Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 321 (2016) 113–127

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Quality of Service Conflict During Web Service Monitoring: A Case Study

Jael Zela Ruiz[1,2] Cecília M. Rubira[3]

Institute of Computing
University of Campinas
Campinas, São Paulo, Brazil

Abstract

Web services have become one of the most used technologies in service-oriented systems. Its popularity is due to its property to adapt to any context. As a consequence of the increasing number of Web services on the Internet and its important role in many applications today, Web service quality has become a crucial requirement and demanded by service consumers. Terms of quality levels are written between service providers and service consumers to ensure a degree of quality. The use of monitoring tools to control service quality levels is very important. Quality attributes suffer variations in their values during runtime, this is produced by many factors such as a memory leak, deadlock, race data, inconsistent data, etc. However, sometimes monitoring tools can impact negatively affecting the quality of service when they are not properly used and configured, producing possible conflicts between quality attributes. This paper aims to show the impact of monitoring tools over service quality, two of the most important quality attributes - performance and accuracy - were chosen to be monitored. A case study is conducted to present and evaluate the relationship between performance and accuracy over a Web service. As a result, conflict is found between performance and accuracy, where performance was the most affected, because it presented a degradation in its quality level during monitoring.

Keywords: Web Services, SOA, Quality of Service, Quality Attributes, Conflict, Performance, Accuracy, Monitoring Tools.

1 Introduction

In recent years, the Web service technology has become the most popular and used technology to build SOA applications [26]. Web services are based in a set of protocols and standards as SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), and UDDI (Universal Description, Discovery, and Integration). Web services are distributed components which are self-contained, discoverable, reusable, composable, and have a transparent location [3]. As a result

[1] This work was supported by the Laboratory of Distributed Systems and Software Engineering, at Institute of Computing, UNICAMP, Campinas, Brazil.

[2] Email: jael.ruiz@students.ic.unicamp.br

[3] Email: cmrubira@ic.unicamp.br

http://dx.doi.org/10.1016/j.entcs.2016.02.007
1571-0661/

of its popularity, an increasing number of functionally similar Web services can be found on the internet [7], which entails to the service consumer to ask the question: "what are the better services?" or "which of them better fit my needs?" [6]. Service consumers have a difficult task to choose an appropriate service for their requirements. Quality of Service (QoS) has become the most appropriate criterion to distinguish non-functional characteristics between equivalent Web services.

QoS is described as a number of properties, named quality attributes, which take in play the Web service quality. Some of these attributes are, for instance, availability, throughput, robustness, and integrity where a set of quality attributes compose a quality model. Currently, there are several quality models proposed in the academia and in the industry [3] [6] [19] [20]. Web services promise quality levels based on quality models. A negotiation between the service provider and the service consumer is carried out, in order to assure a specific level of QoS for Web services. A Service Level Agreement (SLA) is the result of this negotiation, where quality is defined, negotiated and tasks to assure quality are established [16]. Nevertheless, afterwards an SLA is arranged for both parties, a new question is asked by service consumers: How can we be sure that the supposed QoS defined in the SLA is really satisfied?. As a consequence, monitoring tools emerge to control the Web service quality levels. Monitoring tools are based on quality model. They are used to capture, collect, filter, and analyse information from the Web service during runtime [8]. Currently, there are many monitoring tools which come from the research and the industry, such as Dynamo [4], Cremona [14], SALMon [1], WebInject [11], SOAP Monitor [2], Webmetrics Web Services Monitoring [18], FlexMonitorWS [10].

However, because of the dynamic and unpredictable nature of Web services [12], quality attributes can suffer variations in their values during runtime. The relationship among quality attributes can produce conflicts between them when they are monitored at the same time. For example, in a response time and throughput scenario, response time can be a better quality value when it is monitored in isolation than in parallel with throughput. The reason is because the Web service receives a larger number of requests, producing that the Web service takes more time to respond to the user. On the other hand, throughput is also affected, because a small number of requests are attended by unit of time, due to the required time to respond each request. These conflicts are produced mainly for scalability reasons, Web services can have many service consumers sending many service requests at the same time. Monitoring tools become a factor for quality attribute conflicts.

Monitoring tools can become a double-edged sword, because they are a useful QoS control tool, but they can become the principal reason for conflicts when they are not properly configured. They can turn out to be an intrusive agent for the Web service, creating a stressful environment. This is important to know what is being measured, where you are monitoring, how it is being monitored, and how frequently it is monitoring. An active monitoring not properly configured can overload the Web service and produce a breakdown in the values for response time, throughput, or availability to current consumers. In order to demonstrate the monitoring tools effects over the Web service quality, we present a case study to measure the quality