Journal of Geographic Information System, **2010, 2, 40-44**
doi:10.4236/jgis.2010.21008 Published Online January 2010 (http://www.scirp.org/journal/jgis)

A QoS-aware Method for Web Services Discovery

Bian WU, Xincai WU[1,2]
[1]*Research Center for GIS Software and Application Engineering, Ministry of Education, Wuhan, China*
[2]*School of Earth Sciences and Resources, China University of Geosciences, Beijing, China*
*Email: nimo*299@126.*com*

Abstract: The non-functional QoS (quality of service) information helps us to select a proper Web-service from the web applications, by using component services such as UDDI[1](Universal Description, Discovery, and Integration) and MDS(Monitoring and Discovery System). MDS is a suite of web services to monitor and discover resources and service on Grids, but MDS only based on function aspects. This paper studies on an approach to provide the QoS information and a discovery model by using MDS and gives a system deployment and implementation plan. The simulation results show that the method is effective in service discovery.

Keywords: quality of service (QoS), web services, monitoring and discovery system (MDS)

1. Introduction

Web services are frequently used to build a distributed system which can be accessed over Internet. It is loosely coupled to enhance productivity, simplify using, get reusability and improve system expansibility. But with the widespread proliferation, the QoS becomes an important factor in distinguishing the success of a web service application.

QoS is the ability to provide different priority to different applications and users to guarantee a certain level of performance to data flow. Typical QoS properties associates with a web service execution cost and time, availability, successful execution rate, reputation, and usage frequency [2]. The QoS covers a range of non-functional properties of Web-service such as response time, performance, security, availability and reliability. The service providers provide differentiated capacity model for different service types to ensure appropriate QoS levels for different customers' applications. For example, a complex spatial analysis Web service might require good throughput, while a simple searching Web service might require a good response performance.

In order to describe the contract between the service provider, the customer and the Grid system, several specifications defined by XML based language have been proposed such as WSDL, which is complement to the service description implemented, WS-Agreement specification [3] which is used within a framework allowing the management of web services and their compositions.

In this work we propose to modify the WSDL file in order to increase QoS support to the web service. Then we propose a method of web services discovery on GT4 MDS.

1.1. Web Service QoS Requirements

The major requirements[4] for supporting QoS in Web services are as follows:

- **Regulatory** is the aspect of the Web service in compliance with the standards such as SOAP, UDDI, WSDL, etc and the established service level agreement. Strict adherence to correct versions of standards by service providers is necessary for proper invocation of Web services by service consumers.
- **Availability** is the quality aspect of whether the Web service is ready for immediate use. It represents the probability that a service is available. Larger values represent that the service is always ready to use while smaller values indicate unpredictability of whether the service will be available at a particular time.
- **Reliability** is the quality aspect of a web service that represents the capability of maintaining the service and service quality. The number of successfully used per month represents a measure of reliability of a web service.
- **Performance** is the quality aspect of Web service, which is measured in terms of throughput and latency, and response time. Throughout represents the number of Web service requests served at a given time period. Latency is the round-trip time between sending a request and receiving the response. Higher throughput, lower

latency and less response time values represent good performance of a GIS Web service.

- **Security** is the quality aspect of the web services of providing confidentiality by authenticating the parties involved, encrypting messages, and providing access control. Because distributed GIS system is deployed and invoked over the public Internet, security becomes more important. The service providers have different approaches and levels of providing security depending on the service requester.

2. QoS-Based Web Services Discovery Method

MDS is the information services component of the Globus Tookit4 and provides information about the available resources on the Grid and their status [5]. It is a suite of web services to monitor and discover resources and services on Grids[6]. MDS builds on query, subscription, and notification protocols and interfaces defined by the Web Services Resource Framework (WSRF) and Web Services Notification families of specifications and implemented by the GT4 Web Services Core. Discovery is the process of finding a suitable resource to perform a task. This process may involve both finding which resource are suitable and choosing a suitable member from that set. The discovery process requires the ability to collect information from multiple, perhaps distributed information sources. To meet this need, MDS provides so-called aggregator services that collect recent state information from registered information sources. The MDS architecture is shown in Figure 1.

2.1. QoS Properties in WSDL

Web Service Description Language (WSDL) is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information[7]. Specifically, it defines via its portType component, the Web service's abstract interface, specifying the operations that the service supports and for each operation, the format of the messages that the service sends and receives. The client programs read the WSDL files to connect and communicate with the service. The current WSDL file contains the following information: the number of inputs and outputs, the variable type of each input and output, the order the inputs are given and outputs are returned, and how the Web service should be invoked. But from the WSDL file, we can not get the QoS information. To solve this problem we propose to extend the WSDL to support the QoS information description.

