A NOVEL SERVICE ROUTING METHOD IN A SERVICE-ORIENTED NETWORK

Xiaoxiang Luo and Junde Song
Institute of Computer Science, BUPT, Beijing, China

Keywords: Service Routing, Service-oriented Network, Service Naming.

Abstract: In this paper, a service routing method is proposed to support Service Orchestration. The service-oriented network is composed of service node and assisted node. A service node is designed to employ the service execution and service routing functions. In our method, it is an efficient service routing as there is no centric point to control the orchestration path, thus there're no single point bottleneck problem. It is also a pervasive method for a service-oriented network.

1 INTRODUCTION

Today, Web 2.0 and service-oriented architectures (SOAs) are among the top issues. Both are poised for exponential growth over the next few years, due to their flexibility, cost effectiveness, and ease of integration.

In a Service-oriented network, a user's request is decomposed into several single services and an service orchestration serves as a composite to fulfil users requirements. The concept of service routing is to connect and unite service service nodes to form higher-level target services. (F. Bennani, 2006). This form of routing would serve as a coarse-grained coordination mechanism, because it would govern both the inclusion and order of services following a service orchestration path.

In a Service-oriented network, routing is a fundamental execution mechanism.

2 RELATED WORK

Most approaches in service composition literature are based on Web Service, such as the Semantic Web. (Torsten Klie, 2007), (Tim Berners-Lee, 2001).

The service implementation is based on the interfaces described in standardized machine-executable format (such as WSDL). With sharable Web Services, tremendous reuse of applications based on functionality may be achieved.

Automatic, dynamic or QoS-aware Web service composition proposals are discussed. Those proposals all act in the mode of the Client-Server which means that tremendous protocol information should be exchanged to achieve a user demanded generic service. (L.A.G. Dacosta, 2004), (Liangzhao Zeng, 2004). The core managing service node inevitably becomes the performance bottleneck. Meanwhile, the method for composition planning is computation intensive, and QoS limitations can be identified. (Snehit Prabhu, 2007)

For service convergence, SOA (Service Oriented Architecture) based service composition is emerging. SOA is defined as a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. (M. MacKenzie, 2006). It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations". The services in SOA architecture are sharable and reusable, however its architecture is not flexible by following the Client-Service mode, thus some service type cannot be included (e.g. the services in user device).

It is clear up to now that a service routing method is necessary in all service composition system to support efficient service generation.

3 SERVICE ROUTING

3.1 Service-oriented Network

A service oriented network is constructed of two

Luo X. and Song J..
A NOVEL SERVICE ROUTING METHOD IN A SERVICE-ORIENTED NETWORK.
DOI: 10.5220/0003587605600563
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (SSE-2011), pages 560-563
ISBN: 978-989-8425-53-9

kinds of node: service node and assisting node

(1) Service Node

- Request Parser

The goal of the Request parser is to select the best set of Single Service for user request. The identification of the minimal Single Service set requires only an inspection of the service functionality that fulfills the task completely or in parts. The shortest sequence(s) of Services are selected. Then an Orchestration Plan is derived by organizing the selected set of Service in a ordered sequence.

- Service Execution Environment

It is an execution environment for service and service integration where services are integrated for various business applications.

Service with an input are executed in this execution environment, and returns with an output.

- Service routing function

Service routing function is a matching process based on local routing tables. It is in charge of looking up in routing table for next service, and sends the user request out again.

(2) Assisting Node

- Registry

Registry stores non-functional characteristics of single Service, including service provider, service location, service quality and abstract interface. The information is described in a predefined data structure.

Registry updates the information of new available services. As service registry increases, the Request parser has more choice on services selected for each task.

Whenever a service node cannot find a next service node, the orchestration plan is forwarded to a service registry. Then the next service node IP is returned to the original service node and the plan is forwarded to the next service node.

3.2 Service Routing Method

The network literature typically defines routing as the processes when a router receives a message, looks its destination up in a routing table, and sends the message out again.

In service routing, local routing tables are also maintained. But when a service node receives a user request, it first does its own job, executes the single service it is in charge of and then looks up in a table for next service, and sends the user request out again.

Whenever a user request is generated, it is first sent to the service Request Parser, it analyze and then decompose the request into single services and an Orchestration Plan is generated. Orchestration Plan includes all the necessary single services and the execution order of services.

