This is the published version (version of record) of:


Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30004865

Copyright : ©2002 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
A Web-DB Model on Multicast and Anycast

Shui Yu, Wanlei Zhou, Ying Zhao, Mingjun Lan and Yang Xiang

1 School of Computing and Mathematics, Deakin University, 221 Burwood HWY, Burwood, VIC 3125, Australia {syu, wanlei, mlan, yang}@deakin.edu.au

2 International Network Center, Beijing University of Chemical Technology, 15 Beisanhuan East Road, Chaoyang District, Beijing, 100029, P. R. China zhaoy@buct.edu.cn

Abstract

Most of the current web-based database systems suffer from poor performance, complicated heterogeneity, and synchronization issues. In this paper, we propose a novel mechanism for web-based database system on multicast and anycast protocols to deal with these issues. In the model, we put a castway, a network interface for database server, between database server and Web server. Castway deals with the multicast and anycast requests and responses. We propose a requirement-based server selection algorithm and an atomic multicast update algorithm for data queries and synchronizations. The model is independent from the Internet environment, it can synchronise the databases efficiently and automatically. Furthermore, the model can reduce the possibility of transaction deadlocks.

1. Introduction

Increasing application requirements, such as performance, accuracy, reliability, security, etc., are demanded with the explosive development of the Internet. A Web-based database is an important and widely used application on the Internet, for example, E-commerce, E-banking, etc. Because of the huge number of users, most of the popular Web-based database systems suffer from poor performance, complicated heterogeneity, and synchronization issues. Distributed or replicated database servers can reduce the problem in some degree, but there are still a number of outstanding issues for distributed or replicated database systems, such as performance, data synchronization, heterogeneous database platforms integration, and so on.

According to the demands of Internet applications, researchers and industry players are working hard to meet the ever increasing and changing requirements. Multicast and anycast services are the results of this kind of developments.

Multicast is a service in networks trying to transmit a packet to a group of hosts. It is an important service for improving the efficiency and robustness of distributed systems and applications. Internet multicast communication services have been widely recognized as a very useful tool for many Internet applications, such as audio and video conference, replicated database updating, resource discovery, and so on.

Anycast is a service providing a stateless best effort delivery of an anycast datagram to at least one host, and preferably only one host, which serves the anycast address. It tries to find the “best” server among the replicated or mirrored servers on the Internet.

In this paper, we proposed a novel Web-based database model by combining multicast and anycast protocols. The model is independent from the heterogeneous database platforms; it can synchronise the databases efficiently and automatically. It also improves query performance based on the advantages of multicast and anycast protocols.

The rest of the paper is organized as follows. Section 2 discusses the related work about anycast and multicast application in database. In section 3, we propose a novel model for Web-based database on multicast and anycast protocols. The algorithms applied for the proposed model are presented in section 4. Finally, in section 5, summary and future work are described.

2. Related Works

Multicast [4] capability has been recognized as an important facility for networks and the Internet because of its growing use in distributed systems. [7] presented 4 protocols for distributed replicated databases that take advantage of atomic broadcast systems to simplify message passing and conflict resolution in hopes of making replication efficient

1. Broadcast all protocol. This protocol requires that a transaction initiated at a server site broadcasts all its operations (read or write) to all other sites using atomic broadcast.

2. Broadcast writes protocol. This protocol executes read operations locally and broadcast only write operations to the other sites.

3. Delayed broadcast writes protocol. This protocol attempts to completely localize transaction execution. The protocol defers update operations until commit is ready, when a single message with all updates is sent to all other sites.

4. Single broadcast transactions protocol. This protocol maintains a version number with each page in the database to make sure the correct sequence of read in databases.
These protocols can be applied to replicated database recovery as well [8], and the work shows that multicast service is a good solution for the data synchronization and data recovery issues for distributed systems.

The original work by Partridge, Mendez, and Milliken [13] proposed the idea of anycast for the IP next generation, and discussed its network layer support. A number of research is quickly conducted in the area from then on, and researchers have archived some results [1][9][11][15] in the network layer. At the middle of 1990s, some researchers found the limitations of network-layer anycast, for example, inflexibility and limited supported by current routers, hence, they presented the idea of application-layer anycast [2][3][5], focusing the research on anycast in the application layer. The application-layer anycast is compatible with the nature of current Internet facilities and suites for current application requirements too. Some anycasting routing algorithms [3][12][14][16] have been proposed.

