

Summary: Dagstuhl Seminar 08241: Transactional Memory: From Implementation to Application

Organizers:

Christof Fetzer, Technische Universität Dresden, Germany

Tim Harris, Microsoft Research, Cambridge, GB

Maurice Herlihy, Brown University, USA

Nir Shavit, Tel-Aviv University, Israel

A goal of current multiprocessor software design is to introduce parallelism into software applications by allowing operations that do not conflict in accessing memory to proceed concurrently. The key tool in designing concurrent data structures has been the use of locks. Unfortunately, coarse grained locking is easy to program with, but provides very poor performance because of limited parallelism. Fine-grained lock-based concurrent data structures perform exceptionally well, but designing them has long been recognized as a difficult task better left to experts. If concurrent programming is to become ubiquitous, researchers agree that one must develop alternative approaches that simplify code design and verification.

The *transactional memory* programming paradigm is on its way to becoming the approach of choice for replacing locks in concurrent programming. Combining sequences of concurrent operations into atomic transactions seems to promise a great reduction in the complexity of both programming and verification, by making parts of the code appear to be sequential without the need to program fine-grained locks. Transactions will hopefully remove from the programmer the burden of figuring out the interaction among concurrent operations that happen to conflict when accessing the same locations in memory. There has been a flurry of work on transactional memory systems, hardware implementations (HTM), purely software based ones, i.e. software transactional memories (STM), and hybrid schemes (HyTM) that combine hardware and software.

The need for techniques such as transactional memory has become ever so urgent with the shift in our basic computing paradigm from single core chips to multi-core chips: we now have multi-cores on every desktop and laptop, and we need ways of programming them efficiently. Many of the big hardware and software vendors are scrambling to invest in transactional memory research and development, and academia is in a great need to coordinate the many world-wide research efforts on the topic.

This Dagstuhl seminar brought together leading researchers working on transactional memory (HTM, STM, and HyTM) from both industry and academia, in order for the dialog to help to create a concerted effort in pushing forward this important research agenda. We discussed several ongoing research directions in transactional memory:

- What are the fundamental approaches in development of STM platforms for the new class of multi-core machines?
- What support do transactions need in hardware, compilation, library support, and how would HTM and STM systems interoperate?
- How do we build TM-based applications and what should the STM/HTM implementations provide to simplify their design given application development experience?
- What programming language transactional constructs would best help in programming, and what are the issues with their implementation?
- Can we put together a complete transactional programming environment given current knowledge, that is, hardware, software, and programming language support?

In summary, our seminar addressed various issues of STM design from both the implementation and application perspectives, with a meaningful and intense discussion among the researchers that improved our understanding of both. Hopefully, this will eventually result in better transactional memory platforms that make for faster and simpler-to-program concurrent applications.

1 List Participants

Name, Firstname	Institution/Company	Country
Adl-Tabataba, Ali-Reza	Intel	USA
Afek, Yehuda	Tel Aviv University	Israel
Attiya, Hagit	Technion	Israel
Annette Bieniusa	Universität Freiburg	Germany
Dice, Dave	Sun Microsystems	USA
Lucia Draque Penso Rautenbach	Brown Univ. - Providence	USA
Faith Ellen	(U of Toronto)	Canada
Alexandra Fedorova	Simon Fraser University - Burnaby	Canada
Pascal Felber	University of Neuchatel	Switzerland
Christof Fetzer	Technische Universität Dresden	Germany
Grossman, Dan	Washington	USA
Guerraoui, Rachid	EPFL	Switzerland
Tim Harris	Microsoft Research, Cambridge	GB
Maurice Herlihy	Brown University	USA
Lisa Higham	(U of Calgary)	Canada
Eshcar Hillel	Technion - Haifa	Israel
Idit Keidar	Technion	Israel
Guy Korland	Tel Aviv University	Israel
Kozyrakis, Christos	Stanford	USA
Petr Kuzin	Max Planck Institute	Germany
Yossi Lev	Brown Univ. - Providence	USA
Luchangco, Victor	Sun Microsystems	USA
Maged Michael	IBM TJ Watson Research Center	USA
Moir, Mark	Sun Microsystems	USA
J. Eliot B. Moss	Univ. of Massachusetts - Amherst	USA
Nussbaum, Dan	Sun Microsystems	USA
Olukotun, Kunle	Stanford	USA
Panagiota Fatourou	University of Ioannina	Greece
Rajwar, Ravi	Intel	USA
Torvald Riegel	TU Dresden	Germany
Rodrigo Rodrigues	MPI - SWS	Germany
William N. Scherer III	Rice University	USA
Marc Shapiro	INRIA	France
Nir Shavit	Tel-Aviv University	Israel
Gad Sheaffer	Intel	Israel
Asuman Suenbuel	SAP Research Labs - Palo Alto	USA
Osman Unsal	Barcelona Supercomputing Center	Spain
David A. Wood	Univ. Wisconsin - Madison	USA
Craig Zilles	Univ. of Illinois - Urbana	USA