ISSN:0975-9646

A Self-Healing Approach for Service Unavailability in Dynamic Web Service Composition

S.Poonguzhali[1], L.JerlinRubini[2], S.Divya[3]

Saradha Gangadharan College,
Puducherry, India

Abstract- **Web service composition is defined as the orchestration of multiple component services into single composite service. A dynamic web service composition leads to several faults like poor response, service un-availability, incorrect order etc. Self healing is one of the properties in autonomic computing which makes the system to heal itself without any human intervention. We proposed a self healing architecture for service unavailability in dynamic web service composition.**

Keywords- dynamic web service composition, self-healing, autonomic computing, service unavailability.

I Introduction

Web service composition is defined as the orchestration of atomic web services into a composite service. The web service composition facilitates creation of new web service from a set of available service. Thereby web service composition facilitates the reusability. The dynamic composition is very challenging as it is done at run-time based on the user's request [10]. Many researches have been done on dynamic web service composition. Many service providers provide the web services which performs same function as other web services provided by the some other web service providers. Web service composition is done by choosing the service which satisfies user request to the maximum level. The component web service's running environment and the composer running environment may differ. By the composite must balance with changing environment. The environments may be bandwidth, and software support. If the web service cannot balance itself then it leads to several faults such as ***incorrect order, misunderstood behavior***, QoS service failure such as ***poor response*** and ***service unavailability***, etc [1]. Incorrect order occurs due to message flow through SOAP. When the transfer of packets is in different order in receiver side than at the sender side it leads to incorrect order. Misunderstood behavior occurs when the requester receives the different service than he expects. This is because sometimes the service description may not be correct. When failure occurs the composite service will not be provided until it has been healed. [1]

This paper focuses on unavailability of service in dynamic web service composition. The service will be unavailable when the network route is in congestion and the network has been broken down. When any one of the component service is unavailable then we cannot provide a composite service to the user. In order to avoid such circumstances we provide a self healing approach which provides an alternate path to the service to reach the composer when the network path has been broken down.

Self healing is one of the properties of autonomic computing. Self-healing is a mechanism which makes the system to detect and correct the faults to itself. In web service composition the self healing is needed to heal the unavailability of service without any human intervention. The proposed self-healing approach detects which service is not invoked and finds the root cause for the failure and corrects those faults. If any one of the services is not available it is known that either the server of the particular service has been inactive or the data cannot be passed through the path by which it has to be. If the network path is broken then the self healing mechanism will go for another path which reachable from the router where the data is to the composer. Self healer chooses one of many paths which is more closest to composer (i.e. minimum number of hops).

II related Works

In this section we will review the works done on self-healing web service composition.

In [2], the self-healing policy is proposed as an enhancement of WS-BPEL engine.. This policy monitors for unexpected failures in web service process and diagnoses the root cause for the failures and finds the solutions to recover from that failure. The diagnosis part this policy uses a database which stores the details of failures which has been occurred already to detect the failures. If failure is not encountered in database the manual intervention is needed in this policy.

In [3], they proposed a web service composition cycle called MAPE cycle. The MAPE cycle consists of three modules: Plan Generation Module in which the user request is parsed and converted into the form understandable by the composer. Plan execution module executes the plan generated by the plan generation module. If it requires it re plans the current plan. The Failure analysis module finds the root cause of the failure. This cycle is used to evaluate the existing self healing techniques in web service composition.

In [4] authors have discussed some key technologies to achieve web service composition. They used aspect oriented programming to monitor the failures by keep tracking the WS-BPEL process. They used some statistical and machine learning techniques for diagnosis process. In recovery stage instead of giving exact solution to the failure they give desirable solutions.

In [5] authors have proposed aspect based self-healing healing approach onto a three level architecture namely invitation, instantiation, and deployment they developed some exceptions for different types of failures.

In [6] authors have proposed approaches for monitoring and analysis of QoS failures. The QoS parameters they considered are response time, throughput, and availability. They monitored the SOAP messages between the web services. They developed an algorithm which discriminates network and processing deficiencies.

