the two sides. Service provider sends the SOAP response message through HTTP. In the sending and receiving mode, all events encountering sending of data may affect the performance of web services. The problems include overcrowding, lack of

access to the host and the routing problems. A number of these problems can be solved by modern communication protocols. The communication protocols provide some information in which could be used to monitor quality parameters of web services [7].

3.1.3 Orchestration Level Monitoring Approach

In this approach, the monitoring of quality parameters is carried out on BPEL services using aspect-oriented architecture done.

In reference [8], the monitoring of quality parameters supports the sample monitoring and class. Using the aspect-oriented architecture makes the business logic of service separate from the logic of monitoring.

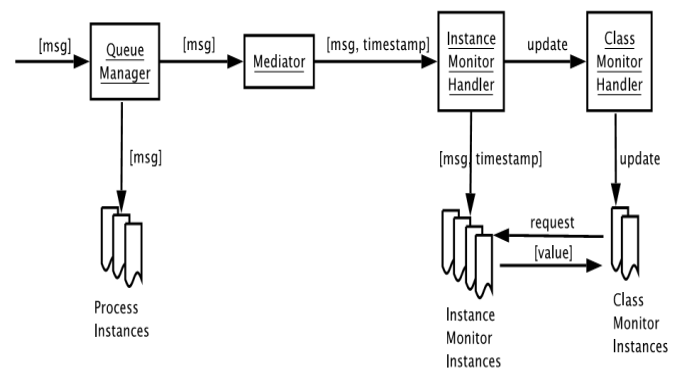


Fig -2: Message flow in run-time monitoring

For example, sample monitoring is performed to obtain the behavior of database web services and class monitoring is performed to obtain the behavior of online payment service websites.

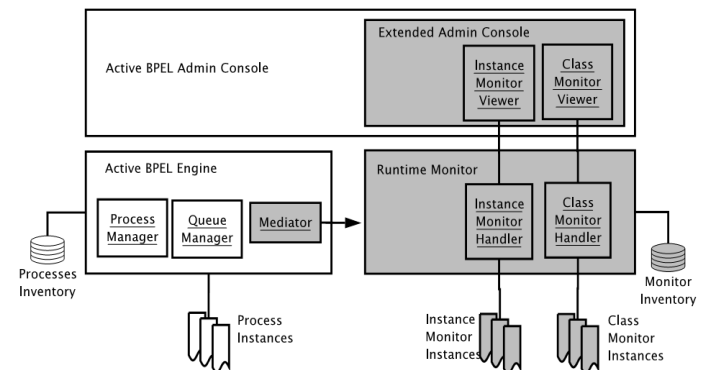


Fig -3: The Active BPEL engine developed by media monitoring in the runtime

Here, the monitoring of quality parameters is performed by aspect-oriented architecture and the values of quality parameters are determined by analyzing the input/output

messages. The input/output data are sent and received by processes [8].

3.2 Monitoring Practice Locations

3.2.1 Service-level Monitoring of Quality Parameters on the Client _ Side

One way to monitoring quality parameters, is the implementation on the client side. The latter is carried out by using added tools on the client side. In this section, one of these tools is studied.

Webinject is a tool on the client side that determines the actual response time of web services by sending requests of web service and analyzing the response time. Also, by counting the number of successful and unsuccessful applicants, it calculates the reliability. In this device, three parameters including response time, reliability and security are analyzed.

The tool calculates the response time of service by sending service request to the server and receiving the time parameter service. For this purpose, a statistical timer calculates the response time through recording the sending and receiving times. To calculate the reliability parameter, the tools counts the number of successful and unsuccessful requests. The reliability percentage is obtained by dividing the number of unsuccessful requests to the number of successful requests. This tool is written in the Perl programming code, and is compatible with all environments [9].

