DBIS-Toolkit: Adaptable Middleware For Large Scale Data Delivery

Mehmet Altinel[†], Demet Aksoy[†], Thomas Baby[†], Michael Franklin[†], William Shapiro[†] and Stan Zdonik[‡]

[†]Department of Computer Science
University of Maryland
College Park, MD 20742
{altinel,demet,thomas,franklin,billshap}@cs.umd.edu

[‡]Department of Computer Science
Brown University
Providence, RI 02912
sbz@cs.brown.edu

1 Introduction

The proliferation of the Internet and intranets, advances in wireless and satellite networks, and the availability of asymmetric, high-bandwidth links to the home, have fueled the development of a wide range of new "dissemination-based" applications. These applications involve the timely distribution of data to a large set of consumers, and include stock and sports tickers, traffic information systems, electronic personalized newspapers, and entertainment delivery. Dissemination-oriented applications have special characteristics that render traditional client-server data management approaches ineffective. These include: tremendous scale, significant overlap in user data needs, and asymmetric data flow from sources to consumers.

The mismatch between the data access characteristics of these applications and the technology used to implement them on the WWW results in scalability problems [Fran98]. For example, WWW based applications employ the HTTP protocol which uses a request-response (or client-server), unicast method of data delivery. Using request-response, each user sends requests for data to the server. The large audience for a popular event can generate huge spikes in the load at servers, resulting in long delays and overloaded servers. Compounding the situation is that users must continually *poll* the server to obtain the most current data, resulting in multiple requests for the same data items from each user. In an application such as an election result server, where the interests of a large part of the population are known *a priori*, most of these requests are unnecessary.

In order to address the needs of this new class of applications, we are developing a Dissemination-Based Information Systems (DBIS) toolkit. The toolkit serves as an adaptable middleware layer that incorporates several different data delivery mechanisms and provides an architecture for deploying them in a networked environment. The toolkit also includes facilities for performance monitoring, which can allow a system developer to examine the impact of using different data delivery mechanisms. We have implemented an initial version of this toolkit and have used it to develop a weather map dissemination application.

2 DBIS-Toolkit Overview

2.1 The DBIS Framework

The basic concepts of the DBIS framework were presented at the OOPSLA 97 conference [Fran97]. A more recent description appears in [Akso98b]. The two major features of the framework are: First, it incorporates a number of different options for data delivery, including traditional request-response, publish/subscribe, Broadcast Disks [Acha95, Acha97] and on-demand broadcast [Akso98a]. Second, it is based on the notion of *network transparency*, which allows different data delivery mechanisms to be mixed-and-matched within a single application. Network transparency is provided through the use of *Information Brokers*, which acquire information and distribute it to other consumers. Brokers are middlemen; a broker acts as a client to some number of data sources, collects and possibly repackages the data it obtains, and then functions as a data source to other nodes of the system. Along the way, brokers may add value to the information, such as integrating it with data from other sources or enhancing its organizational structure. By creating hierarchies of brokers, information delivery can be tailored to the needs of many different users.

SIGMOD '99 Philadelphia PA

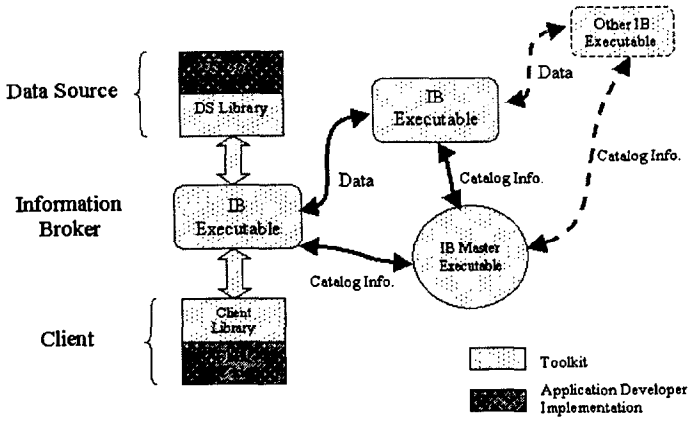

Figure 1: An Instantiation of a DBIS

2.2 Toolkit Description

The toolkit provides a set of application programming interfaces (APIs) and libraries that allow a developer to construct and experiment with a DBIS application. Figure 1 shows an example instantiation of a DBIS using the current toolkit. The DBIS-Toolkit consists of four main components (shown as lightly-shaded items in the figure):

Data Source (DS) Library - a data source wrapper that encapsulates network communication and provides conversion functions for data.

