Malleable Services

Venu Vasudevan
Networks and Infrastructure Department
Motorola Labs
1301 E. Algonquin Rd.
Mail Stop IL02/2240
Schaumburg, IL 60196
847-576-4182
venuv@rsch.comm.mot.com

Sean Landis
Networks and Infrastructure Department
Motorola Labs
1301 E. Algonquin Rd.
Mail Stop IL02/2240
Schaumburg, IL 60196
847-576-3469
Sean.Landis@motorola.com

Abstract

Deploying distributed services over a complex network topology presents a challenge, one of mapping the objects to locations in an optimal manner. This mapping needs to be dynamic, taking current network conditions into consideration. Remapping services is manual-intensive, requires operator effort and may result in service downtime. The Mojave project described here investigates an architecture for implementing malleable (auto-configuring) services using reactive and mobile agents. In contrast to past efforts, Mojave views agents as a wrapper technology implemented over a Jini-tuplespace based architecture. The paper describes the Mojave architecture and implementation, experiences in building an adaptive systems manager application, and benefits of the Mojave architecture for thin-client computing.

1. Introduction

While distributed object technology has accelerated the implementation of large and complex services, their deployment presents a challenge. Distributed service deployment involves mapping application components to the available machine topology. This might entail placing compute-intensive components on large machines, data-intensive components close to the data sources, and components that perform sensitive tasks on platforms that are less vulnerable to intrusions or attacks. Given the continual change in an application’s network environment, the mapping of components to locations and hardware is likely to continually change in response to changing resource conditions.

The current model of operator-mediated service mapping is laborious in terms of effort, and expensive in terms of potential service downtime. The goal in Mojave (MOBILE JINI [10] Agent enVironmEnt) is to build a component platform for auto-configuring (or malleable) services. Services that automatically and proactively configure themselves to changing conditions will experience reduced downtime while also reducing the operator overhead in service reconfiguration.

The approach taken in Mojave to implementing auto-configuration is to enable application components to be transportable and environment-aware. Components are transportable in that they can pause execution at one location and begin executing at another without loss of state. They are environment-aware in that the component code is augmented with a mobility certificate, a non-functional specification of the environment required for the component to operate effectively. The Mojave infrastructure monitors the mobility certificates of application components, and relocates the components whose certificates are violated. In aggregate, the mobility policy specifications of an application’s components automate reconfiguration that the operator would otherwise carry out manually.

We refer to Mojave components as mobile agents, since they are autonomously mobile from the point of view of the operator. However, our emphasis on mobility management as a wrapper technology for adaptive computing differs from past usage of the term for mobile objects with pre-programmed (internal) itineraries. Table 1 contrasts the use of mobile agents in Mojave from other agent projects.