Efficient Replication Management Techniques for Mobile Databases

Ziyad Tariq Abdul-Mehdi
Multimedia University, Malaysia

Ali Bin Mamat
Universiti Putra Malaysia, Malaysia

Hamidah Ibrahim
Universiti Putra Malaysia, Malaysia

Mustafa M. Dirs
College University Technology Tun Hussein Onn, Malaysia

INTRODUCTION

Mobile databases permeate everywhere into today's computing and communication environment. One envisions application infrastructures that will increasingly rely on mobile technology. Current mobility applications tend to have a large central server and use mobile platforms only as caching devices. We want to elevate the role of mobile computers to first class entities in the sense that they will allow the mobile user to work/update capabilities independent of a central server. In such an environment, several mobile computers may collectively form the entire distributed system of interest. These mobile computers may communicate to each other in an ad hoc manner by communicating through networks that are formed on demand. Such communication may occur through wired (fixed) or wireless (ad hoc) networks. At any given time, a subset of the computer collection may connect and would require reliable and dependable access to relevant data of interest. Peer-to-peer (P2P) computing, basically, is an ad hoc network and it can be built on the fixed or along a wireless network. With P2P, computers can communicate directly and share both data and resources. So far, many applications such as ICQ (where users exchange personal messages), similar to Napster and Freenet (where users exchange music files), have taken the advantage of P2P technology. However, data management is an outstanding issue and leads directly to the problem of low data availability. Data availability is the central issue in P2P data management. The most important characteristic that affects data availability in P2P environment is the nature of the network. In the case of an ad hoc network, hosts are connected to the network only temporarily. Furthermore, hosts play the role of router, and they communicate with each other directly without any dedicated hosts. If there are no dedicated hosts that act as a router, obviously the network connections are prone to get disconnected and/or become unreliable. Thus, it is difficult to guarantee one-copy "serializability," since one relies on the mobile hosts, not the fixed hosts, in order to communicate with other hosts not reachable directly (Faiz & Zaslavsky, 1995). When hosts disconnect more often, due to the applications that have high transaction rates, the deadlock and reconciliation rate will experience a cubic growth (Faiz & Zaslavsky, 1995) and, the database is in an inconsistent state and there is no obvious way to repair this problem or allow for this eventuality. In the case of fixed network, the network connection is relatively stable, but the availability of sufficient computing resource depends on the strategies of replication.

Walborn and Chrysanthis (1997) describe the use of mobile computers in the trucking industry. Each truck has a computer with a satellite/ radio link—this is able to interact with the corporate database. Other applications include involving avoiding remote or disaster areas and for military applications with mobile computers forming ad hoc networks without communications and/or with stationary computers. Faiz and Zaslavsky (1995) discuss the impact of wireless technologies and mobile hosts on a variety of replication strategies. Distributed replicated file systems such as Ficus and Coda (Reiher, Heidemann, Ratner, Skinner, & Popek, 1994) have extensive experience with disconnected operations.

In this article, we consider the distributed database that can make up mobile nodes as well as peer-to-peer concepts. These nodes and peers may be replicated both for fault tolerance (dependability), and to compensate for nodes that are currently disconnected. Thus we have a distributed replicated database, where several sites must participate in the synchronization of a transaction. The capabilities of the distributed replicated database are extended to allow mobile nodes to plan for a disconnection, with the capability of update, and for the database—on behalf of mobile node by using fixed proxy server—to make these updates during the mobile disconnection. Once a mobile reconnects, it automatically synchronizes and integrates into database.

By using the notion of planned disconnection (*sign-off and check-out modes*), we present a framework, which allows the replicated data of mobile nodes to be available to access and update for low costs in reading and writing.

This article is organized as follows; in the second section, we review the *read one write all* (ROWA) technique; in the third section, the model of the *diagonal replication on grid* (DRG) technique is presented, and we also present an algorithm to allow disconnection nodes to update using a sign-off and check–out idea adapted in the system. In the fourth section, the correctness and the performance of the proposed technique is analyzed in terms of *communication cost and availability* comparing ROWA techniques and DRG techniques. In the final section, the conclusion is given.

VIEW OF ROWA STRUCTURE TECHNIQUE

The simplest technique to maintain replicated data is when a read-only operation is allowed to read any copy, and write operation is required to write all copies. This is called a read one write all (ROWA) protocol. This protocol only works correctly when a transaction process from one correct state to another correct state is carried out. The ROWA has the lowest read cost because only one replica is accessed by a read operation. The weakness of this method is the low write availability, because a write operation cannot be done in a failure of any replica.

The available copies technique proposed by Bernstein, Hadzilacos, and Goodman (1987) is an enhanced version of the ROWA approach in terms of the availability of write operations. Every read is translated into a "read" of any replica of the data object and every "write" is translated into write of all available copies of that data object. This technique can handle each site either when it is operational or down—and that all operational sites can communicate with them. If a site does not respond to a message within the timeout period, then it is assumed to be down. However, writing is very expensive when all copies are available: forcing read-write transactions to write all replicas.

Lazy replication protocol does not attempt to perform the write operation on all copies of the data object within the context of the transaction that updates that data object. Instead, it performs the update on one or more copies of the data object and later propagates the changes to all the other copies in all other sites. A lazy replication scheme can be characterized using four basic parameters(Bernstein, Hadzilacos, & Goodman, 1987; Borowski, 1996; Goldring, 1995; Ozsu & Valduriez, 1999). The ownership parameter defines the permissions for updating copies. If a copy is updateable it is called primary copy, otherwise it is called a secondary copy. The site that stores the primary copy of a data object is called a master for this data object, while the sites that store its secondary copies are called slaves. The propagation parameter defines when the updates copy must be propagated towards the sites storing the other copies of the same data object. Generally, lazy replication protocols can be classified into two groups. (1) The first group consists of lazy replication methods where all copies are updateable. In this case, there is group ownership on the copies. A conflict happens if two or more sites update the same replica. There are several policies for conflict detection and resolution that can be based on timestamp ordering, node priority, and others (Buretta, 1997; Helal, Heddaya, & Bhargava, 1996). The problem with conflict resolution is that during a certain period of time the database may be in an inconsistent state. (2) The second group consists of protocols where there is a single master that is updated, and each time a query is submitted for execution, the secondary copies that are read by the query are refreshed by executing all received refresh transaction. Therefore, a delay may be caused by that query.

DATABASE MODEL

In considering this environment, it has two types of networks; that is, a fixed network or an ad hoc network. For the fixed network, all sites are logically organized in the form of two-dimensional grid structure (i.e., 5*5) and in a nodes (ad hoc network), this consists of N nodes labeled $N_1, N_2, \ldots, N_n$ (as shown in Figure 1). The data will only be partially or fully replicated in that data items are stored redundantly at multiple sites. Information about the location of all copies of a data item may be stored at each site or kept in directories at several of the sites. Users interact with the database by invoking transactions at any one of the database sites. A

Figure 1. Database model