Client Library - a client program wrapper that encapsulates network communication and provides conversion functions for queries and user profiles. It also provides monitoring and filtering of broadcast or multicast channels.

Information Broker (IB) - the main component of the DBIS-Toolkit. The IB contains communication, buffering, scheduling, and catalog management components and is described in more detail below.

Information Broker Master - The IB Master is responsible for managing global catalog information about data and the topology of the DBIS. All IBs must register with the IB Master and all catalog updates must be sent to the IB Master.

In addition to these four components, the toolkit contains a flexible performance monitoring capability that can be used to graphically display real-time performance metrics such as bandwidth and CPU utilization, response times, etc. on a per-IB basis.

2.3 Data Modeling

As the focus of this project to date has been on the "plumbing" required to integrate multiple forms of data delivery at the application level, the current prototype uses a very simple data model consisting of categories and keywords within those categories. Categories and keywords are used in the specification of *queries* and *profiles*. Queries are *pull* requests that are transmitted from a client to a data source (via one or more IBs). Queries consist of a category and optional keywords. Queries are ultimately processed at a data source — all data items that match the category and at least one keyword (if specified) are sent to the client from which the query originated. In contrast, profiles are used to support *push*-based delivery. When a new data item arrives at an IB, its category and keywords are compared with the user profiles registered at that IB and the item is sent to any clients whose profile indicates an interest in the item. Thus, profiles can be viewed as a form of continually executing queries. The integration of more sophisticated data models such as (XML-based) semistructured models, and more flexible IR-style models is one aspect of our on-going development for the toolkit.

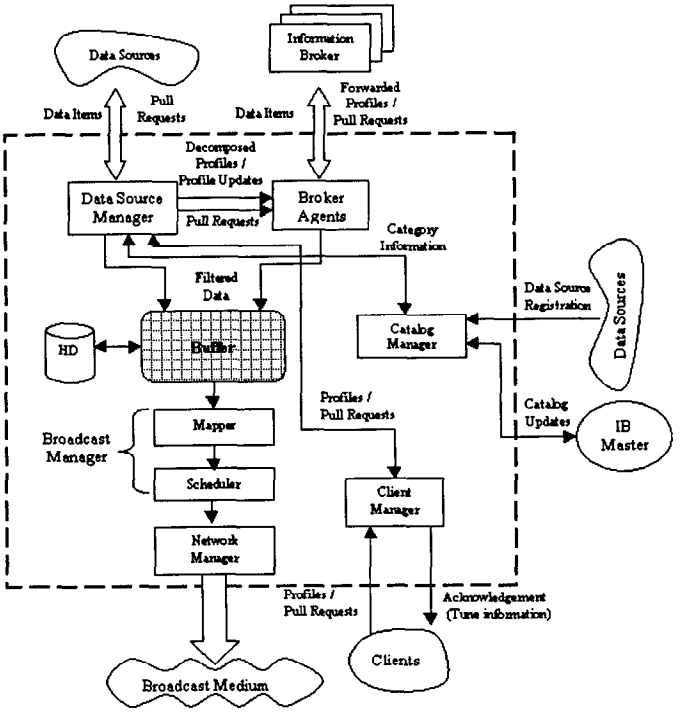

Figure 2: Information Broker (IB) Architecture

2.4 Information Broker Architecture

As stated above, the IB contains much of the functionality of the DBIS-Toolkit. The IB module (shown in Figure 2) consists of the following components:

Data Source Manager (DSM) - This component obtains (via push or pull) data items from the data sources and matches them with client pull requests or profiles.

Broker Agent (BA) - This component performs similar functions as the DSM but for sources that are actually other IBs (rather than data sources). In addition, the BA handles other IB-to-IB functions such as profile and request forwarding.

Catalog Manager - This component manages local copies of catalog information for use by the processes running at the broker. All catalog changes are sent to the IB Master, which propagates them to the catalog managers of all other IBs.[1]

Broadcast Manager - Once data have been filtered through the DSM or BA, they are passed to the Broadcast Manager, which has two main components. The *Mapper* assigns data items to one or more physical communication channels. The *Scheduler* makes decisions about the order in which data items should be placed on those channels.

