EVENT SCHEDULE

8:30-8:50	Breakfast & Registration
8:50-9:00	Welcome
9:00-9:30	Yossi Lev, Brown University
9:30-10:10	Ulrich Drepper, Red Hat
10:10-10:25	Break
10:25-11:05	David Christie, AMD
11:05-11:45	Srinivas Sridharan, University of Notre Dame
11:45-1:00	Buffet Lunch
1:00-1:40	Ben L. Titzer, Sun Microsystems
1:40-2:20	Mark Moir, Sun Microsystems
2:20-3:00	Ali Adl-Tabatabai, Intel
3:00-3:20	Break
3:20-4:00	Maged Michael, IBM
4:00-4:40	Rob Schreiber, HP
4:40-5:30	Reception

DIRECTIONS TO THE CIT BUILDING

*From either I-95 South or I-95 North, take Exit 22A, Downtown, Memorial Boulevard.
*At the fifth traffic light, turn left onto College Street. Cross the bridge and proceed up the hill following College Street to the end.
*At the top of the hill, where College Street intersects with Prospect Street, you will see Brown's Van Wickle Gates. Turn left onto Prospect Street.
*At the next light, turn right onto Waterman Street. At the second light, turn right onto Brook Street. The visitor parking lot will be on your left immediately after turning onto Brook Street.
*The CIT – a large red and tan brick building – is across the street from the visitor parking lot, on the corner of Waterman and Brooke; the entrance is on the other side, facing the quadrangle.
*Event registration is on the third floor.

PARKING

Brown has a Visitor Parking Lot across the street from the CIT on Brook Street; this lot charges an hourly fee.

RSVP to abt@cs.brown.edu by April 20, 2009

The primary goals of the Industrial Partners Program (IPP) are to exceed the expectations of our partner companies in terms of recruiting and outreach; to allow our faculty to engage in challenging and meaningful research collaborations and to provide resources and employment opportunities for our students. The Department wishes to thank our industrial partners:

Premier Partners
Adobe
Network Appliance
Sun Microsystems

Affiliates
Apple
Data Domain
Facebook
Google
GTECH
Microsoft
Oracle
VMware

Individuals
Paul Edelman, Edelman & Associates
Robert Khoury, Worldwide Financial Industry Recruiting Services

To learn more about the Industrial Partners Program, contact:
Ugur Cetintemel, Faculty Director
Telephone: 401-863-7600
ipp@cs.brown.edu

Amy Tarbox, Program Manager
Telephone: 401-863-7610
abt@cs.brown.edu

Thursday, April 30, 2009
8:30 AM – 5:30 PM
Room 368
Watson Center for Information Technology

STANDARDIZING TRANSACTIONAL MEMORY

BROWN UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
INDUSTRIAL PARTNERS PROGRAM
42nd BIANNUAL SYMPOSIUM

HOSTED BY
MAURICE HERLIHY

YOSSI LEV, Ph.D. Student, Brown University

A Generic Debugging Infrastructure for Transactional Programs

We introduce tm_db, an open-source library to provide debuggers with a general debugging support for transactional programs. The library helps debuggers provide programmers with generic transactional debugging features, independent of the particular TM's runtime internals. In addition, it provides TM designers with a well defined interface for transactional debugging support.

ULRICH DREPPER, Red Hat, Inc.

Introducing TM into the "Real World"

TM is a promising technology which can yield significant enhancements in ease-of-programming and performance. There is a huge, existing body of code out there which doesn't use this technology but still has to be used.

This talk will describe the perspective of an OS vendor and ways envisioned to gradually introduce TM into the life of programmers without disrupting the status quo, especially with respect to performance and complexity of the development progress.

DAVID CHRISTIE, AMD

Can Mainstream Processors Support Hardware Transactional Memory?

While Hardware Transactional Memory concepts have been around for more than a decade, actual implementation in any high-volume commercial microprocessor has been a chicken-and-egg problem -- HTM poses enough complexity for the hardware that some serious justification is needed, yet it's been difficult to demonstrate the potential benefit on real applications due to modeling and application development limitations. On top of that, there are many places the line can be drawn between software and hardware, and finding a sweet spot is difficult without knowing how TM applications will tend to behave. Yet another dimension is added by the need for commonality in the software development infrastructure to support portability. This talk will cover some of the difficulties in providing HTM support, and what a best-effort approach might look like for the x86 architecture.

SRINIVAS SRIDHARAN, University of Notre Dame

A Scalable Software Transactional Memory System for the Chapel High-Productivity Language

Chapel is a parallel language being developed by Cray Inc as part of the DARPA-led High Productivity Computing Systems program (HPCS).

Chapel strives to increase productivity by supporting higher levels of abstraction compared to current parallel programming models while supporting the ability to incrementally optimize for performance as and when the programmer chooses. In this talk, we present our on- going work in the design and implementation of a scalable Software Transactional Memory (STM) system for the Chapel High-Productivity Language.

BEN L. TITZER, Sun Microsystems

The Maxine Virtual Machine

Virtual machines for modern object-oriented languages are among the most complicated software systems in the world today, and top performance demands high quality memory management, compilation, and runtime systems. However, the fast pace of language changes and the pursuit of ever more efficient implementation techniques demand more flexibility in the software architecture than current industrial virtual machines provide. This talk will give an overview of the Maxine VM's design and its focus on modularity and explain how increasing meta-circularity can provide advantages over traditional VM designs. Briefly, some experience on the early stages of adapting Maxine to support a STM system will be presented, as well as a perspective on how STM may affect the future design of optimizing compilers.

MARK MOIR, Sun Microsystems

Transactional Memory at Sun

Mark will talk about work related to transactional memory going on at Sun, particularly related to interfaces at various levels, and how it relates to relevant work by others in industry.

ALI-REZA ADL-TABATABAI, Intel

Towards a Common Language and Runtime Interface for Transactional Memory in C++

The transactional memory community is now at a stage where it needs developers to experiment with writing large programs using transactional memory programming constructs. This is important not only to prove and refine the TM programming model, but also to tune the performance and scalability of TM implementations on non-trivial workloads. To facilitate writing large applications using TM, several groups have released compilers that support language extensions for TM in C or C++. In this talk, I argue that incompatible language interfaces will hamper adoption of TM.

MAGED MICHAEL & VIJAY SARASWAT, IBM Watson Research Center

IBM STM API and X10 Extensions

In this talk we present the main components of the API for the IBM STM runtime system. The common IBM STM runtime is written in C and runs on a variety of platforms as well as architectural models, and supports Java and C/C++ compilers as well as direct programmer instrumentation. The X10 programming language and common patterns in high performance computing present opportunities for STM optimization. We present STM API extensions that enable these optimizations.

Rob Schreiber, HP

Talk Title and Abstract TBD