Sustaining High-Availability and Quality of Web Services

Erbin Lim and Philippe Thiran

PRECISE Research Centre
Faculty of Computer Science
University of Namur

Abstract. In this work, we study how to sustain the high availability and quality of Web services by using communities of Web services. Availability is the probability that a Web service is in functioning condition at a specific time. Communities of Web services gather Web services that provide the same functionality, but not necessary with the same quality. Within this framework, an available Web service can be selected or even substituted by another one when it fails. In this research, we aim at investigating how to manage the community to sustain not only the high availability but also to guarantee the quality of service expected by the user.

1 Introduction

Web services are intensively used for developing loosely-coupled, inter-enterprise business processes. Ideally, a Web service is required to accept all the requests of the users. However, a Web service can get busy with an overloaded demand and thus, cannot offer a good quality of service to the number of users that exceeds its capacity. This situation would cause inefficiency in offering service quality and consequently lead to a drop in the satisfaction value of the Web service that is assigned by the users.

1.1 Availability

Availability is the probability that a Web service is in a functioning condition at a specific time. Sustaining high availability of a Web service is a challenging but crucial issue. For a Web service, failing to respond with an acceptable quality of service would cause a negative satisfaction of the users, and hence, drop the future number of requests.

Traditionally, the most common solution to sustain availability is the use of *replicas* which are managed according to specific replication strategies and are used when the original Web service fails [10, 11, 16]. Another solution is to group similarly-functional Web services into *communities* to improve their overall performance and availability [17]. A community of Web services *selects* an available Web service or even *substitutes* the selected Web services when it fails.

1.2 Quality of Service

Replicas or Web services of a community can differ with their quality of service. Ideally, replication, selection or substitution should guarantee the quality expected by the user, and thus his satisfaction. The user satisfaction must be maintained so that a user request can be rejected if the offering service quality is not sufficient. However, rejecting a user request can also impact the user satisfaction. Hence, we need to measure and analyze the tradeoff between availability and user satisfaction.

Quality of Service can be divided into two categories [1], the first category includes objective parameters while the second category includes subjective parameters. Objective parameters are defined as parameters that can be measured quantitatively. For example, *number of requests*, *number of successfully handled requests*, and *uptime* can be considered as objective parameters; while *user-perceived availability*, *attractiveness* can be considered as subjective parameters.

1.3 Scope of the Thesis

The objective of the thesis is to sustain high-availability and quality of services. In the thesis, we use the communities of Web services which are proposed by [17]. The main contribution of this paper lies on the QoS issues in the selection and substitution process within the communities.

2 Community of Web Services

Benatallah et al. [5] define a community as a collection of Web services with a common functionality, although these Web services have distinct non-functional properties like different providers and different QoS parameters. In this work, we consider a community as a means for providing a common description of a desired functionality (e.g., FlightBooking) without explicitly referring to any concrete Web service (e.g., EKFlightBooking) that will implement this functionality at run-time [4].

Figure 1 represents the architecture of communities of Web services. The components of this architecture include Web services providers (WSP), Universal Description Discovery and Integration (UDDI) registries, and communities. A community is dynamic by nature. It is established and dismantled according to specific scenarios and protocols. UDDI registries receive advertisements of Web services from providers.

Two communities of Web services are shown in the figure. They could for example offer AirfareQuotation and HotelBooking functionalities, respectively. A *Master Web service* (MWS) always leads a community. The other Web services are denoted as *slaves* (SWS). Slave Web services offer the same functionality but with possibly different qualities of services.

One of the responsibilities of the master Web service is to attract Web services to sign up in the community it heads using rewards. As a result, the master Web-service interacts with the UDDI registries on a regular basis, so it gets to know the latest changes in the content of these UDDI registries such as advertisements of new Web services. Additional responsibilities of the master Web service are:

Fig. 1. Community of Web Services

- to accept or invite new Web services to be members of the community and to exclude or reject existing Web services;
- to nominate the Web service that will receive the user request.

