An Enhancement of QoS in Web Services
Aziz Nasridinov, KyoungWook Kim, JeongYong Byun
Department of Computer and Multimedia, Dongguk University Gyeongju, South Korea
aziz_nasridinov@yahoo.com, kakaru36@dongguk.ac.kr, byunjy@dongguk.ac.kr

Abstract—Due to the wide acceptance of web services and its rapid spread, a number of mission-critical systems will be deployed as web services in the next years. The availability and efficiency of these systems has to be ensured in cases of failures and network disconnections. An example of web services for which availability and efficiency will be crucial issue is Auto Repair Service. It is critical because success in the Auto Repair business requires highly productive service personnel. So improving communication with Auto Part Shop is essential for Auto Repair Service to improve operational efficiency while providing better customer satisfaction. Thus, there is a three fold benefit of implementing multithreaded approach: increased speed of operation, high fault tolerance and overall better efficiency. In this paper, we propose multithreaded implementation of web services and compare our implemented prototype with previous prototypes in order to provide high-availability while improve communication between services.

Index Terms—QoS; multithreaded web services; Auto Repair Service; Auto Part Shop

I. INTRODUCTION

Web services technology is becoming a highly acceptance in all field. If this trend continues, an increasing number of mission-critical systems will start to be deployed in the next few years. However, to be effectively used in real world applications current web services technologies have to provide application developers with important quality of services (QoS) such as efficiency, and high availability. An example of web services for which efficiency and availability will be crucial issue is Auto Repair Service. It is critical because success in the Auto Repair business requires highly productive service personnel [1]. Needless to say, better customer satisfaction results in increasing profitability and loyalty. Customer satisfaction can be high if his/her request could be fulfilled instantly and properly instead of asking the customer to come later or wait little more. So improving communication with Auto Part Supplier is essential for Auto Repair Service to improve operational efficiency while providing better customer satisfaction.

Improving communication between Auto Repair Service and Auto Part Shop could be done if timeliness of communication will be emphasized [2]. Improving timeliness implies minimizing customer’s waiting time. According to general scenario in Auto Repair Service, many customers are untimely coming to repair their cars. Usually, ARS doesn’t have needed auto part. So after checking the car, service sends a sequence SOAP messages to Auto Part Supplier in order to request needed part. However, when working with web services there is a great risk that a service is nor responding due to several reasons, for instance, the hardware and software faults in web services can fail the client. There are also many network-related problems such as latency of response, loss of messages, corrupted messages, traffic congestion and etc which could waste significant amount of customer’s waiting time.

In order to provide high-availability while improving communication between services, in this paper, we propose multithreaded implementation of web services and compare our implemented prototype with previous prototypes [3,4] Due to more naturally reflection of concurrent activity in the real world, we believe that implementing of multithreaded approach can bring us a three fold benefit: increased speed of operation, high fault tolerance and overall better efficiency.

This paper is organized as following: in the 2nd chapter we will discuss related studies. Then we will focus on system design and show general scenario for proposed method. Implementation will be illustrated in the 5th chapter. We will compare our proposed prototype with previous prototypes. Paper will be concluded with conclusion.

II. RELATED STUDIES

Several specifications have been developed to deliver efficient web services in cases of failures and network related problems. [3] tries to improve quality of communication by emphasizing the role of Functional-Level Mediator in web services architecture, illustrating how the Functional-Level Mediator can enable web services usages in matching customer’s and web service’s goal. The only disadvantage of this approach is that it only deals with reducing heterogeneity existing in representing data between Auto Repair Service and Auto Part Shop. However, only with improving quality of communication it’s hard to deliver fast service to customer to whom time is critical.

In [4], along with improving quality of communication and resolving possible mismatching between Auto Repair Service and Auto Part Shop arise due to heterogeneity existing in representing data, we proposed a method on business process integration with development of time efficient web service in order to improve timeliness of communication between Repair Service and Auto Part Shop. The main point of this research was to establish communication through alternative web services. In other words, Auto Repair Service system sends request to Auto Part Shops system located in same area as Auto Repair Services in some predefined time. If there is no response...
from nearest shops, system had to redirect messages to next nearest area so that Auto Repair Service could save significant amount of customer’s waiting time. However when switching to alternative web service system could fail or messages could be lost or switching time could be increased due to network-related problems.

K Birman et al. [5] proposes extensions to the Web Services architecture to support mission-critical applications. Examples include standard services that track the health of system components, mechanisms for integrating self-monitoring, self-diagnosis of faults and self-repair into applications, automated configuration tools, scalable event reporting mechanisms, and tools for large-scale data mining. Although this research presents a very good to solve problems mentioned in 1th chapter, but it doesn’t consider time which is very important in mission-critical applications. Because self-monitoring, self-diagnosis of faults and self-repair of the system could waste much of user’s time.

III. SYSTEM DESIGN

In this the interaction scenario will be described along with proposed system design to address this scenario by illustrating case in Auto Repair Service. A high degree of availability is necessary while exchanging messages in a business-to-business scenario where important transactions are conducted over the web [6]. According to the this basic scenario between customer, Auto Repair Service and Auto Part Shop, after checking damaged part of vehicle Auto Repair Service system should request it from multiple Auto Part Shop system. Web service based system design is shown in Figure 1.

