Adaptive Bridging with Portable Interceptor for Efficient Integration of Reflective Middleware

Hyun Ko , Seungwok Han, and Hee Yong Youn
School of Information and Communications Engineering
Sungkyunkwan University, Suwon, Korea
osthein@hanmir.com, donny@devg.org, youn@ece.skku.ac.kr

ABSTRACT

For pervasive computing paradigm we need a middleware framework efficiently integrating the pervasive objects. In this paper we propose an adaptive bridging approach where the cool spot objects and hot spot objects are assigned dynamic bridge and static bridge, respectively. Experiment on a system consisting of three different platforms reveals that the proposed approach significantly reduces the response time compared to the case of using only the static bridge or dynamic bridge.

Keywords

Adaptive bridging, agent platform, CCM (CORBA Component Model), pervasive computing, portable interceptor.

1. INTRODUCTION

In the pervasive computing environment there exist a number of objects having distinct characteristics, and thus different middleware are required for effectively manipulating them. For example, some objects need a real-time middleware framework and the others do a component-based middleware framework. One single middleware that is able to support all the objects equally efficiently is hard to develop because of the complexity and overhead of implementation. Therefore, a composite middleware, which is flexible and interoperable with each other, is a more feasible solution.

In this paper we introduce an adaptive bridging approach guaranteeing transparent and efficient connection between heterogeneous middleware. We have already developed an agent-based middleware called CALM (Component-based Autonomic Layered Middleware) [1]. The proposed bridging has been embedded in the CALM, which is a kind of MOM [2]. The proposed approach combines static bridges assigned for hot objects and dynamic bridges assigned for cold objects using DSI/DII (Dynamic Skeleton Interface/Dynamic Invocation Interface) [3]. We also apply the bridging for the connection of the component-based platform and the agent platform.

The proposed approach is evaluated using a test bed comprising of three different platforms – JADE [4], MICO, and CALM - of nine servers. The experiment reveals that the proposed adaptive bridging approach significantly reduces the response time compared to the case of using only the static bridge or dynamic bridge, especially when the number of activated objects is relatively large.

2. THE PROPOSED SCHEME

The target here is to integrate the agent platform (JADE) and CCM platform (MICO) transparently based on the CALM. JADE does not fully support CORBA 3.0. Hence we need a bridge to connect it to the CORBA-based middleware. JADE is a software framework implemented in Java language, and JAVA ORB is fully supported in JADE. If the applications on the CALM do not use CCM, JADE and CALM can communicate with each other through IIOP without a bridge. However, the agents in JADE cannot directly access the CORBA components without a bridge because JAVA ORB does not support full CCM interoperability. In the CORBA spec, the CCM supports low version compatibility using the IDL compiler without a bridge. The ORB platform vendors, who can support the low version compatibility, however, are rare yet. Generally speaking, using the low version compatibility is not feasible [5].

In the proposed scheme all the objects implemented in the server are classified into two objects, *cool spot object* and *hot spot object*, according to the request frequency measured by the portable interceptor. For the cool spot objects the requests are bursty and the frequency is not high. On the other hand, for the hot spot objects the requests are relatively persistent and the frequency is high. Therefore, we let the objects in the cool spot use dynamic bridge based on DII/DSI, while the objects in the hot spot use static bridge based on the on-demand bridging mechanism. By employing this approach we can achieve good balance between throughput and flexibility.

The portable interceptors take a significant role in the proposed scheme. They transparently route the requests of objects to the main bridge and provide the information on the request frequencies of the objects to the main bridge for adaptive judgments. They also inform the objects to use either the static bridge or dynamic bridge using the orders generated by the main bridge.

In this paper three different platforms are bridged for performance evaluation of the proposed approach. Here the CALM is the backbone middleware, JADE is the agent platform middleware, and MICO is the CCM middleware. The main bridge receives the information on the request frequency provided by the portable interceptors, and classifies the objects into two using this information; cool spot objects and hot spot objects. The threshold

This research was supported in part by the Ubiquitous Autonomic Computing and Network Project, 21st Century Frontier R&D Program in Korea and the Brain Korea 21 Project in 2005.
Corresponding author : Hee Yong Youn

used for the classification is not always same but decided empirically.

Figure 1 shows the structure of the proposed bridging. The main bridge provides the function of dynamic bridge to the cool spot objects, and all inter-platform packets are routed by the main bridge. The main bridge is an object of the CALM. It uses not only the CALM ORB but also MICO ORB and JAVA ORB. With this structure the main bridge can provide generic and dynamic bridging. Note here that dynamic bridges are not as efficient as static bridges.

The main bridge provides a static bridge to the objects moving from cool spot to hot spot. Hence, the hot spot objects use static bridges which are made by the main bridge in run-time. By using the static bridge, the objects of hot spot can get high throughput. However, all the bridges cannot get their own static bridge. Then, very large memory and computing resources will be needed. Thus, the objects of cool spot, which are bursty and have low request frequency, use the dynamic bridge.