Then the Request Parser forward Orchestration Plan to the next node and the service routing method controls the path and invokes each service in the list to perform orchestration plan.

The basic operation of a service node is very similar to an IP node. A request arrives on a face, a look-up is done on its name and metadata, and then an action is performed based on the result of that lookup.

3.2.1 Orchestration Plan

The orchestration plan is routable service logic. It involves a set of all the available services with the service input and control information. It is written in a script which wraps the selected services into a service logic describing data flow and control among the ASs.

3.2.2 Service Node Design

Figure 1 is a core schematic of the service node. It has three main data structures: the Service FIB (Forwarding Information Base), Service Store (service execution environment).

The Service Store is the same as the service list that is to be executed in this service node. If the service in the orchestration plan is found in the service store, a service execution is initiated and the output of this service is added to the plan. And then the plan is to be forwarded downstream.

The Service FIB is used to forward orchestration plan toward potential node(s) of matching service.

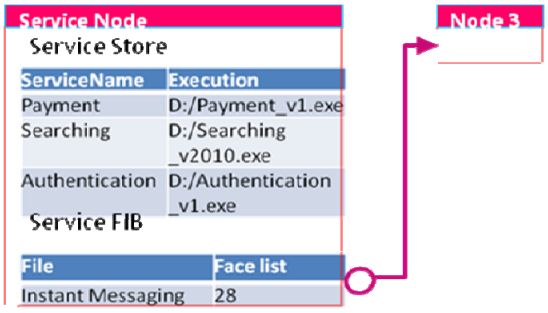

Figure 1: Schematic of the service node.

When an orchestration plan arrives on some face, a lookup is done on its Service Name. The index structure used for lookup is ordered so that a Service Store match will be preferred over a FIB match.

Thus if there is already a service in the Service Store that matches the service listed in the plan, it will initiate the execution of the single service. and after the execution, the output of this service is added to the plan. If there is more than a service in the Service Store that matches the service listed in the plan, it will initiate the execution of the services in the order described in the orchestration plan. And after the execution, the output of this service or these services will be added to the plan.

If there is no service in the Service Store that matches the service listed in the plan, the plan is to be forwarded downstream.

Then a look up is done for a FIB match. If there is a service in the FIB that matches the "first to be done" service listed in the plan, the plan is sent out one of the faces to the next service node.

If there is no match for the services in the FIB, plan is forwarded to a service registry. Then the next service node IP is returned to the original service node and the plan is forwarded to the next service node. The original service node added a record of this <servicename, IP> pair to the FIB.

All service nodes can provide some service executing and routing to next service node.

4 IMPLEMENTATION

To show the service routing process, we present an example of E-Commerce service orchestration in an e-bookshop scenario. We explain how the service nodes can support a process for the user's e-purchase in the on-line bookshop. We depict the orchestration process. In Figure 2, the whole process is accomplished without user intervention.

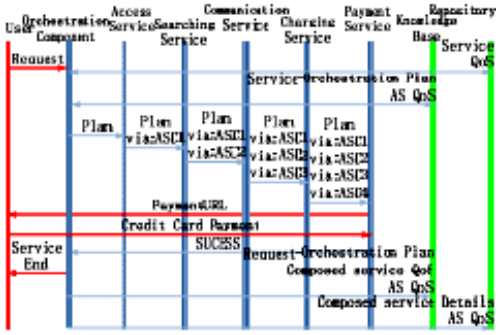

Figure 2: Example of service orchestration for e-commerce service.

The orchestration plan is passed to the involved service nodes one by one. The most suitable service is selected according to the predefined optimal path in the orchestration plan. At the end of each execution of SERVICE, orchestration plan will be sent to the consequent service node with an output added to the orchestration plan. Note that this procedure follows the optimal path and the service nodes communicate with each other based on the routing method without any centralized coordination or control.

