Memory-Mapped Transactions

by

Jim Sukha

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

Author .....
Department of Electrical Engineering and Computer Science
May 19, 2005

Certified by .....
Charles E. Leiserson
Professor of Computer Science and Engineering
Thesis Supervisor

Certified by .....
Bradley C. Kuszmaul
Research Scientist
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Memory-Mapped Transactions

by

Jim Sukha

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2005, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Memory-mapped transactions combine the advantages of both memory mapping and transactions to provide a programming interface for concurrently accessing data on disk without explicit I/O or locking operations. This interface enables a programmer to design a complex serial program that accesses only main memory, and with little to no modification, convert the program into correct code with multiple processes that can simultaneously access disk.

I implemented Libxac, a prototype for an efficient and portable system supporting memory-mapped transactions. Libxac is a C library that supports atomic transactions on memory-mapped files. Libxac guarantees that transactions are serializable, and it uses a multiversion concurrency control algorithm to ensure that all transactions, even aborted transactions, always see a consistent view of a memory-mapped file. Libxac was tested on Linux, and it is portable because it is written as a user-space library, and because it does not rely on special operating system support for transactions.

With Libxac, I was easily able to convert existing serial, memory-mapped implementations of a B^+-tree and a cache-oblivious B-tree into parallel versions that support concurrent searches and insertions. To test the performance of memory-mapped transactions, I ran several experiments inserting elements with random keys into the Libxac B^+-tree and Libxac cache-oblivious B-tree. When a single process performed each insertion as a durable transaction, the Libxac search trees ran between 4% slower and 67% faster than the B-tree for Berkeley DB, a high-quality transaction system. Memory-mapped transactions have the potential to greatly simplify the programming of concurrent data structures for databases.

Thesis Supervisor: Charles E. Leiserson
Title: Professor of Computer Science and Engineering

Thesis Supervisor: Bradley C. Kuszmaul
Title: Research Scientist

Acknowledgments

These acknowledgments are presented in a random order:

I would like to thank my advisor, Charles E. Leiserson for his helpful comments on this thesis, and on his helpful advice in general.

I would also like to thank Bradley C. Kuszmaul, who has been deeply involved in this project from day one. Bradley's initial idea started me on this project that eventually became LIBXAC, and I have had many helpful discussions with him on this topic ever since.

I'd like to thank Bradley for giving me an implementation of a B^+-tree, and Zardosht Kasheff for the code for the cache-oblivious B-tree. The experimental results on search trees without LIBXAC are primarily the work of Bradley, Michael A. Bender, and Martin Farach-Colton.

All of the people in the SuperTech group deserve a special round of acknowledgments. Those people in SuperTech, also in random order, include Gideon, Kunal, Jeremy, Angelina, John, Tim, Yuxiong, Vicky, and Tushara. Everyone has been wonderfully patient in listening to my ramblings about transactions, names for LIBXAC, research, and life in general. They have all kept me (relatively) sane throughout the past year. In particular, I'd like to Kunal, Angelina, and Jeremy for their feedback on parts of my thesis draft, and more generally for choosing to serve the same sentence. I'd also like to thank Ian and Elizabeth, who are not in the SuperTech group, but have also been subjected to conversations about transactions and other research topics. Thanks to Leigh Deacon, who kept SuperTech running administratively.

Thanks to Akamai and MIT for the Presidential fellowship that funded my stay here this past year. Also, thanks to the people I met through SMA for their helpful comments on the presentation I gave in Singapore.

Thank you to my family and to my friends, here at MIT and elsewhere. Without their support, none of this would have been possible.

I apologize to all those other people I am sure I have missed that deserve acknowledgments. I will try to include you all when the next thesis rolls around.

This work was partially supported by NSF Grant Numbers 0305606d and 0324974, and by the Singapore MIT Alliance. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Contents

List of Figures

List of Tables