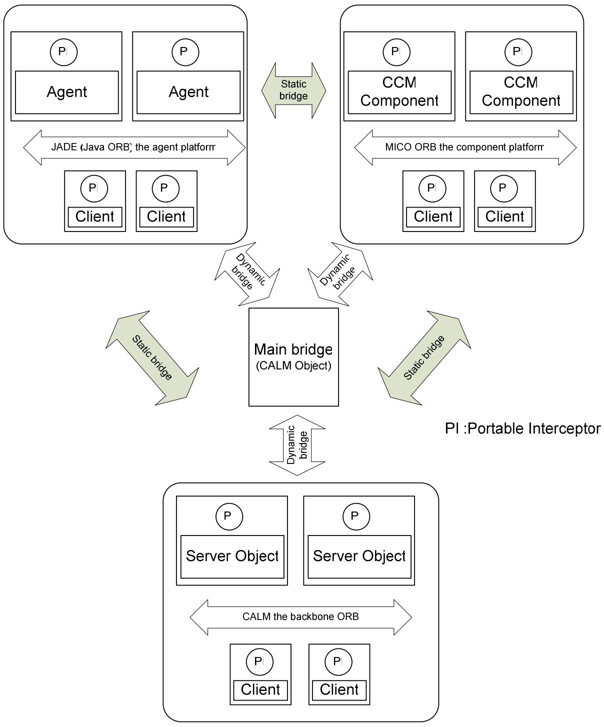

Figure 1. The structure of the proposed bridging.

Interceptor is an optional extension allowing additional services such as security inserted in the invocation path. For example, using the interceptors, we can be notified of the communication between a client and server, and modify the communications, if we wish, by effectively altering the behavior of the ORB. Here we use portable interceptors for measuring the load of each object.

The test-bed is as follows. Jade, the agent platform, comprises of two Pentium 4 (3.0 GHz, 1GB RAM) servers and one SUN (SUN SPARC dual CPU 800MHz , 2GB RAM) server. The agent objects are loaded in this platform. MICO, the CCM platform, comprises of three Pentium 4 (3.0 GHz, 1GB RAM) servers. The CCM objects are loaded in this platform. The CALM platform comprises three Pentium 4 (3.0 GHz, 1GB RAM) servers. In this platform the COS objects are loaded. The agents, CCM, and COS objects are implemented for performance evaluation. These objects are activated by the load generator for experimenting interconnection and interoperability of heterogeneous middleware.

The experiment is performed with three different scenarios. In Scenario 1, the entire objects use the dynamic bridge. In Scenario 2, the entire objects use the static bridge of their own. Here each object accesses the objects in different platform with its own stub and skeleton. In Scenario 3, the main bridge classifies the objects as cool spot objects and hot spot objects by measuring the CDF of Poisson distribution. Scenario 3 is thus the case that we propose. We vary the number of activated objects while measuring the response time of each object. The result in Figure 2 well demonstrates the tradeoff between the static and dynamic bridging.

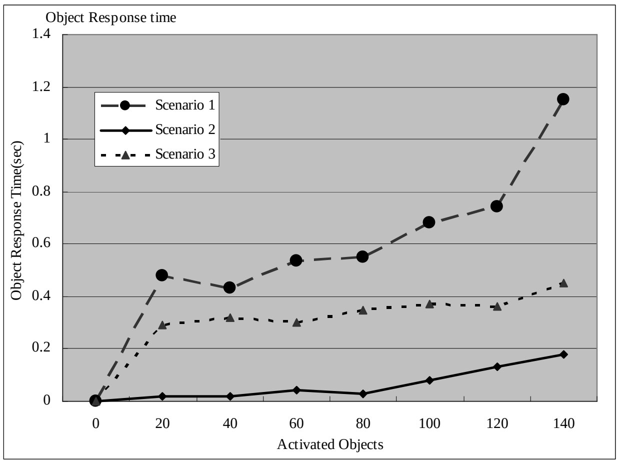

Figure 2. The comparison of response time.

In this paper the experiment was performed with the CALM, JADE, and MICO. More research will be performed with other middleware such as light-weight CORBA and real-time CORBA. We will also develop more transparent and efficient bridging techniques. A formal solution for finding the threshold value used for deciding cool and hot spot objects will also be developed, which was set empirically by experiment.

3. REFERENCES

[1] CALM (Component-based Autonomic Layered Middleware) http://mobile.skku.ac.kr, Sungkyunkwan Univ., 2005.

[2] Menasce, D.A., “MOM vs. RPC: communication models for distributed applications,” Internet Computing, IEEE Volume 9, Issue 2, March-April 2005, Page(s):90 - 93 Digital Object Identifier 10.1109/MIC.2005.42.

[3] Object Management Group. The Common Object Request Broker: Architecture and Specification. Revision 3.0.3, Object Management Group, March 2004.

[4] JADE, Java Agent Development framework web site: http://jade.cselt.it.

[5] Fatoohi, R. et. al, “Performance evaluation of middleware bridging technologies,” Int’l Symp. on Performance Analysis of Systems and Software”, April 2000, Page(s):34 - 39