WORKSHOP ON TRANSACTIONAL SYSTEMS

Call for Position Papers

Chicago, IL (Hilton at O'Hare airport)
April 8, 2005 (8:00 AM to 4:30 PM)

The TS workshop intends to bring together leading researchers and practitioners from various communities, such as hardware architecture, programming languages, distributed systems, applications, runtime environments, and operating systems to develop an understanding of the challenges and requirements, from the software perspective of providing hardware support for transactions and atomicity.

Around 35 faculty and industry participants are expected.

This is a one-day by-invitation workshop and not a conference, and the invitation is not transferable.

OVERVIEW

A transaction is a finite sequence of memory reads and writes, and computation executed by a single thread. In our context, a transaction can be viewed as a block of instructions which executes atomically and with most of the traditional "ACID" properties, namely atomicity, consistency, and isolation but perhaps not durability, Transactions are atomic where each transaction either commits and makes its effects visible or aborts, thereby discarded any of its effects. Transactions are serializable where they appear to take effect one-at-a-time order.

In the transactional memory computational model, each thread starts a transaction, executes a sequence of operations on shared objects, and then tries to commit the transaction. If the commit succeeds, the transaction's operations take effect; otherwise they are discarded. Transactional memory avoids the limitations associated with locks, and has the potential to simplify programming and provide high-performance support for multithreaded applications.

Hardware advances and common hardware techniques such as speculative execution and on-chip hardware caching have made efficient and cheap implementations of transactional memory possible. With the advent of multi-core systems, the potential for transactions as a programming model for developing robust, reliable, and high-performance multi-threaded applications is immense.

However, developing a better understanding of the implications of such a model requires engaging research communities in various disciplines beyond computer architecture.

A broader topic of atomicity in system design and execution was covered at a Dagstuhl seminar last year and its seminar report can be found at:
http://www.cs.wisc.edu/~rajwar/papers/sigmodrecord_mar2005.pdf

CALL FOR POSITION PAPERS

The focus of this workshop will broadly be:

1. Programming methodologies with software and hardware transactions
2. Guarantee/exception recovery models
3. Operating systems/run-time software system interactions with transactions
4. Evaluation/application development strategies for transactional systems

We invite participants to submit (in ONE-page ASCII text format only) a short summary, position statement, or an abstract on current work and interests. The submission may advocate a particular viewpoint, take sides on a controversial subject, or may be a discussion of past experiences in such systems.

The position papers will help us set the agenda and format for the workshop, and ensure a productive workshop. All submissions will be made available to the participants prior to the workshop.

Please send the position paper electronically to ravi.rajwar@intel.com latest by March 15, 2005 by electronic mail in PLAIN TEXT.

ORGANIZING COMMITTEE

Krste Asanovic, MIT
Christos Kozyrakis, Stanford
Konrad Lai, Intel
Ravi Rajwar, Intel
David Wood, Wisconsin

IMPORTANT DATES

Position paper submission deadline:	March 15, 2005
Workshop date:	April 8, 2005