The hierarchical QoS semantics model[8] which is an extension of WSDL document structure (shows in Figure. 2), can be used to describe the quality information of Web Services. Specifically the method for describing the quality information of GIS Web Services properties is to increase QoS elements through the extension of WSDL protocol. Add atQualityInfo property in the WSDL tport element to describe the five QoS attributes of Web services, the tport elements are regulatory, response time, availability, performance, and security level.

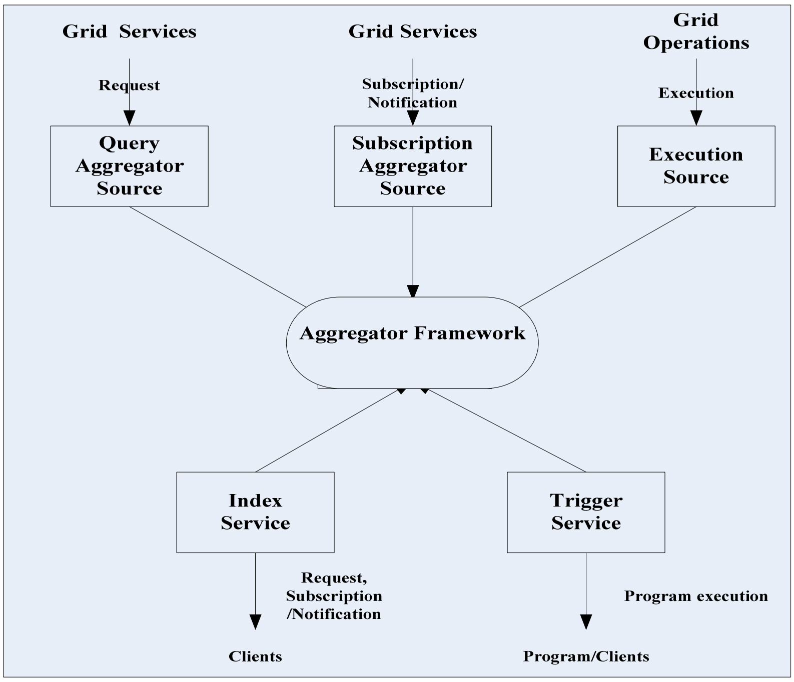

Figure 1. MDS architecture

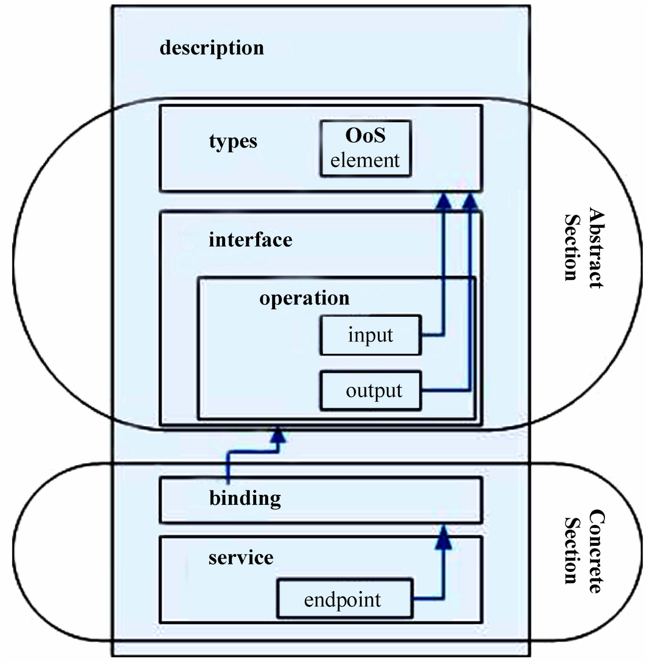

Figure 2. WSDL document structure

The Web services function elements correspond with tport in the general WSDL documents. So add the qualityInfo attributes into the tport element of WSDL to describe the QoS[9]. An example is as follows:

```
< wsdl: qualityInfo name="approve"
qualityID=
"DAJD0891-28da-48B3-234J-24IA2394C342">
< wsdl: constraintInfo name="regulatory" value="75"
probability="9"
constraintID="D2JD2371-4H7P-15A3-414f-15I323r4
C342"/>
< wsdl: constraintInfo name="ResponseTime" value=
"27" probability="28"
constraintID="QEJD1851-RF36-41G6-414f-15I323r4
C342"/>
< wsdl: constraintInfo name="availability" value="65"
probability="39"
constraintID="DBYT5769-2T4P-W1A3-414f-15I323r
4C342"/>
< wsdl: constraintInfo name="performance" value=
"23" probability="27"
constraintID="FUJJ3861-1G5P-1EA7-414f-15I323r4
C342"/>
< wsdl: constraintInfo name="security level" value=
"7" probalility="56"
constraintID="D2JR3431-Y68L-D2A3-414f-15I323r4
C342"/>
< /wsdl: qualityInfo>
```

```
<types>
<xsd:schema targetNamespace="http://localhost/examples/core/QoS_instance"
xmlns:tns="http://localhost/examples/core/OoS_instan
ce"
xmlns:xsd="http//localhost/2012/XMLSchema">
...
<xsd: element name = "regulatory">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="tns:Value" minOccurs="1" maxO
ccurs="9"/>
</xsd:sequence>
<xsd:complexType>
</xsd:element>
...
</xsd:schema>
</types>
...
<xsd:element name="Value" type="xsd:int"/>
...
```

