15-418/618 Final Project Proposal -- Directory Based Cache Coherence Simulator

Yinyi Chen
yinyic@andrew.cmu.edu

Ridhi Surana
rsurana@andrew.cmu.edu

Summary

We will be implementing a directory-based cache coherence simulator with various protocols including MSI, MESI, and MESIF. The simulator will take in memory reference traces from various programs and analyse/report their cache behaviors.

Background

Cache coherence is one of the most important topics in designing multi-processor caches. In the lectures, we discussed both snooping-based and directory-based cache coherence protocols. Comparing to snooping-based which relies heavily on broadcasting on the entire bus, directory-based protocols seems to be more scalable with regard to number of processors as it allows point-to-point communication. Therefore, we decide to develop a deeper understanding of the various directory-based cache coherence protocols by actually implementing them and observe cache behavior of programs with distinct memory traces. We hope that our project would eventually come available as a tool for programmers who are interested in knowing the cache behavior and memory reference characteristics of their programs, which could potentially be helpful in optimizing the code.

The Challenge

A. Since we are implementing a *cache simulator,* we must make sure that simulator is in sync with the protocol (MSI/MESI/MESIF) that the cache is using at every step of the way.
B. Checking for correctness - Currently, there is no harness available that we can use to check the correctness of our simulator, so we will have to build our own correctness evaluation harness. This might get particularly challenging with extremely large memory traces.

C. The Interconnect - In a snooping based protocol, we can have a single split-transaction bus. However, in a directory based scheme multiple nodes must be able to speak to each other simultaneously - designing a bus interconnect scheme that does this efficiently would need special attention.

Resources

Since we are developing a cache coherence simulator, we don't need any specific resources. Our simulator is meant to help improve the cache health of any parallel program so we would like it be resource agnostic.

Goals and Deliverables

Implementation

A. Implement directory-based cache coherence simulator with various protocols (MSI, MESI, etc.)
B. Take in memory traces of programs generated from Intel PIN tool and output 1) statistics like cache hits/misses 2) analysis regarding false-sharing and highly-contended cache lines

Analysis

A. Evaluate cache behavior of programs under different cache coherence protocols
B. Compare the evaluation result with that of using snooping-based cache coherence simulator

Stretch Goals

A. Implement the dragon protocol.
B. Create a graphical user interface for the simulator

Platform Choice

As mentioned before, we are developing a cache coherence simulator that is meant to run with any parallel program. Hence, are project is platform independent.

Schedule

Week #	Tasks
April 10th - April 16th	● Refresh ourselves on directory-based cache coherence and various protocols ● Initial Design ● Familiarize ourselves with Intel Pintool ● Develop the basic test and correctness harness
April 17th - April 23rd	● Implement a simple cache ● Implement the MSI and MESI protocol ● Start on developing an evaluation harness
April 24th - April 30th	● Implement MESIF protocol ● Detailed analysis of cache behaviour of test memory traces
May 1st - May 10th	● User-Friendly output explaining the cache behaviour of any input executable ● If goals achieved, work on stretch goals ● Project wrap-up