15-740 Project Proposal

Oct. 21, 2009

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

Project Web Page

http://www.cs.cmu.edu/~jdtam/15740.html

Project Description

In programs that use "obstruction-free" transactional memory, when a conflict is detected between multiple transactions, one is allowed to continue whereas others are aborted. This is called *contention management.* [1] surveys a number of contention management protocols for software transactional memory.

75%: We would like to explore how different benchmarks affect various contention management policies for software transactional memory. We plan on comparing our findings to those in [1] and analyzing the differences.

For example, we would like to study benchmarks available at the STAMP web page. The majority of benchmarks in [1] manipulate data structures, such as stacks, arrays , and various integer set implementations. The STAMP website features entire applications written specifically for transactional memory systems. We would like to analyze why the existing contention managers fare as they do on these different benchmarks.

We may need to write instrumentation code in order to collect exactly the data we want from our benchmark runs.

100%: We would also like to explore whether a different prioritization metric will have better throughput than those currently used (such as the metrics that the Karma and Eruption policies use: counting the # of accessed and acquired objects). *Example: Considering what type of memory the transaction wants to access, the length of time that a transaction has been "active", combination of these, etc.*

125%: See how our contention managers (with our best performing prioritization metric) scales with many more processors (>100).

Logistics

Plan of Attack and Schedule (A=Anvesh, S=Samir, J=Jenn)

Week 1: Complete all background reading of literature, including searching for more papers than those listed in this proposal, and start learning about simulators which includes downloading the software and making sure we have access to everything we need (A,S,J).

Week 2: Figure out which benchmarks we want to test and why (J), and decide which simulator to use (or use both listed in the resources section below). Go through source code and tutorials (A,S) so that we can test the benchmarks.

Week 3: Code and test the benchmarks (A,S). Begin analysis of data from benchmark tests (J).

Week 4: (Milestone) – Analyze the data from the benchmark tests (J+), figure out what prioritization metrics we want to test (A,S,J). Write up for milestone (A,S,J) and update website (J).

Week 5: Code (S+) and run (S+) new contention managers to reflect the new prioritization metrics. Start collecting statistics to analyze (A,J) for paper and web site

Week 6: Write up final paper (A,S,J) and poster (A,S,J), as well as update website (J)

Milestone
We plan to have completed most of the 75% work and be designing new contention managers with new prioritizations.

Literature Search

[1] Scherer, W. N. and Scott, M. L. 2005. Advanced Contention Management for Dynamic Software Transactional Memory. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing* (Las Vegas, NV, USA, July 17 - 20, 2005). PODC '05. ACM, New York, NY, 240-248. http://doi.acm.org/10.1145/1073814.1073861

[2] Chafi, H., Casper, J., Carlstrom, B. D., McDonald, A., Minh, C. C., Baek, W., Kozyrakis, C., and Olukotun, K. 2007. A Scalable, Non-blocking Approach to Transactional Memory. In *Proceedings of the 2007 IEEE 13th international Symposium on High Performance Computer Architecture* (February 10 - 14, 2007). HPCA. IEEE Computer Society, Washington, DC, 97-108. http://dx.doi.org/10.1109/HPCA.2007.346189

[3] Dolev, S., Hendler, D., and Suissa, A. 2008. CAR-STM: scheduling-based collision avoidance and resolution for software transactional memory. In *Proceedings of the Twenty-Seventh ACM Symposium on Principles of Distributed Computing* (Toronto, Canada, August 18 - 21, 2008). PODC '08. ACM, New York, NY, 125-134. http://doi.acm.org/10.1145/1400751.1400769

[4] Spear, M. F., Dalessandro, L., Marathe, V. J., and Scott, M. L. 2009. A comprehensive strategy for contention management in software transactional memory. In *Proceedings of the 14th ACM SIGPLAN Symposium on Principles and Practice of Parallel*

Programming (Raleigh, NC, USA, February 14 - 18, 2009). PPoPP '09. ACM, New York, NY, 141-150. http://doi.acm.org/10.1145/1504176.1504199

Resources Needed

We will be using Rochester's transactional memory simulator RSTM and potentially ATMTP which runs on top of Simics (Wikipedia) and GEMS. Benchmarks will be derived from RSTM and STAMP web pages.

Getting Started

We have read [1,2] and still need to read [3,4]. We have discussed the project with Professor Mowry who helped us decide which infrastructure tools we would need to complete our project. We do not currently have the software. We will need to obtain a license for Simics which CMU has. We can freely use the other pieces of software we will need.