IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA ITJ2622

The Design of QoS Broker Algorithms for QoS-Capable Web Services

Tao Yu, University of California, Irvine, USA
Kwei-Jay Lin, University of California, Irvine, USA

ABSTRACT

QoS (quality of service) support in Web services is an important issue since it ensures service usability and utility for each client and, in addition, improves server utilization. In this article, we present a QoS-capable Web service architecture, QCWS, by introducing a QoS broker module between service clients and providers (servers). The functions of the QoS broker module include tracking QoS information about servers, making selection decisions for clients, and negotiating with servers to get QoS agreements. We study two resource allocation algorithms (HQ and RQ) used by QoS brokers acting as the front-end of servers. The goals of the algorithms are to maximize the server resource usage while minimizing the QoS instability for each client. The first algorithm, HQ, assigns a homogeneous service level to all clients on the system and adjusts the service level according to the number of active clients. The second algorithm, RQ, assigns different service levels to clients according to their needs. Both algorithms try to minimize the resource reallocations for existing clients. The QoS performance and instability trade-offs are studied by simulation.

Keywords: HQ; QoS broker; QoS instability; resource allocation algorithm; RQ; server utilization; Web services.

INTRODUCTION

As self-contained, self-describing modular applications, Web services have been universally deployed and easily accessible on the Web. They are becoming the prominent system software paradigm for distributed computing and e-businesses. Web services interact with each other, fulfilling tasks and requests that, in turn, carry out parts of complex transactions or workflows. Standards such as SOAP, WSDL and UDDI have been proposed and generally adopted as the foundation for an open Web service framework.

However, most of today's Web service implementations do not guarantee the levels of service quality delivered to their users. At present, UDDI is just a registry database that allows clients to look for Web services based on their functionality but not QoS (quality of service) property. QoS, which defines service quality such as latency, availability, timeliness and reliability, is important for many Web applications that provide real-time information, multimedia content, or time-critical services. Questions on the QoS level of a service such as "Can I get the result in 20 seconds?" or "Is the service reliable?" usually are critical to a user before a service is invoked. The lack of adequate QoS support has impaired the adoption of many Web services.

In this research, we first design a general QoS-Capable Web service architecture (Chen et al., 2003), QCWS, by defining a QoS broker module between Web service clients and providers. In QCWS, the QoS broker module collects the QoS information about service providers (servers), makes service selection decisions for clients, and then negotiates with some of the servers to meet the QoS requirements. For implementation, a QoS broker may be part of a client, part of a server, or an independent Web service (much like broker agencies). The functionalities of a broker may vary depending on the setting of the broker.

In this article, we focus on systems where the QoS broker acts as the front-end of a server. In this setting, the main function of a broker is to help servers decide how much resource should be assigned to each new client to meet its QoS needs and, once assigned, minimize the QoS fluctuation during its execution. We study two resource allocation algorithms used by brokers. The goal of both algorithms is to achieve the best overall system utility while keeping the QoS reconfiguration rate low so as not to disrupt the stability of existing client activities. The first algorithm, HQ, is used by servers that do not provide different levels of service quality to clients. (Most of today's servers fall into this category.) Every client receives the same amount of system resources. The algorithm adjusts the resource assigned according to the number of active clients in the system. The second algorithm, RQ, is designed for QoS-enabled servers that may provide different service quality levels to different clients. The algorithm reserves a certain amount of resources in a virtual client so that future clients may receive a satisfactory service level using the reserved resource. Our algorithms are based on the resource allocation policies for dynamic systems (D-QRAM, Hansen et al., 2001) but improve on them by using more feedback information on the actual system load.

The significance of our work is to promote QoS support in Web services so that clients may receive a consistent service level regardless of other competing requests on the same server. The broker algorithms presented in this article can be used for both legacy and QoS-capable servers. Without such resource allocation and admission control mechanisms, users may experience intrinsically unstable service quality depending on the dynamic server workload.

The rest of the article is organized as follows. We first briefly review related work on QoS research. Then we present the general QoS-capable Web service architecture, QCWS. We introduce the resource allocation algorithms used by QoS brokers that act as the front-ends of servers in the following section. Finally, we present the simulation study of the algorithms.

RELATED WORK

Recently, issues on supporting QoS in Web services have received much research interest. Some have suggested that "… QoS will become a significant factor