Data to the People - It’s a Matter of Control

Mauricio Cortes and J. Robert Ensor

Bell Laboratories, Lucent Technologies
600 Mountain Avenue, Murray Hill, New Jersey 07974, USA
{mcortes, jre}@lucent.com

Abstract. This paper highlights some early results from our current research. We are building prototype multimedia services, e.g., digital television, video on demand, and teleconferencing, for metropolitan area networks. Our work illustrates that application-aware resource management plays a critical role in providing such services economically. The paper describes a middleware system—called Narnia—that supports the development and deployment of these application-aware resource controls. Using Narnia programming abstractions, application developers can create flexible, dynamic resource managers as part of multimedia service implementations. The paper also presents a brief summary of our experiences using Narnia to build applications.

1 Introduction

Our goal is to provide multimedia services to communities of people. More precisely, we intend to make multimedia services, e.g., digital television, video on demand, multimedia teleconferencing, and multi-party games, available through packet networks. Towards this goal, we are building a prototype data center, which is illustrated in Figure 1. It contains a switch/router complex that supports data distribution among its users’ endpoint devices, local application servers, and interfaces to external networks. Its application servers store and/or cache data as well as perform specialized manipulations. Its software manages loads on its network and computing resources.

Our experience building the data center has shown us that flexible, dynamic, application-aware controls of network and computing resource managers are necessary for economical deployment of multiple services to a large number of people. Without such controls, switches and routers can become overloaded as servers try to deliver large files or real-time data streams to users. Similarly, controls are needed to protect application servers from too many user requests. Simply adding switches and application servers to meet uncontrolled service loads would typically be too expensive. On the other hand, control software can effectively manage service loads. For example, a controller can balance the request load among a group of application servers. In addition, this controller could filter and/or aggregate requests, further reducing the load to any application server.
We have built middleware, which we call *Narnia*, to support development of application-aware controls for network and computing resources. According to the Narnia reference architecture, each resource control is built as a specialized application. As Figure 2 indicates, a typical application is built as a collection of controllers and gateways.

Controllers receive requests from users and other controllers. Therefore, they are able to enforce rules (policies) concerning an application’s behavior. Controllers call upon gateways to manage specific resources, such as network elements or media servers. Gateways interact with these resources according to the resources’ command interfaces. A collection of controllers and gateways can run on one or more physical machines, according to specified deployment configurations. Different collections can be created to form different applications; thus, these components can be reused.

## 2 System Components

Narnia provides an extensible set of abstractions to support a variety of multimedia applications. Its base abstractions are sessions, events, event handlers, and shared resources. Controllers and gateways are built upon these abstractions.