Reliable Web Service Management with Backup & Parallel Service

Deepak Moses P[1], Deivanarayanan K[2], Mr.Srinivasan[3]

[1,2,3]Faculty of Computing,
Sathyabama University, Chennai, Tamilnadu, India
sonicdeepak@gmail.com, 2ramarajkesavan@gmail.com, 3 srinijyothish@gmail.com

Abstract

In the EXISTING SYSTEM, we building a reliable service oriented systems is more important when compared with the traditional stand-alone system in the unpredictable internet service and it also a challenging task to build reliable web service. In the PROPOSED SYSTEM, we find the fault tolerance by using heuristic algorithm which is proposed.

Two kinds of strategies available are active and passive strategies. And we also formulate the user requirement as local and global constraints. In the MODIFICATION PROCESS, different services are deployed. We deploy two bus reservation and two train reservation service along with hotel reservation service. User can choose any one of the bus reservation and specify their destination location. If corresponding destination is not available then automatic backup service to another bus reservation system is carried. If same the service is not available then parallel service of train reservation is initiated. Automatic hotel reservation is also initiated based on the mode and type of travel of the user.

Keywords: Reservation, Fault tolerance, heuristic algorithm, Backup service, Parallel Service

I. INTRODUCTION

Web services are applications that can be described and invoked over the Internet. Complex distributed systems can be dynamically composed. As service oriented architecture (SOA) is becoming a large part of IT. Building reliable service-oriented systems is important. However, compared with standalone systems, building reliable service-oriented systems is much more challenging, because (1) Web services are usually deployed across various aspects in the Internet (2) remote web services are developed and hosted by third party providers without internal design details;

To Cite This Article: Deepak Moses P, Deivanarayanan K and Mr.Srinivasan., **Reliable Web Service Management with Backup & Parallel Service** . Journal for Advanced Research in Applied Sciences ;Pages: 8-11

9. Deepak Moses P, Deivanarayanan K, Mr.Srinivasan., **Reliable Web Service Management with Backup & Parallel Service**. **Journal for Advanced Research in Applied Sciences**

(3) Performance of Web services may change dynamically and (4) remote Web services might become unavailable without any kind of notification.

In software reliability engineering, there are different kinds of approaches to increase system reliability, which are fault prevention, fault removal, fault tolerance and fault forecasting.

Reliable Web Service Management with Backup & Parallel Service

Since source-codes and internal designs of Web services are unavailable to service users, it is difficult to build fault-free service-oriented systems. Software fault tolerance makes the system more robust. One approach of software fault tolerance, also known as design diversity, which is to deploy functionally equivalent yet independently designed components to tolerate faults. Due to the cost of redundant components, design diversity is usually only used for critical systems. In the area of service computing, however, it is possible to construct a fault-tolerant service-oriented system..

II. EVIEW OF RELATED WORK

M. Alrifai and T. Rises, (1) discussed about "Combining global optimization with local selection for efficient Quos-aware service composition," discuss about the run-time binding of web services has been recently put forward in order to support rapid and dynamic web service compositions. With the growing number of alternative web services that provide the same functionality but differ in quality parameters, the service composition becomes a decision problem on which component services should be selected such that user's end-to-end Quos requirements and preferences in Proc. 18th Int. Conf. World Wide Web (WWW'09), 2009, pp. 881–890.

D. Ardagna and B. Penrice, explained about (2) "Adaptive service composition in flexible processes," introduce a new modeling approach to the Web service selection problem that is particularly effective for large processes and when Quos constraints are severe, IEEE Trans. Soft. Eng., vol. 33, no. 6, pp. 369–384, Jun. 2007. A. Avizienis, discussed about (3) "The methodology of N-version programming, "is a method or process in software engineering where multiple functionally equivalent programs are independently generated from the same initial specifications in Software Fault Tolerance, M. R. Lye, and Ed. New York, NY: Wiley, 1995, pp. 23–46.

B. Benatallah, M.Dumas,Q.Z. Sheng, and A. H. H.Ngu, explained about (4) "Declarative composition and peer-to-peer provisioning of dynamic web services, "we describe the design and implementation of a system through which existing Web services can be declaratively composed, and the resulting composite services can be executed following a peer-to-peer paradigm, within a dynamic environment. in Proc. 18th Int. Conf. Data Eng. (ICDE'02), 2002, pp. 297–308.

C.-L. Hwang and K. Yoon, Eds., explained (5) "Multiple criteria decision making, "methods have been proposed by researchers in diversified disciplines; half of them are classical ones, but the other half have appeared recently Lecture Notes in Economics and Mathematical Systems. Berlin, Germany: Springer-Verlag, 1981.

P. P.-W. Chan, M. R. Lye, and M. Malek, explained about (6) "Making services fault tolerant," we identify parameters impacting the Web services dependability, describe the methods of dependability enhancement by redundancy in space and redundancy in time and perform a series of experiments to evaluate the availability of Web services in Proc. 3rd Int. Serv. Availability Sump., 2006, pp. 43–61. B.Chun,D.Culler, T. Roscoe, A. Bevier, L. Peterson,M.Wawrzoniak, and M. Bowman, discussed about (7) "Planet Lab: An overlay tested for broad coverage services," describes our initial implementation of Planet Lab, including the mechanisms used to implement virtualization, and the collection of core services used to manage Planet Lab. ACM SIGCOMM. Rev., vol. 33, no. 3, pp. 3–12, Jul. 2003. T. Carmen, C. Leiserson, and R. Rivets, discussed about(8) Introduction to Algorithms. Cambridge, MA: MIT Press, 1990.