The MWS exists in a community as a figurehead to control the SWS and the management of requests. The selection of the MWS is a conceptual one.

3 Problem Definition

The research question can be formulated as follows:

How can we sustain high-availability and the expected quality of Web services using communities of Web services?

This research will look into the research issues when using similarly functional Web services with different QoS. The problems can be divided into two different categories:

- *Member management*, which involves all problems related to the evaluation of the number of SWS and their quality;
- *User request management*, which includes all problems related to the nomination (selection or substitution) of the SWS that will receive the user request.

3.1 Member Management

In a community of Web services, different SWS provide concrete Web services with different QoS. The number and quality of SWS within the community can directly impact the number of participation requests and the user satisfaction.

This drives the MWS to regularly check the *size* of the community and the quality of services provided by the SWS. As a result, a MWS oversees all the events in a community such as arrival of new Web services, departure of some Web services, sanctions on Web services due to unavailability, misbehavior, etc.

If the MWS notices that first, the number of SWS is less than a certain threshold and second, the number of requests that arrive from users over a certain period of time is high than another threshold, the MWS should invite new Web services to join the community [13]. Conversely, when the number of SWS in a community is high but the number of user requests is low, this could lead the MWS to eject some Web services with a low level of QoS and availability from the community and to invite other Web services with better performance to join the community.

In a community, SWS compete to increase their participation. MWS nominates the available SWS that can answer efficiently user request. Thus, an excellent user satisfaction can drive the MWS to prefer certain SWS. This can lead to two undesirable situations:

- Overloading of the efficient SWS that could decrease the offering service quality;
- Leaving of SWS as their participation rate in the community is insufficient.

3.2 User Request Management

A user request is handled by the MWS that selects the most efficient SWS that will answer it. Sometimes the selected SWS can be overloaded, and consequently becomes unavailable. If the best selected SWS becomes unavailable, MWS would substitute it by asking the second best SWS in terms of similar QoS, user satisfaction (and so on).

Finding the most efficient SWS for a user request is not a trivial task. Users have different requirements in terms of the expected QoS. Moreover, they rate differently their satisfactions about the received quality and responsiveness.

Moreover, a MWS must reserve his most efficient SWS for certain users. Indeed, all user requests do not necessary require a high QoS. Hence, these requests can be handled by low efficient SWS so that the most efficient SWS are kept available for future requests.

4 Research Methodology

The first part of the research will introduce the framework of a community of Web services. This also includes the definition and evaluation of QoS metrics of each individual SWS. This provides the basis to define the QoS metrics for the community as a whole.

The next step will be to define the member management of the community. This process involves creating mechanisms and heuristics to determine how to manage the members in the community. The mechanisms are used to determine the optimal number and quality of members within a community.

The user request management comes next. Mechanisms are to be used to select the best available SWS to serve a user request. In line with the goal of achieving high availability, it is also crucial to define a substitution mechanism to handle the situation where a SWS fails. Both the selection and substitution mechanisms use the QoS metrics defined in the first part.

An experiment or prototype would be critical in determining the effectiveness of the proposed mechanisms. This can be achieved by running experiments to adjust the parameters in the proposed mechanisms to determine if they achieve the desired result.

5 Related Work

The use of community of Web services to sustain High-availability was mentioned in [17]. But the authors do not mention the need for sustaining a certain level of QoS. To determine the size and type of a community, the level of QoS of the community needs to be calculated. Similarities between calculating the QoS of a community of Web services can be found with calculating the level of QoS in a composition of Web services [7, 6].

The selection process when a request is handled by the community is similar to the selection process within a composition of Web services [18, 3, 12]. The substitution process is the main contributing factor to sustaining high-availability and QoS in Web services. Similar work has also been done in the area of web services discovery in [15]. In [2, 14], the authors select the web service based on QoS metrics and client preferences.

In [8], the authors present a tool that sustains the service quality within workflow management systems. They use mathematical models to meet specific goals for performance and availability due to outages of individual servers. This is similar to what we are proposing because we aim to sustain the quality of service of the web service even if the original service provider were to fail.

