Chapter 8
The PADRES Publish/ Subscribe System

Hans-Arno Jacobsen
University of Toronto, Canada

Alex Cheung
University of Toronto, Canada

Guoli Li
University of Toronto, Canada

Balasubramaneyam Maniymaran
University of Toronto, Canada

Vinod Muthusamy
University of Toronto, Canada

Reza Sherafat Kazemzadeh
University of Toronto, Canada

ABSTRACT

This chapter introduces PADRES, the publish/subscribe model with the capability to correlate events, uniformly access data produced in the past and future, balance the traffic load among brokers, and handle network failures. The new model can filter, aggregate, correlate and project any combination of historic and future data. A flexible architecture is proposed consisting of distributed and replicated data repositories that can be provisioned in ways to tradeoff availability, storage overhead, query overhead, query delay, load distribution, parallelism, redundancy and locality. This chapter gives a detailed overview of the PADRES content-based publish/subscribe system. Several applications are presented in detail that can benefit from the content-based nature of the publish/subscribe paradigm and take advantage of its scalability and robustness features. A list of example applications are discussed that can benefit from the content-based nature of publish/subscribe paradigm and take advantage of its scalability and robustness features.

DOI: 10.4018/978-1-60566-697-6.ch008

INTRODUCTION

The publish/subscribe paradigm provides a simple and effective method for disseminating data while maintaining a clean decoupling of data sources and sinks (Cugola, 2001; Fabret, 2001; Castro, 2002;Fiege, 2002; Carzaniga, 2003; Eugster, 2003; Li, 2005; Ostrowski, 2006; Rose, 2007). This decoupling can enable the design of large, distributed, and loosely coupled systems that interoperate through simple publish and subscribe invocations. While there are many applications such as information dissemination (Liu, 2004; Nayate, 2004; Liu, 2005) based on group communication (Birman, 1999) and topic-based publish/subscribe protocols (Castro, 2002; Ostrowski, 2006), a large variety of emerging applications benefit from the expressiveness, filtering, distributed event correlation, and complex event processing capabilities of *content-based publish/subscribe systems*. These applications include RSS feed filtering (Rose, 2007), stock-market monitoring engines (Tock, 2005), system and network management and monitoring (Mukherjee, 1994; Fawcett, 1999), algorithmic trading with complex event processing (Keonig, 2007), business process management and execution (Schuler, 2001; Andrews, 2003;), business activity monitoring (Fawcett, 1999), workflow management (Cugola, 2001), and service discovery (Hu, 2008).

Typically, a distributed content-based publish/subscribe systems is built as an application-level overlay of content-based publish/subscribe brokers, with publishing data sources and subscribing data sinks connecting to the broker overlay as clients. In a content-based publish/subscribe system, message routing decisions are not based on destination IP-addresses but on the content of messages and the locations of data sinks that have expressed an interest in that content.

To make the publish/subscribe paradigm a viable solution for the above applications, additional features must be added. This includes support for *composite subscriptions* to model and detect composite events, and to enable event correlation and in-network event filtering to reduce the amount of data transferred across the network.

Furthermore, the publish/subscribe substrate that carries and delivers messages must be robust against non-uniform workloads, node failures, and network congestions. In PADRES[1], robustness is achieved by supporting alternate message routing paths, load balancing techniques to distribute load, and fault resilience techniques to react to broker failures.

It is also essential for a publish/subscribe system to provide tools to perform monitoring, deployment, and management tasks. Monitoring is required throughout the system to oversee the actual message routing, the operation of content-based brokers, and the interaction of applications via the publish/subscribe substrate. Deployment support is required to bring up large broker federations, orchestrate composite applications, support composition of services and business processes, and to conduct controlled experiments. Management support is required to inspect and control live brokers.

This chapter presents the PADRES content-based publish/subscribe system developed by the *Middleware Systems Research Group* at the University of Toronto. The PADRES system incorporates many unique features that address the above concerns and thereby enable a broad class of applications. The remainder of this chapter beings with a description of the PADRES language model, network architecture and routing protocol in Section 1. This is followed by an outline of the PADRES load balancing capabilities whereby the system can automatically relocate subscribers in order to avoid processing or routing hotspots among the network of brokers. The section then addresses failure resilience describing how the PADRES routing protocols are able to guarantee message delivery despite a configurable number of concurrent crash-stop node failures. Some of the PADRES distributed management features are presented, including topology monitoring and