Network Manager - This is the lowest level of the communication component of the IB. It sends data packets to the network according to the information provided by the broadcast manager.

Client Manager - This module handles requests that arrive from the IB's clients. It forwards them to the proper modules within the IB and maintains communication sessions with the clients.

3 A DBIS Application

An initial version of the DBIS-Toolkit has been built using Windows NT and its IP Multicast support. The toolkit has been used to create a weather map dissemination application (see Figure 3). In this application "map servers" send out updated maps of different types (i.e., radar, satellite image, etc.) for different regions of the United States. Clients can subscribe to receive updates for specific types of maps for specific regions. Users can also pose queries to obtain the most recent versions of specific maps or to zoom in on specific regions of the maps. Maps are delivered over unicast or multicast links. The application serves as a demonstration vehicle emphasizing the following unique aspects of the DBIS-Toolkit:

- The incorporation of multiple delivery mechanisms and the ways in which they are supported by the various components of the toolkit.
- The ability to make efficient use of available resources by choosing appropriate delivery mechanisms.
- The exploitation of Network Transparency through the use of multiple levels of Information Brokers.
- The ability to monitor the system dynamically using the graphical performance monitor.

[1] Future versions of the toolkit will employ more distributed catalog management to avoid the potential bottlenecks of the current centralized approach.

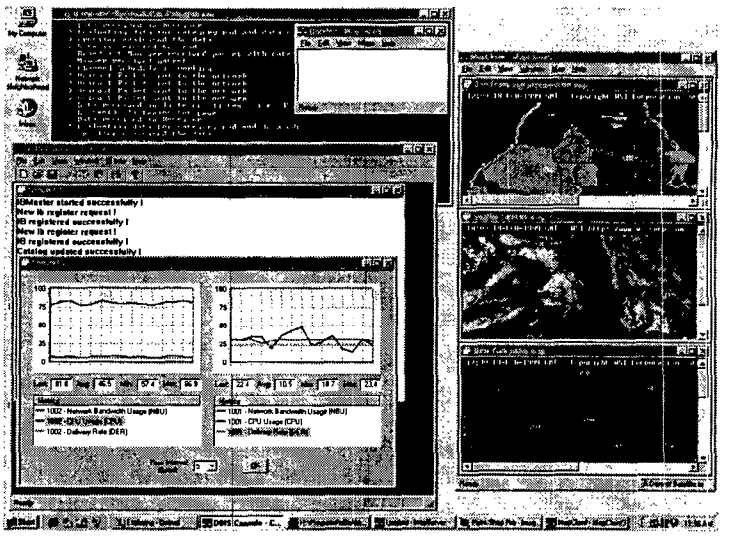

Figure 3: Example DBIS Application

Acknowledgments

The authors would like to thank Rahul Bose, Jane Wang and Rick Rhodes for their contributions to the DBIS prototype design and implementation. This research has been partially supported by Rome Labs agreement number F30602-97-2-0241 under DARPA order number F078, by the NSF under grant IRI-9501353, and by Intel, Microsoft, NEC, and Draper Laboratories.

References

[Acha95] S. Acharya, R. Alonso, M. Franklin, S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments", *Proc. ACM SIGMOD Conf.*, San Jose, CA, May, 1995.

[Acha97] S. Acharya, M. Franklin, S. Zdonik, "Balancing Push and Pull for Broadcast Data", *Proc. ACM SIGMOD Conf.*, Tucson, AZ, May, 1997.

[Akso98a] D. Aksoy, M. Franklin "Scheduling for Large-Scale On-Demand Data Broadcasting" *IEEE INFOCOM '98*, San Francisco, March, 1998.

[Akso98b] D. Aksoy, M. Altinel, R. Bose, U. Cetintemel, M. Franklin, J. Wang, S. Zdonik,"Research in Data Broadcast and Dissemination", *Proc. 1st Int'l Conf. on Advanced Multimedia Content Processing*, Osaka University, Osaka, Japan, November, 1998.

[Fran97] M. Franklin, S. Zdonik, "A Framework for Scalable Dissemination-Based Systems", Proc. ACM OOPSLA Conference, Atlanta, October, 1997.

[Fran98] M. Franklin, S. Zdonik. "Data in Your Face: Push Technology in Perspective", *Proc. ACM SIGMOD Conf.*, Seattle, WA, June, 1998.