

Technical Perspective

Software and Hardware Support for Deterministic Replay of Parallel Programs

By Norman P. Jouppi

PARALLEL PROGRAMMING HAS long been recognized as a difficult problem. This problem has recently taken on a sense of urgency: the long march of single-thread performance increases has stopped in its tracks. Due to limitations on power dissipation and decreased return on investments in additional processor complexity, additional transistors provided by Moore's Law are now being channeled into a geometrically increasing number of cores per die. These cores can easily be applied to embarrassingly parallel problems and distributed computing in server environments by running multiple parallel tasks. However, there are many applications where increased performance on formerly single-threaded applications are highly desirable, ranging from personal computing devices to capability supercomputers.

A key problem in parallel programming is the ability to find concurrency bugs and to debug program execution in the presence of memory races on accesses to both synchronization and data variables. Subsequent executions of a parallel program containing a race or bug are unlikely to have the same exact ordering on each execution due to nondeterministic system effects. This may cause rare problems to occur long after deployment of an application. Rerunning programs in a slower debug mode also changes the relative timing, and can easily mask problems. Ideally what we'd like is a way to deterministically replay execution of parallel programs, by recording the outcome of memory races without significantly slowing down the execution of the original program. Additionally, one would like logging requirements of the execution to be manageable, and the replay of applications to occur at a speed similar to that of the original execution.

An important step in this direction appeared five years ago in the University of Wisconsin's Flight Data Recorder.¹ This original system required significant hardware state. However, for mainstream adoption, the additional hardware should be very small, since all users of a microprocessor design will be paying for the hardware support whether they use it or not. After several evolutionary enhancements in previous years, two systems appeared this year that make a quantum leap forward in reducing the overheads needed to support deterministic replay: instead of recording individual memory references, both of these systems only record execution of atomic blocks of instructions.

Rerun, also from the University of Wisconsin, reduces overhead by recording atomic episodes. An episode is a series of instructions from a single thread that happen to execute without conflicting with any other thread in the system. Episodes are created automatically by the recording system without

The following paper is a first for *Communications'* Research Highlights section: it contains a synthesis of recent work from two competing (but collegial) research teams.

modification of the applications. Rerun uses Lamport Scalar Clocks to order episodes and enable replay of an equivalent execution. Rerun reduces the hardware state per core to 166 bytes per core and the log size to only around 1.67 bytes/kiloinstruction per core in an 8-core system. This results in a core*log overhead product for many-core systems that is more than an order of magnitude smaller than previous work.

DeLorean, developed contemporaneously at the University of Illinois, executes large blocks of instructions atomically separated by checkpoints, like in transactional memory or thread-level speculation. Executing larger chunks of instructions provides benefits in both log size and replay speed. For example, for an 8-core processor, DeLorean is able to achieve a log size of only 0.0063 bytes/kiloinstruction per core while still being able to replay at 72% of the original execution speed. To put this in perspective, with this log size an entire day's execution of an 8-core processor would only take 20GB, a small fraction of a 1TB disk drive.

The following paper is a first for *Communications'* Research Highlights section: it contains a synthesis of recent work from two competing (but collegial) research teams. Both the Rerun and DeLorean teams were invited to contribute to this paper since their approaches appeared in the same conference session, both represent significant advances, and their approaches are actually complementary—Rerun requires very little additional hardware, whereas DeLorean can achieve much smaller log sizes but requires checkpoint and recovery hardware similar to that provided in transactional memory systems. Both research streams have the potential to make a significant impact on the productivity of future parallel programming. **□**

Reference

1. Xu, M., Bodik, R., and Hill, M.D. A "flight data recorder" for enabling full-system multiprocessor deterministic replay. *ACM/IEEE International Symposium on Computer Architecture*, June 2003.

Norman P. Jouppi (Norm.Jouppi@hp.com) is a Fellow and Director of Hewlett-Packard's Exascale Computing Lab in Palo Alto, CA.