Analyzing QoS for Web Service Compositions by QoSDIST

Huiyuan Zheng, Jian Yang, and Weiliang Zhao

Department of Computing, Macquarie University, Australia
{huiyuan.zheng,jian.yang,weiliang.zhao}@mq.edu.au

Quality of Service (QoS) analysis and prediction for Web service compositions is an important and challenging issue in distributed computing. In existing work, QoS for service compositions is either calculated based on constant QoS values or simulated based on probabilistic QoS distributions of component services. Simulation method is time consuming and can not be used in real-time applications for dynamic Web service compositions. Experimental results in [5] show that our proposed QoS calculation approach significantly improves the efficiency in QoS estimation when the QoS of component Web services are probability distributions. In this paper, we present a tool QoSDIST to analyze the QoS for Web service compositions based on our proposed QoS calculation method in [5]. QoSDIST does not put any constraints on the modeling of the QoS of component Web services, i.e., the QoS of a component Web service can be in single value, discrete values with frequencies, standard statistical distribution, or any general distribution regardless of its shape.

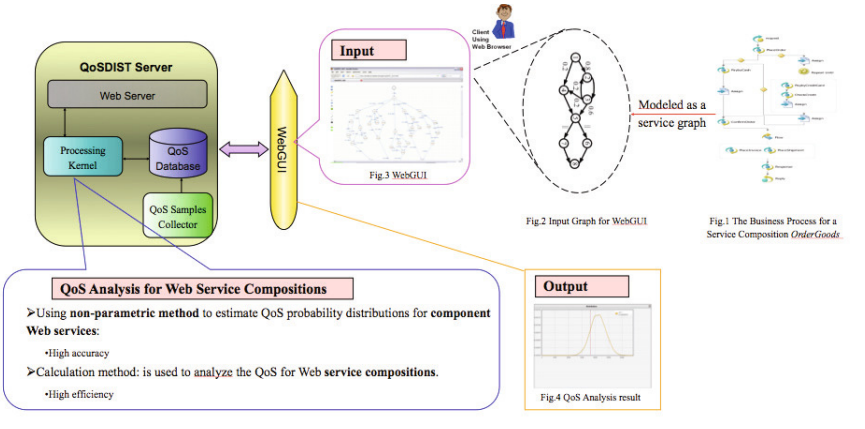

Fig. 1. System Architecture of QoSDIST

Figure 1 shows the system architecture of QoSDIST. Estimating the QoS distributions of a service composition involves a large amount of computing. Therefore, QoSDIST adopts the AJAX client-server architecture. The server is in charge of most of the processing. A Web-based graphical user interface

G. Pallis et al. (Eds.): ICSOC 2011, LNCS 7221, pp. 246–247, 2012.

(WebGUI) runs on the client side which is based on JavaScript[1]. The input information from WebGUI is encoded into XML by the clients and then sent to the server side. On receiving the XML document, the server side will start processing it and return the processing result in XML. QoSDIST involves four major modules: a WebGUI, a QoS Collector, a QoS Database, and a Processing Kernel. The functions of these modules are as follows:

`WebGUI`: *WebGUI* is on the client end of QoSDIST. Through *WebGUI*, a user can input the service graph [6] of a Web service composition and the QoS information for the component Web services in the service composition. The input information will be encoded into XML format and transmitted to the server side of QoSDIST. The QoS analysis result for a service composition will also be output through *WebGUI*.

`QoS Collector`: *QoS collector* is in charge of collecting QoS data for Web services by testing the QoS of Web services at a regular time interval [3]. These collected QoS data for per QoS metric of a Web service is referred to as a QoS sample for that QoS metric of the Web service. QoS samples collected by *QoS Collector* will be transformed into probability distributions based on the method proposed in [5] by *Processing Kernel* and stored at *QoS Database*.

`QoS Database`: The QoS of component Web services are stored at *QoS Database*. The QoS can be in single value, discrete values with frequencies, or probability distributions.

`Processing Kernel`: *Processing Kernel* generates the probability distributions for component Web services based on the samples collected by *QoS Collector* [4]; decodes the received XML information from the client side and gets a graph structure for the service composition; analyzes the QoS of the service composition based on the method proposed in [5]; outputs the QoS analysis result to *WebGUI*.

In this demonstration, we will show how to use QoSDIST to draw the service graph of a Web service composition, edit the properties of the components in the service graph, run the QoS analysis, and read the QoS analysis results for Web service compositions. A screencast video of the demonstration is available at online demonstration for ICSOC2011.

References

1. `http://www.jgraph.com/mxgraph.html`
2. `http://code.google.com/p/flot/`
3. Rosenberg, F., Platzer, C., Dustdar, S.: Bootstrapping performance and dependability attributes of web services. In: ICWS, pp. 205–212 (2006)
4. Zheng, H., Yang, J., Zhao, W.: Qos probability distribution estimation for web services and service compositions. In: SOCA (2010)
5. Zheng, H., Yang, J., Zhao, W., Bouguettaya, A.: QoS Analysis for Web Service Compositions Based on Probabilistic QoS. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R. (eds.) ICSOC 2011. LNCS, vol. 7084, pp. 47–61. Springer, Heidelberg (2011)
6. Zheng, H., Zhao, W., Yang, J., Bouguettaya, A.: Qos analysis for web service composition. In: SCC, pp. 235–242 (2009)

[1] mxGraph[1] and flot[2] libraries are used.