Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Network and Computer Applications 28 (2005) 57–74
Journal of NETWORK and COMPUTER APPLICATIONS
www.elsevier.com/locate/jnca

Using service brokers for accessing backend servers for web applications

Huamin Chen*, Prasant Mohapatra

Department of Computer Science, Engineering II, One Shields Avenue, University of California, Davis, CA 95616, USA

Received 22 November 2003; received in revised form 21 February 2004; accepted 27 February 2004

Abstract

Web service infrastructures usually are comprised of front-end Web servers that accept requests and process them, and backend servers that manage data and services. Current Web servers use various API sets to access backend services. This model does not support service differentiation, overload control, caching of contents generated by backend servers. We have proposed a framework for using service brokers to facilitate these features. Service brokers are software agents that are the access points to backend services in Web servers. Unlike the current API-based scheme where accesses to backend services are through stateless and isolated APIs, in service broker framework, they are undertaken by passing messages to service brokers who gather all the requests and intelligently process them. We have prototyped this framework and validated its function in providing request clustering and service differentiation in accessing backend services. In addition, the performance in terms of the processing time is enhanced by this approach.

Keywords: HTTP; Web services; Service broker; Service differentiation; Overload control; Dynamic content caching

1. Introduction

Web servers have established their presence and usage in a variety of environment. More and more servers are being deployed for complex service environments, which also involve a variety of auxiliary servers. The platform independence and universal accessibility of Web servers have been leveraged to access other services like database,

* Corresponding author. Tel.: +1-5172829734; fax: +1-5172829734.
E-mail addresses: chenhua@cs.ucdavis.edu (H. Chen), prasant@cs.ucdavis.edu (P. Mohapatra).

1084-8045/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2004.02.001

mail, and directories. Web services like Microsoft.NET initiatives push such practice even further by facilitating more services accessibility through Web interfaces. It is conceivable that future Web servers will involve even more heterogeneous auxiliary service providers (hereafter, referred to as backend servers) to serve various tasks. Most large Web servers include a set of front-end servers that receive the requests from the clients. The requests are served by accessing a set of backend servers, which provide database, directory services, secure transactions, and other services. A schematic diagram of a typical Web server environment is shown in Fig. 1.

Backend servers can be categorized as tightly coupled or loosely coupled based on their connectivity and ownership with the initiating Web servers. Tightly coupled servers, like database and directory servers, are closely connected, usually in the same LAN, to the Web servers and belong to the same administrative authority. Tightly coupled servers are usually reliable and of high capacity. Loosely coupled servers represent Web servers belonging to other owners, which are not under control of the request initiating front-end servers. Web syndicates like My.Yahoo! and My.Netscape belong to loosely coupled Web servers. In accessing their services, the requests and response traffic must traverse WAN networks, which may incur higher latency and jitters than LANs. In more security-sensitive applications, authentication must proceed before further transactions. Since the loosely coupled servers are shared resources, service guarantee becomes an outstanding problem. We envision that in the future such services would be contract-based such that the service availability is honored only when the incoming traffic are within the contracted specifications. Loosely coupled services present a business model that has been existing in the current society. For instance, a travel agency has no sole control over airliners' ticketing services. Rather it contacts multiple airlines and selects the best deals for the customers.

The connectivity distinction between the two categories exposes different performance concerns. For tightly coupled services, the major performance issue is

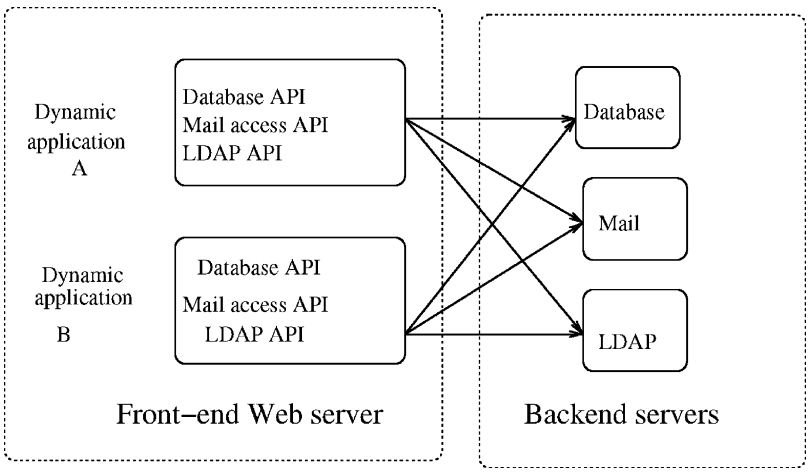

Fig. 1. API paradigm.