In [7] authors have proposed QoS driven approach by performance prediction. They developed availability and reliability algorithm for self healing web service composition.

In [8] authors have used two special purpose languages such as WSCOL and WSREL for monitoring and recovery strategies. WSCOL is used for specifying constraints on the execution of BPEL processes.WSREL is used to state recovery strategies which should occur when the constraints are violated. They used BPEL activities for monitoring. The monitoring and recovery strategies are based on JBOSS rule engine. The recovery strategies are based on JBOSS production rule which is based on ECA (Event-Condition-Action).

In [9] authors have proposed recovery framework for composite web services. This framework focuses on observable faults and interaction faults. They used interaction pattern and control pattern for monitoring. They make use of symptom database for determining the symptoms of error. Fault taxonomy is used to find the root cause of error. Formal concept analysis is used to find where to adjust the composition. Table 1 shows self-healing approaches in each work.

In [10], a self-healing approach for dynamic web service composition is proposed. It detects performance based failures and heals that failure. It takes the performance based parameters as response time. It detects the service which has greater response time and replaces another service which has the same input, output, pre and post conditions.

III PROPOSED WORK

The dynamic web service composition is done by choosing and orchestrating the component web service during run time when the user request the service. When the user request the service the service selector in the composite service selects the service from UDDI and BPEL template is generated dynamically and it is executed by the BPEL engine. The BPEL engine executes the service by parsing the tags in the BPEL template. This BPEL engine is monitored by the BPEL monitor in fig.1 which checks whether all the web services in the composite web service is executed when the BPEL engine executes the invoke command. When the <invoke> is executed by the BPEL engine the BPEL monitor checks whether the service is invoked within the threshold time. The threshold time is the time limit for the web service it takes to reach the composer. When the web service does not reach the within the threshold time, the BPEL monitor gives a signal to the diagnoser. The diagnose checks whether there is any break in the network path. If there is any break in the network it informs the path substituter. The path substituter checks whether there is any alternate path from the router where the path broke. If there is a path then it replaces the path in BPEL engine for the particular service.

A.WORKING COMPONENTS

The web service composition is performed by executing the BPEL processes. The BPEL process called <invoke> is used to execute the particular service in the composite web service. Each web service has its threshold to reach the composer. If the time to reach the composer side, then it is said to be the service is unavailable. To overcome this service unavailability problem, the self healing architecture is proposed. The components involved in the proposed architecture are BPEL Monitor, Diagnoser, path substituter.

1) BPEL MONITOR:

The BPEL monitor monitors the BPEL engine to check whether any service is not invoked. Generally the BPEL engine executes compiled BPEL file to provide composite service. BPEL engine calls the component service by executing the BPEL process. When the BPEL engine executes the <invoke> process, the BPEL monitor verifies whether the service has reached the composer. It sets the time limit to reach the composer. If the component service does not reach the composer when the BPEL engine executes the invoke process within a time limit, then it sends the report to diagnoser.

2) DIAGNOSER:

The diagnoser finds the root cause for the failure. The diagnoser in the proposed architecture finds where the problem arise whether in server itself or in the network path. If it is in network path it prepares a report of which service’s route path has been broken down from which router the packet has to be routed. Then the report is sent to the path substituter.

3) PATH SUBSTITUTOR:

The path substituter takes the report sent by the diagnoser as an input and starts searching for new path to reach the composer. From the router where the packet is currently present, the path substituter finds the alternate path route by sending hello packet to router nearby the router where the packet is present. If it finds the route reach the composer then stores that route and again it give that alternate route to that composite web service. It also make use of the routing path registry to find the alternate path.

