Dusty Caches for Reducing Reference-Counting Memory Traffic

Scott Friedman, John Lockwood, Ron Cytron, Roger Chamberlain, and Jason Fritts
Washington University
cytron@acm.org

The major goal of our project is to measure and improve application performance, by providing an easily and efficiently reconfigurable architecture along with software support to expedite its use.

Our FPGA-based system is implemented as an extensible hardware module on the Field-programmable Port Extender (FPX) platform at Washington University. The FPX platform provides an environment where a circuit implemented in FPGA hardware can be developed, uploaded, and tested with live data using web and network interfaces.

Our research begins with LEON, a soft-core implementation of a SPARC instruction set. We can easily add instructions and microarchitecture enhancements to LEON and evaluate their performance using a cycle-accurate profiling circuit that we have designed to operate with LEON.

Reference counting is a garbage-collection technique that maintains a per-object count of the number of pointers to that object. Whent he count reaches zero, the object must be dead and can be collected. Although it is not an exact method, it is well suited for real-time systems and is widely implemented. A disadvantage of reference counting is the extra storage traffic that is introduced.

In this talk we describe a new cache write-back policy that can substantially decrease the reference-counting traffic to RAM. Write-back implementations maintain a *dirty* bit that indicates that represented unit of storage has potentially changed and must be written back to RAM. A very common sequence of events for a reference count is that the count temporarily increases from some value v but then very soon returns to the value v. For example, iteration across a link list has such an effect. The reference count in cache is marked dirty, but the value that would be written back to RAM is in fact the same value already stored there.

We propose a new cache architecture that remembers the first-fetched value of a cache subblock, so that the subblock need not be written back to RAM unless a different value is present. We are implementing the approach in the context of our FPGA work described above, and we present results from experiments that show the effectiveness of this approach. This work is supported by the National Science Foundation under grant ITR–0313203.