Proceedings of the International Conference on Computer and Information Science and Technology
Ottawa, Ontario, Canada, May 11 – 12, 2015
Paper No. 135

Dynamic Timestamps Ordering

A. Farrag
Faculty of Computer Science
Dalhousie University
Halifax, N.S., Canada, B3H 3J5
zizo@dal.ca

Extended Abstract

This paper proposes several timestamp ordering mechanisms in which the timestamps assigned to transactions can be changed dynamically during execution. These timestamps are not stored with the entities (of the database) and the process of modifying them is simple. The proposed mechanisms achieve a higher level of concurrency (than traditional timestamp ordering mechanisms) for the following reasons. First, the operations of (certain classes of) read-only transactions can always be accepted. Second, the mechanisms avoid rejecting an operation of an update transaction by modifying its current timestamp. The proposed mechanisms do not require multiversion of each entity to be stored, but it maintains a digraph that represents the "conflicts" among transactions. The complexity is reduced by searching the digraph only when the acceptance of an operation may violate consistency; otherwise, the operation is accepted without search.

1. Introduction

In any multiuser database system, a concurrency control mechanism is needed to resolve any conflicts that might arise among transactions, and to ensure that their overall execution is correct, i.e., cannot violate consistency (Papadimitriou1979). Many such mechanisms have been previously proposed, e.g., (Agrawal1985, Bayer1980, Conway2012, Eswaran1976, Farrag1982, Garcia-Molina1982, Gray1978, Kung1981, Lamport1976, Laux2009, Lessner2011, Reed1978, Rosenkrantz1978); and most of them utilize some form of locking, e.g., (Eswaran1976, Farrag1987, Gray1976, Kedem1979, Rosenkrantz1978, Usui2010). That is, before a transaction can access any entity, it must first obtain an (appropriate) lock on it, and if this lock cannot be granted, the transaction will be delayed. This can potentially lead to a deadlock, which can be resolved by aborting one or more transactions.

Other approaches to controlling concurrency avoid delaying the transactions by assigning each of them a unique timestamp, and then requiring all conflicting operations to be processed in a timestamp order, e.g., (Farrag1987, Lamport1976, Reed1978, Thomas1978). In this case, when an operation arrives out of order, it will be rejected, that is, the transaction that issued it will be aborted. Abortion is a serious drawback, and can degrade performance if it occurs frequently.

To avoid unnecessary abortions, we propose some dynamic timestamps ordering methods in which the timestamps assigned previously to transactions can be dynamically modified during execution. The timestamps are not stored with the entities in our methods, and the process of modifying them is a simple operation. As will be shown, separating timestamps from the entities will help to solve some issues that arise in all timestamp ordering mechanisms.

References

Agrawal R., Dewitt D. (1985). “Integrated Concurrency Control and Recovery Mechanisms: Design and Performance Evaluation”, ACM Trans. Database Systems, 10, 4, Dec, pp. 529-564.

Bayer R, Heller H, Reiser A. (1980). "Parallelism and Recovery in Database Systems," ACM Trans, Database Systems 5, 2, Jun, pp. 139-156.

Conway N, Marczak W., Alvaro P., Hellerstein J, Maier D. (2012), Logic and lattices for distributed programming, "Proceedings of the 3rd ACM Symposium on Cloud Computing", San Jose, CA, USA Oct 14, pp. 1-10.

Eswaran K, Gray J, Lorie R., Traiger I. (1976). "The Notions of Consistency and Predicate Locks in a Database System", Comm. ACM, 19, 11, Nov, pp. 624-633.

Farrag A., Kameda T. (1982). "On Concurrency Control Using Multiple Versions," Technical Report TR 82-13, Dept. Computing Science, Simon Fraser University, BC, Canada, 1982.

Farrag A., Ozsu M. (1987). "Towards a General Concurrency Control Algorithm for Database Systems" IEEE Trans. on Soft. Eng.\fR, Vol. SE-13, No. 10, Oct, pp. 1073-1079.

Garcia-Molina H.,Wiederhold G. (1982)."Read-only Transactions in Distributed Database", ACM Trans Database Systems, 7, 2, Jun, pp, 209-234.

Gray J. (1978). "Notes on Database Operating Systems", in Operating Systems: An Advanced Course, Lecture Notes in Computer Science, Springer-Verlag, N.Y, pp. 393-481.

Gray J., Lorie R., Putzolu G. and Traiger L. (1976). Granularity of Locks and Degrees of Consistency in a Shared Database, "Proc. IFIP Working Conf. on Modeling of Databases", Jan, pp, 695-723.

Kedem Z., Silberschatz A. (1979). Controlling Concurrency Using Locking, "Proceedings of the 20th International Conference on Foundation of Computer", Oct, pp. 274-285.

Kung H., Robinson J. (1981). "On Optimistic Methods for Concurrency Control", ACM Transactions on Database Systems, 6, 2, Jun, pp. 213-226.

Lamport L. (1976). "Towards a Theory of Correctness of Multi-user Database Systems", Massachusetts Computer Associates Tech. Rep. CA 764-0712, Oct.

Laux F., Lessner T. (2009). "Escrow Serializability and Reconciliation in Mobile Computing using Semantic Properties", Int. Journal on Advances in Telecommunications, V 2, N 2&3, pp; 72-87.

Lessner T., Laux, F. Connolly T., Crowe M. (2011). "Transactional Composition and Concurrency Control in Disconnected Computing", Int. J. on Advances in Software, V. 4, N. 3&4, pp. 442- 460.

Papadimitriou C. (1979). "The Serializability of Concurrent Database Updates", J. ACM 26, 4, Oct, pp. 631-653.

Reed D. (1978). "Naming and Synchronization in Decentralized Computer Systems", Dept. Electrical Engineering and Computer Science, MIT Tech. Rep.-205, Sep.

Rosenkrantz D., Stearns R. and Lewis P. (1978). "System Level Concurrency Control for Distributed Database Systems", ACM Trans. Database Systems, 3, 2, Jun, pp. 178-198.

Thomas R. (1978). A Solution to the Concurrency Control Problem for Multiple Copy Databases, in "Proceedings of the COMPCON Conference", N.Y.

Umar A., Teichroew D. (1990), "Pragmatic issues in conversions of database applications", Information & Management, V.19, N. 3, Oct, pp. 149-166.

Usui T., Behrends R., Evans J., Smaragdakis Y. (2010). "Adaptive locks: Combining transactions and locks for efficient concurrency", J. of Parallel and Dist Comp., V. 70, N. 10, Oct, pp. 1009-1023.