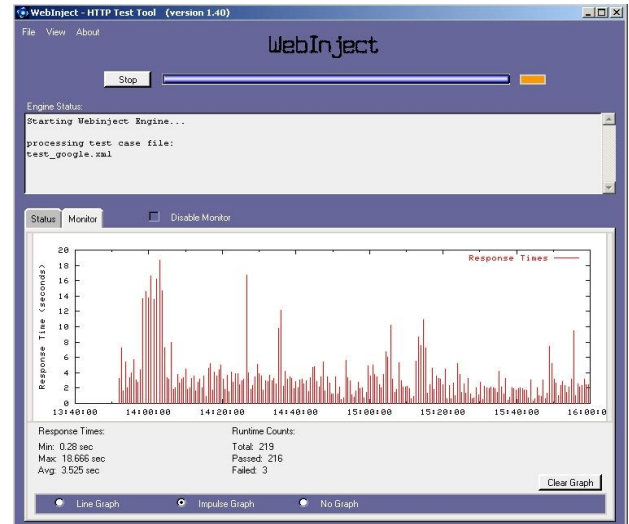


Fig -4: The response time measured by WebInject software

3.2.2 Monitoring of Quality Parameters on the Server_ Side

In this method, monitoring is performed on the server side, so that the server will attempt to calculate the actual values of quality parameter through recording the sending and receiving times, the requests of counting successful and unsuccessful responses to the client. The method of monitoring of quality parameters on the client side is most popular.

Sandcat tool is one of the tools available in this area. This tool can provide a web service and website monitoring. This means lists some of the security weaknesses of web service or web site. Among the advantage of this tool is its high flexibility. This tool can quickly adjust itself to the web media. [10].

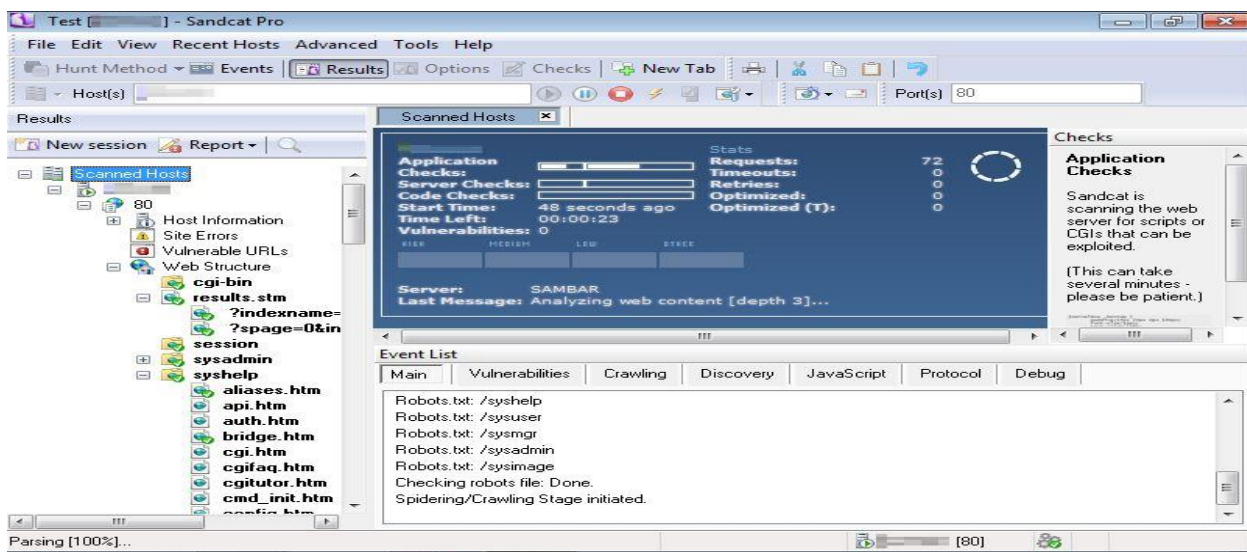


Fig -5: Working media of Sandcat software

Another tool in this field is Soapknox software which measures the response time, reliability and security parameters. This tool performs monitoring in the communication layer level, through monitoring SOAP messages. The communication messages exchanged between customers and provider contain information about the time of sending request and time of service received which are used in calculating the response time. SOAP messages contain safety points in their header by monitoring of which, the tool can determine the values of security parameters [10].