Most literature consider *availability* and QoS parameters as a deterministic value. However, as mentioned earlier, we consider *availability* and various QoS parameters to be based on a probability. This approach was also mentioned in [9].

6 Work in Progress

To date, we have reviewed the literature of community of Services and its QoS metrics. We have also started to define the QoS metrics to be used in our research. We have compared the framework of a community of Web services with current more researched frameworks, for example, a composite of Web services.

7 Conclusion

In this paper we have outlined the project proposal for a method of sustaining high availability and quality of Web services. This research creates a framework to allow communities of Web services to achieve this goal.

We believe that our research contributes to the area of Web engineering by providing a new approach to sustain high availability and quality of Web services.

References

1. Zeitham, V.A., Parasuraman, A., Berry, L.L.: Quality counts in service too. *Business Horizons* (May-June 1985)
2. Al-Masri, E., Mahmoud, Q.H.: Discovering the best web service. In: *WWW 2007* (May 2007)
3. Ardagna, D., Pernici, B.: Global and local qos guarantee in web service selection. In: *First International Workshop on Business Processes and Services* (2005)
4. Bentahar, J., Maamar, Z., Benslimane, D., Thiran, P.: An argumentation framework for communities of web services. *Intelligent Systems* 22(6), 75–83 (2007)
5. Sheng, Q.Z., Benatallah, B., Duman, M.: The self-serv environment for web services composition. *IEEE Internet Computing*, 40–48 (January-February 2003)
6. Canfora, G., Di Penta, M., Esposito, R., Luisa Villani, M.: A lightweight approach for qos-aware service composition. In: *Proc. 2nd International Conference on Service Oriented Computing (ICSOC 2004)* (2004)
7. Cardoso, J., Miller, J., Sheth, A., Arnold, J.: Modeling quality of service for workflows and web service processes. *VLDB Journal* (2002)
8. Gillmann, M., Weikum, G., Konner, W.: Workflow management with service quality guarantees. In: *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 228–239 (2002)
9. Hwang, S.-Y., Wang, H., Tang, J., Srivastava, J.: A probabilistic approach to modeling and estimating the qos of web-services-based workflows. *Science Direct* (2007)
10. Osrael, J., Frohofer, L., Goeschka, K.: What service replication middleware can learn from object replication middleware. In: *Proceedings of the Workshop on Middleware for Service Oriented Computing held in conjunction with The ACM/IFIP/USENIX 7th International Middleware Conference*, pp. 18–23 (2006)
11. Salas, J., Perez-Sorrosal, F., Patino-Martinez, M., Jimenez-Peris, R.: Ws-replication: A framework for highly available web services. In: *Proceedings of the 15th International World Wide Web Conference (WWW 2006)*, pp. 357–366 (2006)
12. Liu, Y., Ngu, A.H.H., Zeng, L.: Qos computation and policing in dynamic web service selection. *ACM, New York* (2004) 1-58113-912-9/04/0005
13. Maamar, Z., Subramanian, S., Thiran, P., Benslimane, D., Bentahar, J.: An approach to engineer communities of web services - concepts, architecture, operation, and deployment. *International Journal of E-Business Research* 5(4) (2009)
14. Maximilien, E.M., Singh, M.P.: Toward autonomic web services trust and selection (2004)
15. Ran, S.: A model for web services discovery with qos. *ACM, New York* (2003)
16. Ye, X., Shen, Y.: A middleware for replicated web services. In: *Proceedings of 2005 IEEE International Conference on Web Services, ICWS 2005*, p. 638 (July 2005)
17. Benslimane, D., Maamar, Z., Sheng, Q.Z.: Sustaining web services high-availability using communities. In: *2008 Third International Conference on Availability, Reliability and Security* (2008)
18. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven web services composition. In: *Proceedings of the 12th International Conference on World Wide Web*, pp. 411–421 (2003)