**Adaptive Write-Update and Write-Invalidate Cache Coherence Protocols**

**for Producer-Consumer Sharing**

**Group Members:**

Bangjie Liu ([bangjiel@andrew.cmu.edu)](mailto:bangjiel@andrew.cmu.edu))

Hao Li ([haol2@andrew.cmu.edu)](mailto:haol2@andrew.cmu.edu))

**Project Web Page**

<https://lihao98722.github.io/15740>

**Project Description**

Cache coherence protocol has great influence on performance of shared memory multicore systems. Remote misses and bus traffics become the bottleneck of shared memory applications with high-performance demand. For this project, we focus on improving the performance of producer-consumer applications. The most popular cache coherence protocol used in modern multiprocessor architecture is directory-based write-invalidate protocol, which is inefficient for producer-consumer sharing due to extensive invalidation traffics and expensive remote misses.

In this project, we propose an adaptive cache coherence protocol optimized for producer-consumer sharing that reduces remote misses and communication traffics required to maintain coherence. It adaptively switches from write-invalidate to write-update on detecting producer-consumer sharing pattern, thereby converting remote misses to local misses and eliminating unnecessary invalidation traffics. Specifically, we will implement adaptive cache coherence as proposed protocol and directory-based write-invalidate as baseline protocol, and use cache simulator to compare latency, cache misses and bus traffics between proposed protocol and baseline protocol on the same runtime tasks in multicore systems.

**Goals**

* 100% goal: Implement proposed adaptive cache coherence protocols and producer-consumer sharing pattern detector. Evaluate the performance on producer-consumer applications. We expect to achieve a performance speed-up over the baseline.
* 75% goal: Implement cache simulators and testing tools (stack trace, logger, etc.) to evaluate baseline protocol on producer-consumer applications.
* 125% goal: Evaluate the performance of the proposed approach on general applications and verify its sequence consistency at our best efforts.

**Logistics**

Please address each of the following issues regarding how you will carry out your project:

Plan of Attack and Schedule: How will you go about completing your project? Please include a week-by-week schedule of exactly who in your group will be doing what. We will not strictly hold you to this schedule (other than meeting your milestone, as described below), but it is important that you have a concrete and realistic plan. Identify what you expect to be the critical path in your schedule. Indicate how the work will be divided among your group members.

**Milestone**

Indicate what your group plans to accomplish by Tuesday, April 11th (i.e., after four weeks). On that date, you will submit a brief report describing your progress up to that point. Your ability to set a reasonable milestone and to meet this goal will be factored into your final project grade.

**Literature Search**

[1] Cheng, Liqun, John B. Carter, and Donglai Dai. "An adaptive cache coherence protocol optimized for producer-consumer sharing." *High Performance Computer Architecture, 2007. HPCA 2007. IEEE 13th International Symposium on*. IEEE, 2007.

[2] Qureshi, Moinuddin K., et al. "Adaptive insertion policies for high performance caching." *ACM SIGARCH Computer Architecture News*. Vol. 35. No. 2. ACM, 2007.

[3] Danny. "The Disruptor - Lock-free Publishing." Codeaholics.org. http://blog.codeaholics.org/2011/the-disruptor-lock-free-publishing/, 28 June 2011. Web. 10 Mar. 2017.

**Resources Needed**

* Pin: <https://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool>
* Murphi: <http://seclab.stanford.edu/pcl/mc/mc.html>
* Stack trace: implement a stack trace tool which dynamically traces the sequences of multithread calls.

**Getting Started**

We have read multiple papers and articles on the joint research field of cache coherence and producer-consumer sharing pattern in multicore systems, and had two meetings with Prof. Beckmann discussing about the feasibility of research approach. We now have a clear idea of the current direction we should dig into: adapting write-update and write-invalidate cache coherence protocols according to sharing patterns. However, we don’t have a very good estimation of the project complexity and it is possible that the project scheme may be adjusted in the future due to unexpected difficulty. We will get started by researching on the sequential consistency criterions of alternating two protocols.