
Copyright : 2004, IEEE
A Novel Middleware Based Web Database Model

Shui Yu and Wanlei Zhou
School of IT, Deakin University, Burwood, VIC 3125, Australia
{syu, wanlei}@deakin.edu.au

Abstract

In this paper, we propose a novel model for web-based database systems based on the multicast and anycast protocols. In the model, we design a middleware, castway, which locates between the database server and the Web server. Every castway in a distributed system operates as a multicast node and an anycast node independently, respectively. The proposed mechanism can balance the workload among the distributed database servers, and offers the “best” server to serve for a query. Three algorithms are employed for the model: the requirement-based probing algorithm for anycast routing, the atomic multicast update algorithm for database synchronization, and the job deviation algorithm for system workload balance. The simulations and experiments show that the proposed model works very well.

1. Introduction

The Web-based database is an important and widely used application on the Internet, such as, E-commerce, E-banking, etc. Because of the huge number of users, most of the popular Web-based database systems suffer from poor performance, network congestion, heterogeneity, and so on. The distributed database model is a solution, but we must pay for it: replication adds great complexity to the system development [7] [15], moreover, replication jeopardizes data consistency. Maintaining the data consistency is very expensive [4].

In order to meet the demands of Internet applications, researchers and industry players are working hard to develop new protocols, Internet services, and applications for the ever increasing and changing requirements. The multicast and anycast services are the results of these kinds of developments.

Multicast [5] has been widely employed in the Internet, such as, database synchronization, replicated database updating, audio and video conference, and so on. Anycast [16] is an Internet service proposed in IPv6, which provides a stateless best effort delivery of an anycast datagram to at least one host out of the n mirrored servers, and preferably only one host, which serves the anycast address. The anycast service tries to find the “best” server among the replicated or mirrored servers of the anycast group. The anycast services are powerful in information retrieval.

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

The rest of the paper is organized as follows. Section 2 discusses the related work about anycast and multicast applications in databases. In section 3, we propose the novel model for Web-based databases based on the multicast and anycast protocols and the related algorithms. Performance evaluations are conducted in Section 4. Finally, in Section 5, summary and future work are presented.

2. Related work

The capability of the multicast service has been recognized as an important facility for networks and the Internet because of its growing usage in distributed systems [5]. Paper [9] presented four protocols for distributed replicated databases that take advantage of atomic broadcast system to simplify message passing and conflict resolution in hopes of making replication efficient. These protocols can be applied to replicated database recovery as well. All the work, such as [5] [9] [10], show that the multicast service is a good solution for the data synchronization and data recovery for distributed databases.

[16] proposed the idea of anycast for the IP next generation, and discussed its network layer support. A number of research were quickly conducted in the area from then on, and researchers have achieved some results [1] [11] [13] [18] in the network layer anycast, and also some results [2] [3] [6] in the application layer anycast. Some anycast routing algorithms [3] [14] [17] [19] have been proposed for both of the two layers. Paper [11] [12] integrated the...
multicast and anycast protocols to provide new Internet services.

3. A new Web-based database model

We obtained that the combination of the multicast and anycast services can provide a bi-directional service. The multicast provides a “one-to-many” delivery service, and it can guarantee the delivery. This characteristic can be used in data synchronization for Web-based distributed databases; meanwhile, the anycast offers a service which can find the “best” server out of a group. This feature meets the request of finding the “best” database server among the distributed group in terms of workload, performance, bandwidth, and so on. Therefore, we propose a Web-based database (Web-DB) model on the multicast and anycast protocols. In our model, there are two groups: a multicast group \( G_m = \{m_1, m_2, m_3, ..., m_k\} \), and an anycast group \( G_a = \{a_1, a_2, a_3, ..., a_j\} \), where \( m_j (j = 1, 2, 3, ..., k) \) represents a mirrored database in the Internet, and \( a_j (j = 1, 2, 3, ..., j) \) does the same.

Furthermore, we let \( m_i = a_i (j = 1, 2, 3, ..., k, k = i) \) for a group of mirrored Web-based databases.

The architecture of the Web-DB model is shown in Figure 1. We keep the traditional three-layer model of the Web-based databases, and locate a middleware, castway, between the Web server and the backend database server at each site of a distributed database system. As a middleware, the castway is independent from the database platforms, the operating systems and the Web server platforms. It integrates the functionalities of the multicast and anycast protocols to serve for distributed databases.

Once a client initiates a database query to a distributed database system, the destination address of the request will be initially set as the anycast address, which is related to the distributed databases; an application layer anycast routing algorithm will send the client an unicast address of the “best” database server among the server group, and the destination address of the request will be replaced with the unicast address at the client’s site; and then the updated database request will be forwarded to the “best” database server using the traditional IPv4 routing protocols.

We make some definitions here in order to make the following descriptions clear. An original transaction means that a transaction is received by a database server in the distributed database group for the first time. The site which processes the original transaction is an original site. While a node receives an original transaction, it will copy the transaction and broadcasts the copies to the multicast group, these transactions are called copy transactions. The sites serve for the copy transactions are called copy sites. Copy transactions are used to synchronize the distributed databases.

When a database request arrives at an original site, the castway catches the request, and strips the transaction from the packets and submits it to the backend database engine to execute. At the same time, the castway assembles the statements of the transaction except the commit statement in a new packet addressing with the multicast address and multicasts the packets to synchronize the distributed databases. The castway will forward the result from the backend engine to the client where the request comes from.

