Transaction Management for Object-Oriented Databases: Performance Advantages of Using Multiple Versions

Xiao Ying Liu
John A. Miller
Nilesh R. Parate

Department of Computer Science
The University of Georgia
Athens, Georgia 30602

1. Extended Abstract

Object-Oriented Databases are becoming very popular as these databases reflect the real world more accurately and realistically than the Relational Databases that are being used today. However, Object-Oriented Databases are currently at the stage of research and development that Relational Databases were ten years ago. Much work needs to be done, in theory and implementation, to fully exploit and realize the potential of Object-Oriented Databases.

The main reason for the widespread interest in Object-Oriented Databases today, is due to the realization that Information Intensive applications like Simulation, Expert Systems, Knowledge Bases, and Computer-Aided Design will in the future, demand very high performance databases, which are beyond the scope of the currently used Relational model.

The protocols that are being used for Transaction Management and Concurrency Control in the Relational Databases, will not exploit the object-oriented features of Object-Oriented Databases sufficiently to handle the above mentioned applications. In addition, the fact that parallel and distributed computing systems are becoming more prevalent, suggests that protocols supporting concurrency levels higher than those provided by two-phase locking (today's most common protocol) are needed.

Versioning capabilities are becoming more and more common in object-oriented databases. The primary motivation for the inclusion of such a capability is that many advanced applications demand it. By being able to manage different versions of prototypes of a project (e.g., a VLSI chip design project), Object-Oriented Databases could help in better management of research programs, and in reducing the amount of resources required when a project has to be backtracked to previous versions. Any project that requires a large number of development versions

over a period of time, will find these features of Object-Oriented Databases invaluable. Another benefit of having multiple versions of objects is that performance can be enhanced, since conflicts that can lead transactions to block or abort are reduced.

1.1. Concurrency Control Protocols

The purpose of concurrency control protocols is to ensure the correctness or the serializability of a database. To achieve the goal of concurrency control, many protocols are proposed and basically fall into three categories:

1. Locking protocols such as Two-Phase Locking (TPL) (Eswaran, Gray, Lorie & Traiger, 1976; Stonebraker, 1979; Traiger et al., 1982).
2. Timestamp based protocols such as Timestamp Ordering (TSO) (Thomas, 1979; Bernstein & Goodman, 1980).
3. Optimistic protocols such as Kung & Robinson Optimism (KRO) (Kung & Robinson, 1981).

To improve the performance of a database, multiversion concurrency control protocols are designed based on the basic single version protocols. Multiversion protocols that fit into the three basic categories are shown below.

1. Two-Version Two-Phase Locking (TVTPL) (Bayer, Heller & Reiser, 1980; Stearns & Rosenkrantz, 1981) and Multiversion Two-Phase Locking (MVTPL) (Bernstein & Goodman, 1981).
2. Multiversion Timestamp Ordering (MVTSO) (Reed, 1978, 1983).
3. Time Warp (TW) (Jefferson, 1985, 1986; Miller & Griffeth, 1991, 1992).

Performance and complexity of implementation are two criteria for choosing a concurrency control protocol (Ahuja & Browne, 1988; Dan, Towsley & Kohler, 1988). The findings of performance studies on concurrency control protocols are inconclusive and sometime contradictory because different assumptions about the underlying systems, various parameters and different measurement metrics were used in those studies.

Some researchers reported that Multiversion Timestamp Ordering protocols offered a significant performance improvement when compared to single version Timestamp Ordering protocols (Miller, 1986; Carey & Muhanna, 1986), and that Timestamp Ordering outperformed Two-Phase Locking (Miller, 1986). Other researchers, however, found that Multiversion Timestamp Ordering only marginally better than basic Timestamp Ordering (Lin & Nolte, 1983; Son & Haghighi, 1990). When the transaction size was small, Timestamp Ordering had better performance when compared to Two-Phase Locking. When the transaction size was larger, however, Two-Phase Locking outperformed both basic and multiversion Timestamp Ordering (Lin

& Nolte, 1983). Moreover, some researches demonstrated that Two-Phase Locking outperformed Timestamp Ordering which was in turn better than an optimistic timestamp-based protocol (Carey & Livny, 1989).

Surprisingly enough, although the Two-Version Two-Phase Locking and Multiversion Locking protocols were proposed for more than ten years, no study on the performance of those protocols was found in the literature. Additionally, recent results (Miller & Griffeth, 1991, 1992) indicate high promise for versions of the Time Warp protocol. (Time Warp protocols which are popular for parallel and distributed simulation, have been adapted by Jefferson to work for database concurrency control, and modifications to Jefferson's version have recently been done by Miller and Griffeth.) Therefore, more research is definitely needed to determine the performance of concurrency control protocols for object oriented databases, particularly those that support multiple versions.

1.2. Simulation Models

The present study is designed to investigate the performance of concurrency control protocols for object oriented databases. As mentioned previously, versioning capabilities are becoming more common in object-oriented databases. Therefore, in this paper we consider the possible performance advantages that can be obtained by exploiting the fact that multiple versions of objects are present in object-oriented databases. Specifically, the performance of single version Two-Phase Locking is compared with both Two-Version Two-Phase Locking, and Multiversion Two-Phase Locking; Timestamp Ordering is compared with Multiversion Timestamp Ordering; and Kung & Robinson Optimism is compared with Time Warp.

The performance of the protocols is evaluated by comparing the results of simulation models in terms of the throughput in transactions per second. The simulation models are coded in the SIMODULA simulation language (Miller et al., 1988, 1989). Our preliminary results indicate that multiple version based protocols can perform substantially better than their single version equivalents.