

15-740 Project Milestone Report

Nov 17, 2009

Anvesh Komuravelli (akomurav), Samir Sapra (ssapra) and Jenn Tam (jdtam)

Major Changes

We read more recent work [1,2] discussing contention management from a different angle. Contention management is “not just” conflict resolution. If properly done, it is something important throughout the life of a transaction. So, our project now focuses on comparing this new contention management policy (called Fair policy) with existing policies for STAMP benchmarks.

Work done so far

Initially, we spent considerable amount of time trying to make the tools work, viz., STAMP on RSTM (Rochester Software Transactional Memory), simics to work together with GEMS. While we were successful in making STAMP run on RSTM, we still have some problems in making GEMS run. Meanwhile we talked to Prof. Todd Mowry and got access to an 8 core Intel Xeon parallel machine on which we started taking results directly instead of using simulators.

Soon after our project began, we were inspired by a personal discussion with Prof. Onur Mutlu, to first analyze the various STAMP benchmarks in order to study the behavior of the transaction lengths. We posted the results on our [website](#) and they can be found [here](#). In order to do this, we had to go through the source code of RSTM implementation and insert our instrumentation code to keep track of the lengths of transactions (considering only the useful time spent in a successful attempt of the transaction and avoiding calculating the time spent in aborts). We used the ‘Fair’ contention manager in order to get the data we wanted. We would like to study these distributions to see if we can propose a better contention manager for real applications like those in STAMP.

We have started instrumenting the RSTM code to collect more detailed timing information including the time taken for repeated aborts in a transaction so that we can collect statistics like number of commits per unit time, time taken for a transaction to commit, etc. We will be needing these values to make a comparison of ‘Fair’ with other existing contention managers.

Meeting our Milestone

We are slightly behind the milestone we stated in our proposal where we should have completed instrumenting RSTM and comparing the various managers by now, for the following reasons: Firstly, we had to change the aim of our project slightly in the light of some new papers; secondly, we had trouble making the tools work on the platforms with which all the tools comply.

Schedule for future

Phase 1

Samir : Complete instrumentation of RSTM to do the timing analysis we require.

Anvesh and Jenn : Take the results on various STAMP benchmarks for various managers and make a reasonable comparison.

Phase 2

Samir : Code up the new manager designed based on the observations.

Anvesh and Jenn : Compare it with the earlier managers for various benchmarks.

[1] Mohammad Ansari and Christos Kotselidis and Mikel Luján and Chris Kirkham and Ian Watson
Investigating Contention Management for Complex Transactional Memory Benchmarks.

In: MULTIPROG~'09: Proc. 2nd Workshop on Programmability Issues for Multi-Core Computers.
January 2009.

[2] Michael F. Spear and Luke Dalessandro and Virendra J. Marathe and Michael L. Scott

A comprehensive strategy for contention management in software transactional memory.

In: PPOPP~'09: Proc. 14th ACM SIGPLAN symposium on Principles and practice of parallel
programming. pp. 141--150. February 2009.