2.2. Modifying Query Aggregator Source Configuration File

Besides editing WSDL documents. another necessary step is to increase QoS support to the Web Services' XML schema documents, and through the extension of the GetMultipleResourcePropertiesPollType which is one of the parameters for Query Aggregator Source in configuration file. The QoS properties can be retrieved from a Index Service as resource properties.

The Aggregator Framework is a software framework used to build services that collect and aggregate data.

Index Services built on the Aggregator Framework are sometimes called aggregator services. The Index Services collect XML-formatted data via Aggregator Source. The Query Aggregator Source is a java class that implements an interface to poll a WSRF service for resource property information. An example is as follows:

```
  <Content

xmlns:agg="http://mds.globus.org/aggregator/types"
        xsi:type="agg:AggregatorContent">
     <agg:AggregatorConfig
xsi:type="agg:AggregatorConfig">
        <agg:GetMultipleResourcePropertiesPollType>

<agg:GridGISServices>Spatial_Operations</agg:GridGI
SServices>

<agg:ResourcePropertyNames>rp1_namespace:regulator
y</agg:ResourcePropertyNames>

<agg:ResourcePropertyNames>rp1_namespace:response
_time</agg:ResourcePropertyNames>

<agg:ResourcePropertyNames>rp1_namespace:availabili
ty</agg:ResourcePropertyNames>

<agg:ResourcePropertyNames>rp2_namespace:performa
nce</agg:ResourcePropertyNames>

<agg:ResourcePropertyNames>rp3_namespace:security_
level</agg:ResourcePropertyNames>
        </agg:GetMultipleResourcePropertiesPollType>
     </agg:AggregatorConfig>
     <agg:AggregatorData/>
  </Content>
```

Then Query Aggregator Source will request five QoS properties in the Grid services. There is no limit on the number of “ResourcePropertyName” parameters that can be specified.

Qos properties provide support for service publishing and selection, but how to select web services that can well meet the requirements of Web services' consumers becomes a problem.

2.3. Web Services Discovery from MDS

The users use the web browser to send a HTTP request, including some web form arguments, to the web server container. The web server invokes the MDS servlet, which uses the form arguments to determine what plugins to use to retrieve the request GML data and the XSLT transform tools to apply to it. Then use the standard WSRF resource property query interfaces to retrieve service WSDL information from an Index. Every GT4 web services container includes a default MDS-Index service with which any GT4 services running in that container are automatically registered. Thus, each Grid system has an index that allows one to discover what services are available. GT4 is configured to use MDS mechanisms to good effect for discovering and monitoring of GT4 services. Every GT4 web services supports a minimal set of QoS properties and thus can be registered easily into one or more aggregators for monitoring and discovery. The MDS use servlet passes arguments to the transaction tool, which then retrieves the appropriate data and XSLT transform. The MDS servlet applies the XSLT transformation to the XML data and returns the result to the web server, which sends it back to the client's web browser. The discovery model is shown in Figure 3.

