E

Ensuring Serializability for Mobile–Client Data Caching

Shin Parker
University of Nebraska at Omaha, USA

Zhengxin Chen
University of Nebraska at Omaha, USA

IMPORTANCE OF ENSURING SERIALIZABILITY IN MOBILE ENVIRONMENTS

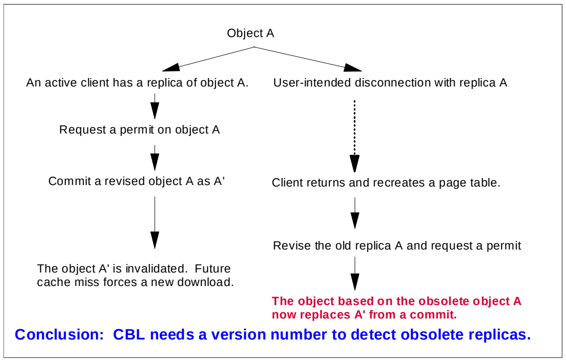
Data management in mobile computing has emerged as a major research area, and it has found many applications. This research has produced interesting results in areas such as data dissemination over limited bandwidth channels, location-dependent querying of data, and advanced interfaces for mobile computers (Barbara, 1999). However, handling multimedia objects in mobile environments faces numerous challenges. Traditional methods developed for transaction processing (Silberschatz, Korth & Sudarshan, 2001) such as concurrency control and recovery mechanisms may no longer work correctly in mobile environments. To illustrate the important aspects that need to be considered and provide a solution for these important yet "tricky" issues in this article, we focus on an important topic of data management in mobile computing, which is concerned with how to ensure serializability for mobile-client data caching. New solutions are needed in dealing with caching multimedia data for mobile clients, for example, a cooperative cache architecture was proposed in Lau, Kumar, and Vankatesh (2002). The particular aspect considered in this article is that when managing a large number of multimedia objects within mobile client-server computing environments, there may be multiple physical copies of the same data object in client caches with the server as the primary owner of all data objects. Invalid-access prevention policy protocols developed in traditional DBMS environment will not work correctly in the new environment, thus, have to be extended to ensure that the serializability involving data updates is achieved in mobile environments. The research by Parker and Chen (2004) performed the analysis, proposed three extended protocols, and conducted experimental studies under the invalid-access prevention policy in mobile environments to meet the serializability requirement in a mobile client/server environment that deals with multimedia objects. These three protocols, referred to as extended server-based two-phase locking (ES2PL), extended call back locking (ECBL), and extended optimistic two-phase locking (EO2PL) protocols, have included additional attributes to ensure multimedia object serializability in mobile client/server computing environments. In this article, we examine this issue, present key ideas behind the solution, and discuss related issues in a broader context.

BACKGROUND

In a typical client-server computing architecture, there may exist multiple physical copies of the same data object at the same time in the network with the server as the primary owner of all data objects. The existence of multiple copies of the same multimedia object in client caches is possible when there is no data conflict in the network. In managing multiple clients' concurrent read/write operations on a multimedia object, no transactions that accessed the old version should be allowed to commit. This is the basis of the invalid-access prevention policy, from which several protocols have been proposed. The purpose of these protocols is to create an illusion of a single, logical, multimedia data object in the face of multiple physical copies in the client/server network when a data conflict situation arises. When the server becomes aware of a network-wide data conflict, it initiates a cache consistency request to remote clients on behalf of the transaction that caused the data conflict. Three well-known invalid-access prevention protocols are Server-based Two-Phase Locking (S2PL), Call-Back Locking (CBL), and Optimistic Two-Phase Locking (O2PL).

In order to extend these policies to the mobile environment, we should understand that there are four key constraints of mobility which forced the development of specialized techniques, namely, unpredictable variation in network quality, lowered trust and robustness of mobile elements, limitations on local resources imposed by weight and size constraints, and concern for battery power consumption (Satyanarayanan, 1996). The inherent limitations of mobile computing systems present a challenge to the traditional problems of database management, especially when the client/server communication is

Figure 1. CBL failure analysis tree in a mobile environment



unexpectedly severed from the client site. The standard policy does not enforce the serializability to the mobile computing environment. Transactions executing under an avoidance-based scheme must obey the Read-Once Write-All (ROWA) principle, which guarantees the correctness of the data from the client cache under the CBL or the O2PL protocol. The standard CBL and O2PL protocols cannot guarantee the currency of the mobile clients' cache copies or prevent serializability violations when they reconnect to the network. Figures 1 illustrates how error conditions (appearing toward the end of the figure) arise after mobile clients properly exit the client application when the traditional CBL protocol is used.

FUNDAMENTAL ISSUES AND APPROACHES TO DEALING WITH THESE ISSUES

In order to extend invalid-access prevention policy protocols to mobile environments, there are three fundamental issues that need to be addressed for mobile-client multimedia data caching, namely:

- to transform multimedia objects from databases' persistent data type to the clients' persistent data type;
- to handle client-server communication for multimedia objects; and
- to deal with the impact of mobility, particularly to deal with the case when the client-server communication is unexpectedly severed from the client site.

Research work from various authors (Breitbart et al., 1999; Franklin, Carey & Livny, 1997; Jensen & Lomer, 2001; Pacitti, Minet & Simon, 1999; Shanmugasundaram et al., 1999; Schuldt, 2001) have contributed to the investigation of aspects related to ensuring serializability of data management. Based on these studies, Parker and Chen (2004) have conducted a more recent research to deal with the three issues mentioned above and developed algorithms to achieve extended invalid-access prevention protocols. The basic ideas of this research are summarized below.

Table 1. Extended invalid-access prevention policy

ATTRIBUTE	S2PL	O2PL	CBL
Version Numbers	X	**X**	**X**
Recreate/release page table rows		**X**	**X**
Permit before Commit			X
Lock before Commit	X		
Commit before Lock		X	
Invalidation		X	X
Dynamic Replication	X	X	
Broadcast		X	X
Read-write conflict		X	X
Write-read conflict	X	X	X
Write-write conflict	X	X	
Relinquish unused locks at sign-off	**X**	**X**	**X**
Maximum lock duration	**X**	**X**	**X**
Server knows who has locks	X	X	X
Server knows who has what objects		X	X

First, in order to prevent the serializability failure scenario described above, we summarize important features of the extended invalid-access prevention policy protocols for the mobile client/server environments that guarantee the serializability. As shown in Table 1, an X denotes an attribute under the standard invalid-access prevention policy, while a bold-face **X** as an additional attribute under the extended invalid-access prevention policy. The revised algorithms for extended invalid-access prevention policy protocols are developed based on these considerations.

As an example of these attributes, here we take a brief look at the important role of the page table. To detect or avoid invalid-accesses from all transactions, all clients and the server each need to keep a separate table to detect or avoid data conflict situations. For clients, page tables are the current inventories of their cached multimedia objects. For the server, a page table is the information