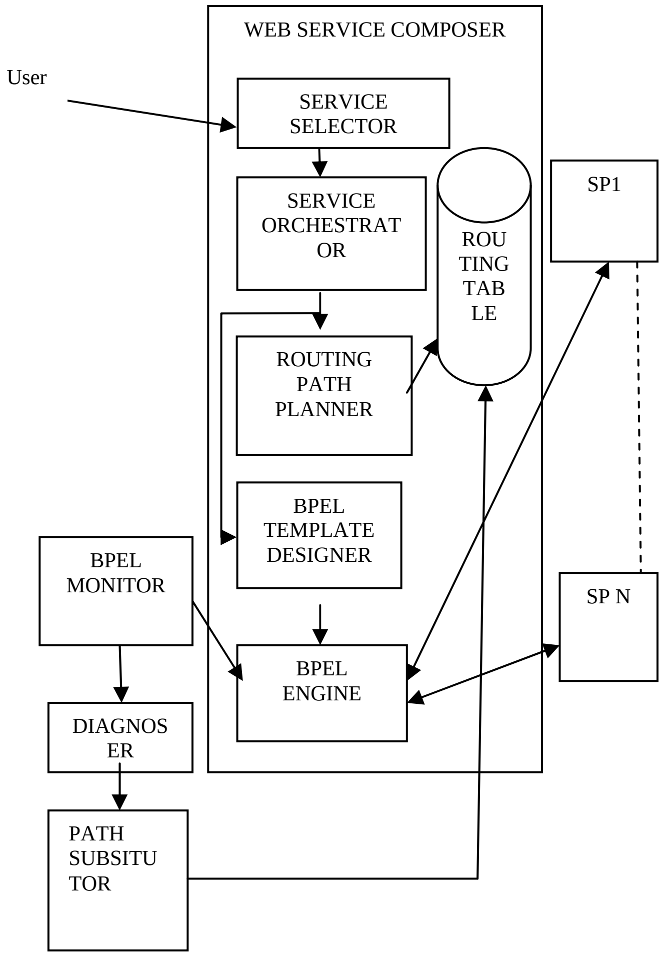

Fig 1. Self-Healing Architecture for Dynamic Web Service Composition

IV CONCLUSION

Our approach to self healing in dynamic web service composition provides an alternate routing path to the service which cannot reach the composer by monitoring the all the service invocation process of BPEL engine by predicting the time it takes to reach the composer. The alternate routing path is identified dynamically by sending hello packet to finds the nearby router to the composer.

REFERENCES

[1] K.S. May Chan, Judith Bishop, Johan Steyn, Luciano Baresi and Sam Guinea, A Fault Taxonomy for Web Service Composition, ICSOC2007Workshops,LNCS 07,2009,pp.363-375

[2] Sattanathan Subramanian, Philippe Thirananjangud C. Narendra, On the Enhancement of BPEL Engines for Self- Healing Composite Services International Symposium on Applications and the Internet,Saint,2008,PP 33-39

[3] K.S. May Chan ,Judith Bishop, The Design of a Self-healing Composition Cycle for Web Services, *SEAMS'09*, Vancouver, Canada,PP20-29

[4] Guoquan Wu, Jun Wei, Tao Huang, Towards self healing webservice composition,*Internetware'09*,Beijing, China,2009.

[5] Ghita Kouadri Mostefaoui, Zakaria Maamar, On Modeling and Developing Self-Healing Web Services Using Aspects,1-4244-0614-5/07,2007.

[6] Riadh Ben Halima, Karim Guennoun, Khalil Dria,Mohamed Jmaiel, Non-intrusive QoS Monitoring and Analysis for Self-Healing Web Services**,** 978-1-4244-2624-9/08 pp:549- 554,IEEE,2008

[7] Yu Dai, Lei Yang, Bin Zhang, QoS-Driven Self-Healing Web Service Composition Based on Performance Prediction, Journal of computer science and technology,pp 250- 261,2009

[8] Luciano Baresi, Sam Guinea, Liliana PasQuale,"Self-Healing BPEL Processes with Dynamo and the JBOSS rule engine",ESSPE'07,ACM,pp:11-20,2007

[9] K.S.May Chan,"Towards aFramework for web service compositions recovery",ICSE-WUP'09,The warm-Up Workshop for ACM/IEEE ICSE 2010,pp:17-20,2009.

[10] S.Poonguzhali,R.Sunitha,Dr.G.Aghila, "Self-Healinh in Dynamic Web Service composition",IJCSE,pp:2054-2060, Vol. 3 No. 5 May 2011