The castway examines each of the received multicast packets. If the source address of the multicast packet is the same to the current node’s own address, then the castway discards the packet; if the two addresses are different, that means the received packets are used for database synchronization only, then the castway forwards the transaction to the backend database, and discards the result which is feed back by the backend engine.

A database request can be initiated anywhere from the Internet, the anycast mechanism inherently balance the workload among the mirrored databases, and it also choose the “best” server to serve the clients according to the given criteria. The multicast protocol guarantees the delivery of the copy transactions, which are employed to synchronize the databases. The two services are implemented in the middleware, the castway, which is a critical component of the proposed model. The original CGI layer in the three layer architecture is replaced by the castway as a middleware. The castway includes three modules: an interface module, a multicast module and an anycast module, shown as Figure 1.

The interface module connects the castway to the Web server. It receives the requests, and checks every incoming request to find that weather it is an original transaction or a copy transaction. If the incoming transaction is an original transaction, then the interface module examines the transaction and makes sure whether it is a read only transaction (includes select operations only) or an update transaction (includes update, delete or insert operations). If it is a read only transaction, the request will be forwarded to the database engine by the multicast module without
multicasting the transaction; however, if it is an update transaction, the request will be delivered to the backend database engine, and furthermore, the transaction will be submitted to the group $G^*$ using multicasting. For both of the cases at an original site, the database engine submits the result of transaction back to the interface module, it is then delivered to the client through the Internet. If the incoming transaction was a copy transaction, the original site discards the transaction, because it has been executed; the copy sites forward the transaction to the local database engine, and the interface module discards the result, which comes from the backend engine. Table 1 summaries the actions in the castway. We must notice that the interface module forwards the incoming transactions to the anycast module only when it considers that the current server is overloaded.

<table>
<thead>
<tr>
<th>Castway Modules</th>
<th>Original Transaction</th>
<th>Copy Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Module (IM)</td>
<td>Forward to MM</td>
<td>Submit Result to client</td>
</tr>
<tr>
<td>Multicast Module (MM)</td>
<td>Forward to Database Engine</td>
<td>Broadcast Multicast Packets</td>
</tr>
<tr>
<td>Anycast Module (AM)</td>
<td>Deviate</td>
<td>Deviate</td>
</tr>
</tbody>
</table>

The multicast module takes the responsibility of broadcasting the original transactions to the multicast group, and forwards the transactions to the backend database engine. This module works independently as a multicasting node in the Internet. All the multicasting modules in the distributed database system use the same algorithm, and it can be choose from spanning tree, reverse path forwarding (RPF), core-based tree (CBT), et al.

The anycast module can be considered as an independent anycast node in the Internet. All nodes of the anycast group $G$, work together to offer the anycast routing service. The anycast group can take any existing anycast routing algorithms, such as the Shortest-Shortest Path Method (SSP), the Minimum Distance Method (MIN-D) [17], the Requirement-based Probing Algorithm (RPA) [19], and so on. Once the interface module finds the workload of the local node is heavy, it will forward the incoming transactions to the anycast module, rather than the multicast module, and then a job deviation algorithm will be employed by the local anycast module to find another site to serve for the overloaded transaction(s).

Three algorithms are employed in the proposed model: 1) the Requirement-based Probing anycast algorithm for anycast routing. The details of this algorithm can be found in [19]; 2) the atomic multicast update algorithm for data synchronization. The details are included in [20]; and 3) the anycast job deviation algorithms (the best neighbor, the best node, and the random selection algorithms) for load balance issue among the anycast group servers [21].

4. Performance evaluation

In order to evaluate the performance of the proposed anycast routing algorithm, we take common Web transaction as a benchmark. The common Web transaction is defined as following: for an anycast service, the common Web transaction chooses one of the distributed database server randomly; for the multicast service, the common Web transaction builds connections to each server of the multicast group, and delivers the packet to the servers independently and respectively. The result of the simulation using ns 2 is shown in Figure 2. We found that the performance of the anycast method is better than that of the common Web transaction method in most of the individual transactions; from the system viewpoint, the performance of the anycast method is better than that of the common Web transaction method.

![Figure 2 Performance difference between the common Web transaction and anycast methods](image1)

In order to measure the synchronization performance of distributed database systems, the Replication Transaction Time (RTT) is defined as follow: To a group of replicated servers $\{S_1, S_2, ..., S_i, ..., S_n\}$, the time last for each replication is denoted as $T_i$, then the replication transaction time for the distributed databases synchronization is $\text{Max}\{T_i\}, i = 1, 2, ..., n$.

![Figure 3 Performance comparison of the common Web transaction and the multicast replications](image2)
In another simulation, we varied the number of replica servers and examined the RTT of the multicast methods and the common Web transaction method. For the multicast method, we used two multicast algorithms: Centralized Multicast (CM) and Dense Mode (DM). The result is shown in Figure 4. Here COM stands for the common Web transaction method. The result shows that the multicast methods are steady and efficient.

Figure 4 Comparison of the multicast and the common replications with different number of replicas

Figure 5 shows the result of a simulation on network delays of the anycast job deviation algorithms in comparing with the central control algorithm as a benchmark. It is obvious that the best neighbor deviation algorithm is the best. More details about the anycast job deviation can be found in [20] for the interested readers.

Figure 5 Network delays in job deviation algorithms

5. Summary and the further work

In this paper, we proposed a novel Web-db model based on the anycast and multicast protocols. The model has the inherent advantages from the two protocols, such as platform independency, efficient performance, and so on. Three of our proposed algorithms are employed in the model, and the simulation and experiments show that the model works well.

The near future, we will implement a prototype of the model and evaluate its performance, independency and the other features in the real application environment.

References