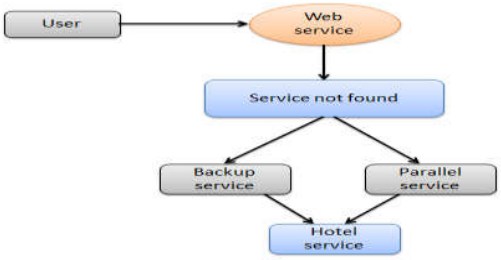

Fig: System Architecture

10. Deepak Moses P, Deivanarayanan K, Mr.Srinivasan., **Reliable Web Service Management with Backup & Parallel Service**. **Journal for Advanced Research in Applied Sciences**

III. BACKUP WEB SERVICE

In this module, different services are deployed. We deploy either two bus reservation or two train reservation service along with hotel reservation service. User can choose any one of the bus or train reservation and specify their destination. If the corresponding service is not available then automatic backup service to another bus or train reservation system is carried.

```
if (travelmode == "Train")
        {  Session["travel"] = "Train";
            Session["busticket"] = "nobus";
            if (journeymode == "One Way")
            {
                sqlquery = "select * from Oneway where Origin='" +
DropDownList1.SelectedItem.ToString() + "' and Destination='" +
DropDownList2.SelectedItem.ToString() + "' and Date='" + TextBox5.Text + "'";
            }
            else if (journeymode == "Round Trip")
            {
                sqlquery = "select * from Roundtrip where Origin='" +
DropDownList1.SelectedItem.ToString() + "' and Destination='" +
DropDownList2.SelectedItem.ToString() + "' and Odate='" + TextBox5.Text + "' and Ddate='"
+ TextBox6.Text + "'";
            }
            railcon.Open();
            SqlCommand rcmd = new SqlCommand(sqlquery, railcon);
            SqlDataReader rdr = rcmd.ExecuteReader();
            if (rdr.Read())
            {
                Session["trainnum"] = rdr["Trainno"].ToString();
                if (DropDownList4.SelectedItem.ToString() == rdr["Class"].ToString())
                {
                    HyperLink1.Visible = true;
                 Session["Rinclass"] = DropDownList4.SelectedItem.ToString();
                    string message = "Train Tickets are Available on Selected Class.
Goto the Below Link for Confirmation";
                    System.Text.StringBuilder sb = new System.Text.StringBuilder();
                    sb.Append("<script type = 'text/javascript'>");
                    sb.Append("window.onload=function(){");
                    sb.Append("alert('");
                    sb.Append(message);
                    sb.Append("')};");
                    sb.Append("</script>");
                    ClientScript.RegisterClientScriptBlock(this.GetType(), "alert",
sb.ToString());                }
```

Fig: Process redirection

IV. PARALLEL WEB SERVICE

In this module, if the user opted service is not available then parallel service of train or bus reservation is initiated. User will select the ticket to reach the destination based on this model representation, to generate the lists for personalized travel package recommendation.

V. AUTOMATIC WEB SERVICE COMPOSITION

In this model user can choose their mode of travel either by bus or by train based on the user's selection of travel automatically. Hotel reservation will be recommended by the server automatically based on user's class of service deployed in ticket reservation.

VI. CONCLUSION AND FUTURE WORK

In our work we have proposed fault tolerance to efficiently select the best possible outcome from a variety of possible outcomes under a given scenario. In the PROPOSED SYSTEM, we find the fault tolerance by using heuristic algorithm which is proposed. Two kinds of strategies available are active and passive strategies. And we also formulate the user requirement as local and global constraints.

Heuristic algorithms find solutions among all possible ones, but they do not guarantee that the best one will be found among the possible solutions, therefore they may be considered as approximated algorithms but they are not accurate.

These algorithms usually find the best solution and they find it instantly. These algorithms are not always accurate, but the algorithm is still called heuristic until the best solution is proved.

References

[1] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient QoS-aware service composition," in Proc. 18th Int. Conf. World Wide Web (WWW'09), 2009, pp. 881–890.

[2] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes," IEEE Trans. Softw. Eng., vol. 33, no. 6, pp. 369–384, Jun. 2007.

[3] A. Avizienis, "The methodology of N-version programming," in Software Fault Tolerance, M. R. Lyu, Ed. New York, NY: Wiley, 1995, pp. 23–46.

[4] B. Benatallah,M.Dumas,Q.Z. Sheng, andA. H. H.Ngu, "Declarative composition and peer-to-peer provisioning of dynamic web services," in Proc. 18th Int. Conf. Data Eng. (ICDE'02), 2002, pp. 297–308.

[5] C.-L. Hwang and K. Yoon, Eds., "Multiple criteria decision making," Lecture Notes in Economics and Mathematical Systems. Berlin, Germany: Springer-Verlag, 1981.

[6] P. P.-W. Chan, M. R. Lyu, and M. Malek, "Making services fault tolerant," in Proc. 3rd Int. Serv., 2006, pp. 43–61.

[7] B.Chun,D.Culler, T. Roscoe,A. Bavier, L. Peterson,M.Wawrzoniak, and M. Bowman, "PlanetLab: An overlay tested for broad coverage services," ACM SIGCOMM Computer.. Rev., vol. 33, no. 3, pp. 3–12, Jul. 2003.

[8] T. Cormen, C. Leiserson, and R. Rivest, Introduction to Algorithms. Cambridge, MA: MIT Press, 1990.