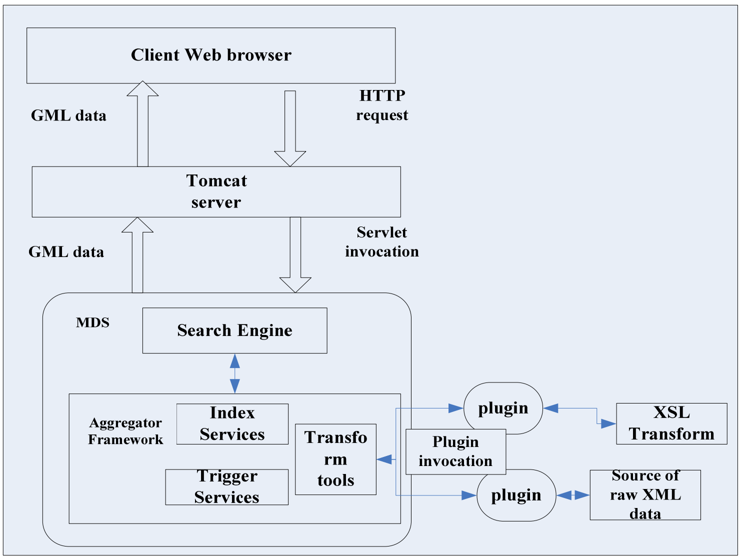

Figure 3. Discovery model

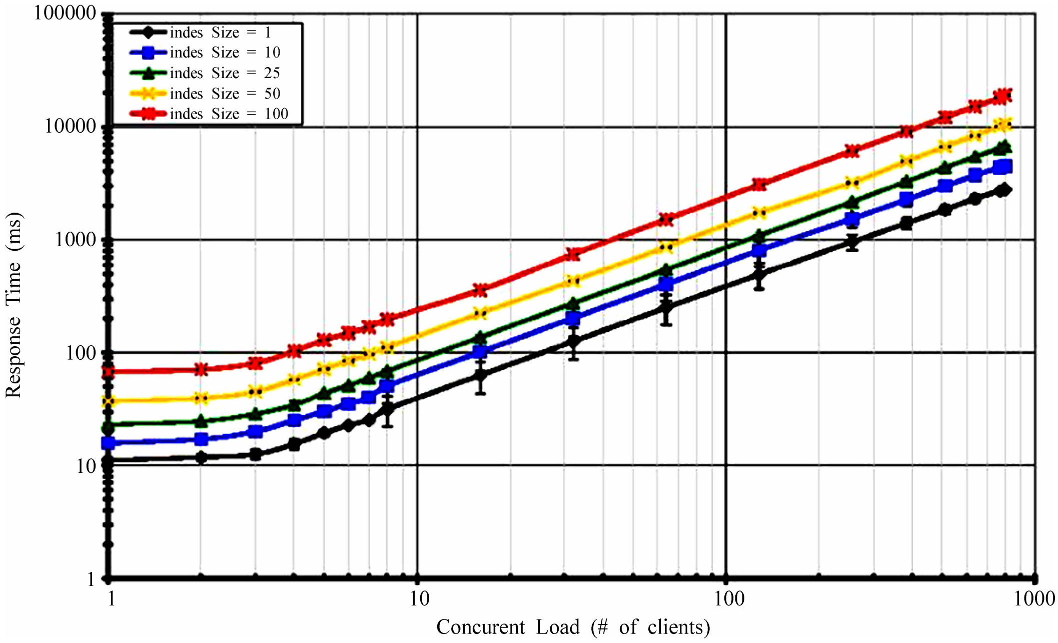

Figure 4. Testing performance

3. Deployment and Testing

Testing executions were performed on a Intel Core 2 Duo T9300@ 2.5GHz machine, with DDR 666MHz 3.0G of RAM and Windows XP sp3 operation system as web server. Java as Developing language, IDE: ecplise5.0, Apache Tomcat as web services container IE7.0 as web browser, GD4 as Grid container and Grid system environment Figure 4 shows the testing results.

4. Conclusion

The work proposed in this paper provides an approach to describe Web services QoS information in WSDL and a discovery model in which the functional and non-functional requirements is to be taken into account for the selection. The test results illustrating the method have high performance in practices.

5. Acknowledgments

We would like to thank Professor Fujiang Liu for his valuable comments and suggestions which have helped us to improve the quality, organization, and presentation of our manuscript.

REFERENCES

[1] Clement, L. et al. [Ed.] UDDI Version 3.0.2 [DB/OL] http://www.uddi.org/pubs/uddi_v3.htm [viewed 12/6/2009].

[2] O'Sullivan.J., Edmond, D.,Hofstede, A.T.,2002. What's in a service? Distributed and Parallel Database 12(2-3), 117-133.

[3] Andrieux, A. et al. 2007 Web Services Agreement Specification (WS-Agreement), OGF proposed recommendation (GFD.107) <http://www.ogf.org/documents/GFD.107.pdf>.

[4] Mani, A. ,Nagarajan, A., 2002. Understanding quality of service for web services, White paper, IBM.http://www.ibm.com/developerworks/library/ws-quality.html?S_TACT=105AGX52&S_CMP=cn-a-ws [viewed 11/1/2009].

[5] GT4.2.1:Information Services [DB/OL]

[6] http://www.globus.org/toolkit/docs/latest-stable/info/#info [viewed 11/2/2009].

[7] Information Services (MDS): Key Concepts http://www.globus.org/toolkit/docs/4.0/info/key-index.html [viewed 10/2/2009].

[8] Christensen, E. et al. Web Services Description Language (WSDL) 1.1

[9] http://www.w3.org/TR/wsdl [viewed 01/10/2009].

[10] Henry B Applying propositional logic to workflow verification [J]. Information Technology and Management, 2004, 5: 293-318.

[11] Mao, Y. &Li, Z.(2009) Research on QoS-based Web service discovery. Journal of Shandong University(Natural Science) Vol.44, No.7, pp. 89-92.