Figure 3 shows the Orchestration Plan described in XMLschema. It shows how the plan controls the path and invokes services in service node perform orchestration. Single Service is identified with a serviceName and each service node is associated with a providerID. The two uniquely identifies the business en-tity offering that service. Each AS has four attributes (serviceName, state, control path, data path) defined in names of (serviceName, state, Ctrl, dataIn, DataOut). Once the service is successfully accomplished, its attributes except serviceName are changed. The execution order of AS is defined by thetag "step". Each AS may have multiple service nodes provided with different negotiated QoS by different provider. The service node candidates are listed in each Service field.

```
<?xml version="1.0" encoding="UTF-8" ?>
<op>
  <e-commerce customerID="cus1"
  customerAddr="215.35.67.33"
ecommerceID="ec1"/>
  <step stepnum="1" Ctrl="orderly">
    <service serviceName="S1" state="wait"
  ProviderID="sohu" DataIN="Din1"
  DataOUT="Dout1" S1.addr="null" />
    <service serviceName="S2" state="wait"
  ProviderID="sohu" DataIN="Din2"
  DataOUT="Dout2" S2.addr="null" />
</step>
  <step stepnum="2" Ctrl="parallel"
returnpoint="S2.addr">
    <service serviceName="S3" state="wait"
  ProviderID="sohu" Ctrl="orderly"
    DataIN="Din3" DataOUT="Dout3"
  S3.addr="null" />
    <service serviceName="S4" state="wait"
  ProviderID="sohu" Ctrl="OL1" DataIN="
  Din4" DataOUT="Dout4" S4.addr="null" />
</step>
</op>
```

Figure 3: Orchestration Plan in XML Schema.

The orchestration plan is passed through

service node one by one. Each service node parses the orchestration plan, execute the service and forward this plan to the next service node. The main method of dealing with the orchestration plan is included.

Once the orchestration plan is created, the Service node passes this orchestration plan to the service node that performs the first step. The address of the service nodes is got from the FIB. Then the service node checks if a service instance is able to be created. The orchestration plan is passed to the service node with the same serviceName one by one until an available service node is found. The available service node executes its operations according to the control input (”Ctrl”) got from the orchestration plan. The data input is a combination of initial data input of this serviceName and the data output of its previous Service. If successfully executed, the state of this AS is marked ”success”. And its ”ProviderID” field is filled and it is used for charging afterwards. Its ”DataOut” field is filled and it is used for constructing the input of next service. Then the orchestration plan is passed to the next service node that performs the second step in the orchestration plan.

The above process is repeated until all the steps are successfully passed. The last service node passed the orchestration plan back to the customer.

The orchestration plan is rounded off until accomplished.

5 CONCLUSIONS

In this paper, a service routing method is proposed to support Service Orchestration. The service-oriented network is composed of service node and assisted node. A service node is designed to employ the service execution and service routing functions. In our method, it is an efficient service routing as there is no centric point to control the orchestration path, thus there’re no single point bottleneck problem. It is also a pervasive method for a service-oriented network.

ACKNOWLEDGEMENTS

This work is financially supported by Program for New Century Excellent Talents in University, No.NCET-08-0738; and the innovation technology star program of Beijing under grant No.2007A045.

REFERENCES

F. Bennani, Z. Benahmed Daho, N. Simoni, Chunyang. Yin. An Informational Framework for Autonomic Networking. GRES 2006, Bordeaux, France.

Torsten Klie, Felix Gebhard.Towards Automatic Composition of Network Management Web Services. *10th IFIP/IEEE International Symposium on Integrated Nezwork Management*, Munich, Germany, 21-25 May 2007.

Tim Berners-Lee, James Hendler and Ora Lassila.The Semantic Web. *Scientific American*, May 2001.

L. A. G. Dacosta, P. F. Pires, and M. Mattoso. Automatic Composition of Web Services with Contingency Plans. *Proc. Int’l Conf. on Web Services* (ICWS ’04)., 2004.

Snehit Prabhu. Towards Distributed Dynamic Web Service Composition. *Proc. Int’l Conf.Symposium on Autonomous Decentralized System* (ISADS ’07), 2007.

Liangzhao Zeng; Benatallah, B.; Ngu, A. H. H.; Dumas, M.; Kalagnanam, J.; Chang, H. QoS-aware middleware for Web services composition. *IEEE Transactions on Software Engineering*, pp. 311 - 327, Volume 30, Issue 5, May 2004.

M. MacKenzie et al. OASIS - Reference Model for Service Oriented Architecture 1.0. http://www.oasis open.org/, 2006.