![Figure 1. Web services design](image1)

Working cycle of the following system is as following: customer requests Auto Repair Service to fix problem of his/her vehicle. After identifying what is the problem and which part is needed to solve the problem of vehicle, Auto Repair Service system sends sequential SOAP messages to Auto Repair Shop system. After processing request Auto Part Shop system will send detailed result of request to Auto Repair Service. If Auto Repair Service system wants to communicate with another Auto Repair should wait until accepting response from firstly requested Auto Part Shop.

When working with web services there is a great risk that a service is nor responding due to several reasons, for instance, the hardware and software faults in web services can failure of the client. There are also many network-related problems such as latency of response, loss of messages, corrupted messages, traffic congestion and etc. In these cases, multithreaded implementation of web services comes in action.

IV. CONSTRUCTION AND TEST

Our main goal in this study was to determine the general level of performance available from commercial usage of multithreaded implementation of web services. We designed a set of test cases of network-related problems and messages lost that reflected typical commercial usage in Auto Repair Service. All of these test scenarios were designed to resemble real life commercial transactions while avoiding any unnecessary server-side processing overhead which may affect to measure performance.

Multithreaded web service test driver written in Netbeans IDE using Java is shown in Figure 3 with the case in local Auto Repair Service. According to our example customer comes to change Wheel of “Opirus” car manufactured by KIA Motors. After filling needed part’s data, Auto Repair Service system sends parallel messages to Auto Part Shop system.

![Figure 2. Overall system design](image2)
In this case, since network condition is good Auto Repair Service system can accept response immediately in which details such as price and response time from five various Auto Part Shops are included. Service then can show it to the customer so that he/she can make choice from various options.

Figure 4 shows case where customer requests Converter of “Sportage” car manufactured by KIA Motors. From this figure, we can see that responses from “Gyeongbuk Com” and “Dongwoo Com” were arrived later than others due to network-related problems. However, in this case, service doesn’t have to wait response from delayed services because it accepts responses from other parallel requested services as it shown.

Figure 5 shows case where customer’s request is Muffler of “Lotze” by KIA Motors. In this figure, we can see that there are no responses from two “Simin Com” and “Daihyun Com” shops due to messages lost in both server sides. Nevertheless, customer’s request will be processed since Auto Repair Service system accepts responses from other parallel requested services.

V. PERFORMANCE ANALYSIS

Assuming that our implemented method is good, we believe that our method achieves those main goals predefined in Chapter 1. In order to show that we compared our performance with other equivalent prototypes introduced in previous conferences. This comparison gives us a standard for comparing performance, allowing us to see how well multithreaded web services are when it is applied to Auto Repair Service.

Identical Samsung Smart Server ZSS122 servers were used as both client and server systems. The hardware and software configuration of these systems are:

- Five Intel Pentium IV processors
- 512 Mbytes of memory
- Microsoft XP with Service Pack 3
- Microsoft .NET Framework 1.1

All the experiments were run over a 100 Mbps switched Ethernet LAN using a 100 Mbps Cisco switch. Auto Repair Shop system was developed in Microsoft Visual Studio 2005, C# and JDeveloper 10g, Java. Web service broker was developed using Netbeans IDE 6.1 and Java.

The performance tests were conducted using multithreaded web services where client Auto Repair Service repeatedly sends request to the Auto Part Shop web services. Figure 6 shows the results obtained in three cases: A is when there is no any problem in both system, network-related problems is considered in case B, and C is drawn from message lost.

<table>
<thead>
<tr>
<th>Case</th>
<th>Current Method</th>
<th>APIS 2009</th>
<th>UCWIT 2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>200</td>
<td>270</td>
<td>320</td>
</tr>
<tr>
<td>B</td>
<td>200</td>
<td>1150</td>
<td>1900</td>
</tr>
<tr>
<td>C</td>
<td>200</td>
<td>1800</td>
<td>4200</td>
</tr>
</tbody>
</table>
Figure 7 shows the visual view of performance results using table 1. From this figure, we can see that in three cases listed above our proposed method has a minimum waiting time comparing other methods. Current results show that the multithreaded implementation web services are able to deliver reasonably good performance in order to improve QoS.

![Response Time Comparison](image.png)

**Figure 7. Performance Analysis**

VI. CONCLUSION

In this research, we have successfully undertaken usage of multithreaded web services in order to reflect a range of typical commercial activities. We believe that we brought a three-fold benefit of implementing multithreaded approach: increased speed of operation, high fault tolerance and overall better efficiency. We have compared our implemented method with previous methods in order to demonstrate the steps of improvement and successfulness of our current research. Our results has shown that implementation of this method was able to deliver reasonably good performance in order to improve QoS.

The improvement steps have been motivating us throughout our researches and as a logical continuation of improvement is considered to be a delivering overall efficiency of system by presenting Orchestrated web services method.

ACKNOWLEDGMENT

This research was financially supported by Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

REFERENCES