[9] proposed an idea of integrating multicast service and anycast service for the Internet service: a group of replicated (or mirrored) servers that provide anycast services may also provide multicast services and need multicast to archive consistent update operations, whereas anycast routing may help multicast requests to reach the ‘nearest’ member in a multicast group. [10] proposed a novel efficient mobile multicast protocol (MMP in short), taking advantage of the anycast routing technology.

### 3. Architecture of Web-DB Model on Multicast and Anycast

From the work in section 2, we find that combining multicast and anycast can provide a bi-directional service. Each of them provides a “one-to-many” service, and the difference is the directions. Based on the previous knowledge, we propose a Web-based database (Web-DB) model on multicast and anycast protocols. In our model, there are two groups, multicast group and anycast group for a group of replicated Web-based databases, and the members of multicast group and that of anycast group are the same. The proposed model can deal with heterogeneous database platforms, reduce transaction deadlocks, and improve the whole system performance.

#### 3.1 The Architecture of Web-DB Model on Multicast and Anycast

The architecture of the Web-DB model on multicast and anycast is shown in Figure 1. In Figure 1, the castway is a component between the Web server and the database server. Its responsibility is to synchronize data among all the replicated databases on the Internet. Anycast resolver is an independent router for all the anycast groups in the Internet, and there are many this type of routers all over the Internet. Here, the anycast resolvers accept the anycast queries from clients and feedback the “best” Web-based database among the anycast space.

The processing of a query from a client is described below.

1. A client issues a database transaction with an anycast address to an anycast resolver.
2. The Resolver then decides which database server is the “best” at that moment by using the requirement-based server selection algorithm, which will be presented in section 4.1.
3. A connection between the client and the “best” server is created.
4. If the database transaction includes read statement(s) only, then the castway does nothing. This is a normal Web-DB process, and the result will feed back to the client through CGI component, or other means.
5. If the database transaction includes update request(s), then the castway will propagate the transaction to other replicated Web-based databases using the atomic multicast update algorithm, which will be discussed in section 4.2.
6. The replicated Web-based databases receive the transaction and process it using the atomic multicast update algorithm.
7. After the executions, all the replicated Web-based databases are synchronised.

The architecture is scalable and flexible because every component is independent. The model can work on heterogeneous environment as well, because the message exchanging in the system based on two Internet standards, multicast and anycast protocols.

![Figure 1. Architecture of web-based database system on multicast and anycast](image)

The key components of the Web-DB model on multicast and anycast are the anycast resolver and the castway, which will be discussed in section 3.2.

#### 3.2 The Structure and Mechanism of Castway

We propose the castway as a kernel component for automatic and efficient data synchronization among a replicated server group. It combines the advantages of multicast and anycast protocols.

#### 3.2.1 The Structure of Castway

Castway is an important component in our proposed model. Its functionality is to synchronize all the members in the replicated servers. The structure of castway and its location in the system are shown in Figure 2.

Castway is located between the database server and Web server. Castway includes two modules, multicast...
module and anycast module. The two modules cooperate to synchronize the data among the replicated servers. As shown in Figure 2, every site in the replicated group has a castway.

We refer the first transaction among the replicated group as an original transaction, and the same transactions executed in other servers in the group as copy transactions. There are two kinds of processes in castway for data synchronization:

- An original transaction initialised in a site
- A copy transaction reaches at a site

![Figure 2. The structure of castway.](image)

### 3.2.2 Processing An Original Transaction

The processing of an original transaction at one site is presented below,

1. The castway checks the coming IP address, if it is a unicast address, then the transaction is an original transaction.
2. If the transaction is a read only transaction, then castway does nothing, besides passing the transaction directly to the backend database server.
3. If the transaction is an update transaction, castway will use the atomic multicast update algorithm to synchronize data among replicated databases.

### 3.2.3 Processing A Copy Transaction

The processing of a copy transaction at one site is described as following,

1. The castway checks the coming IP address, if it is a multicast address, then the transaction is a copy transaction.
2. The castway uses the atomic multicast update algorithm to execute the transaction locally.
3. After the locally execution, the local data is synchronized with the original data source.