Using monitoring of quality parameters, this tool follows up improvements in operational productivity (reducing operating costs, increasing the accuracy of decision making), and strategic leadership based on real data [11].

Besides this tool, there is also other software that performs monitoring of quality parameters on the server side [12, 13].

3.2.3 Monitoring of Quality Parameters on the Third_part_Side

The method of monitoring of quality parameters on third-part side such as BUS, is the most popular solution for monitoring. Monitoring is often performed in BUS using the relevant components. Thereby, the monitoring components in the BUS collect data and calculate the actual value of the quality parameter. As BUS is an interface in all of operations, it accesses to lots of data. The values of quality parameters can be calculated by extracting the data. Because BUS is commonly used in the service oriented architecture, the monitoring in BUS is popular. The task of data collection to determine the values of quality parameters is performed by the components in BUS [14].

4. CONCLUSIONS

The aim of different methods of monitoring of quality parameters is to discover a method with the lowest cost and highest efficiency, while the actual values of quality parameters could be obtained with reasonable accuracy. With a suitable method to monitoring the quality parameters, strategic leadership based on real data, increased accuracy in decision making, improved operations and reduced costs in operations could be achieved. The monitoring of quality parameters to achieve service-oriented architecture is necessary whereas all available methods are trying to present a more optimized and more helpful monitoring.

The methods studied were different at place and level of implementation of monitoring, therefore, the customer could choose an appropriate method according to his/her needs.

REFERENCES

- [1]. Bianco,P.,Lewis, G. and Merson,P." Service Level Agreements in Service-Oriented Architecture Environments ".2008
- [2]. State of the Art Report, Gap Analysis of Knowledge on Principles, Techniques and Methodologies for Monitoring and Adaptation of SBAs Salima Benbernou 10 July 2008.
- [3]. Mani, A. and Nagarajan ,A." Understanding quality of service for web services Technical report, IBM DeveloperWorkswww.106.ibm.com/developerworks/webservice/library/ws-quality.html, Jan 2002 .
- [4]. Rosenberg,F., Platzer, C. and Dustdar, S." Bootstrapping performance and dependability attributes ofweb services." In ICWS '06: Proceedings of the IEEE International Conference on Web Services (ICWS'06),: 205–212, IEEE Computer Society, . 2006.
- [5]. BenHalima, R., Jmaiel, M. and Drira, K." A qos-driven reconfiguration management system extending web services with self-healing properties" In 16th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises WETICE, :339–344. IEEE Computer Society. 2007.
- [6]. Repp, N.,Berbner, R.,Heckmann, O. and Steinmetz, R." A cross-layer approach to performance monitoring of web services" In Proceedings of the Workshop on Emerging Web Services Technology. CEUR-WS, .2006.
- [7]. Thio, N. and Karunasekera, S." Automatic measurement of a qos metric for web service recommendation " In ASWEC '05: Proceedings of the Australian conference on Software Engineering, : 202–211. IEEE Computer Society. 2005.
- [8]. Bianculli, D. and Ghezzi, C." Monitoring conversational web services" In IW-SOSWE '07: 2nd international workshop on Service oriented software engineering, :15–21, New York, NY, USA, ACM.2007.
- [9]. WebInject. Available at <http://www.webinject.org>
- [10]. Sandact. Available at <http://www.syhunt.com/>
- [11]. SoapKnox.Available in soapknox.com/index.html
- [12]. Java System Application Server and GlassFish Project.Available at <http://glassfish.dev.java.net/>
- [13]. Microsoft Baseline Security Analyzer.Available at <http://www.microsoft.com/technet/security/tools/mbsahome.mspx>
- [14]. Halima,R.,Guennoun,K.,Drira,K.and Jmaiel,M. "Non-intrusive QoS Monitoring and Analysis for Self-Healing Web Services" IEEE.2008.