### 4. Algorithms for Web-DB Model on Multicast & Anycast

Based on the advantages of multicast and anycast protocols, we present two protocols here: The requirement-based server selection algorithm for Internet clients to locate the “best” server, and an atomic multicast update algorithm for data synchronization among replicated databases.

#### 4.1 The Requirement-based Server Selection Algorithm

When an Internet user issues an anycast query, there is an anycast resolver trying to tell the client which server among the anycast group is the “best” one for the query. In this paper, we present an algorithm, called requirement-based server selection algorithm to deal with this issue. The process of the requirement-based server selection algorithm is described below.

1. An Internet user issues an anycast query with an anycast address A.
2. The query reaches an anycast resolver.
3. The anycast resolver sends probing packets to every member in the anycast group.
4. The anycast resolver takes the first reply representing the best network and server performance, and responses to the client with the routing path, and the unicast IP address of the “best” server. The anycast resolver discards the rest replies from other replicated database servers.
5. The client creates the connection according to the received information using traditional unicast methodologies.

In this case, the probed replicated database servers should respond to the probing requirements, such as ping, respectively. If a server’s load is heavy or performance is low, then the response must last longer than a server whose load is light or performance is better. At the same time, the replies to the resolver include the information of network performance as well. More detailed description of this algorithm is described in [16].

#### 4.2 Atomic Multicast Update Algorithm

A transaction is a sequence of read and write operations on the data items that is executed atomically. For atomic multicast described in this paper, we consider it satisfying the following properties [6]:

1. If a site broadcasts a message \( m \), the primitive ensures that the message will be delivered to all operational sites.
2. If a site delivers message \( m \), then all operational sites deliver \( m \).
3. If sites \( p \) and \( q \) deliver broadcast messages \( m \) and \( m' \), then \( m \) and \( m' \) are delivered in the same order at all sites.

The detail of the atomic multicast update algorithm is described below.

1. An update transaction is initialised at one site.
2. Castway passes the statements to the local backend database, at the same time, the castway copies all the statements except the commit statement, and multicasts the statements to the multicast group.
3. The copy sites execute the statements of copy transactions.
4. For both of the original site and the copy sites, a transaction, \( T_i \) tries to execute a read locally; if the lock manager cannot issue a lock for the operation, such as the...
related page(s) is/are locked by another transaction, then the anycast module will submit the read operation to the rest of the members using the anycast mechanism to get the related page(s) efficiently.

5. When the original transaction is committed then the multicast module will issue the commit demand to all members to the rest of the replicated group.

6. After that, all the members in the multicast group are synchronized.

The proposed algorithm can reduce the possibility of transaction deadlock, because the algorithm uses all the readable resource among replicated databases. Besides, the algorithm can prevent the possibility of long transactions, because it is not necessary to wait for an unlock of local resource.

5. Summary and the Future Work

Multicast and anycast are two useful services in the Internet, in this paper, we propose a novel Web-based database model by cooperating multicast and anycast protocols. Two algorithms, the requirement-based server selection algorithm and the atomic multicast update algorithm, are proposed for data queries and synchronizations. Based on our knowledge, this is the first paper applying multicast and anycast together for Web-based database systems.

The proposed model has several advantages compared with the current Web based database architectures:
1. Independent from the Internet environment. Multicast and anycast are international standards, therefore, the proposed model can work among heterogeneous environments, such as operating systems, database platforms, etc.
2. Server load balance and network load balance. Take the advantages of anycast, our model can provide server load balance and network load balance.
3. Efficient and automatic method of data synchronization. The proposed algorithm of atomic multicast update, can ensure the serialization of transactions, and this helps the data synchronization among the replicated database servers automatically and safely.
4. Efficient database operations. An anycast service extends the range of read operations, and helps to reduce transaction deadlock and prevents the possibility of long transactions, therefore it can improve the performance of database operations.
5. A practical model and easy for implementation. All original components of a system can be kept unchanged; only a castway component is inserted to the existing systems.

Our next step is to simulate the model and try to obtain further confirmations about the advantages of our Web-based database model on multicast and anycast.

References: