Multi-Core Cache Simulator with Different Cache Coherence Protocols

Haoyang Wang (haoyang4)
Zhuoyu Ji (zhuoyuj)

URL: https://github.com/why1998101/ParallelCacheSimulator

Summary

We plan to build a multi-core cache simulator which supports MESI, MESIF, and MOESI cache coherence protocols. We plan to perform experiments using the cache simulator to investigate how different protocols perform under different use patterns.

Background

MESI is a basic 4-state protocol to achieve cache coherence when there are multiple cores working in parallel. MESIF and MOESI are two different extensions of MESI that both add another state to achieve more efficiency.

Challenges

There are multiple major challenges in this project:

- The simulator tool we plan to use, ZSim, is quite complicated. We need to learn the basic framework and architecture of ZSim, especially modules that dictate the behavior of cache and memory subsystems.

- The ZSim tool only supports MESI protocol. We need to implement MESIF and MOESI protocols on our own by modifying the ZSim source code.

- In order to collect input to the cache simulator, we need to get a collection of real-world programs for benchmarking and memory access traces of those workloads.

- For measurement, we need a mechanism to verify the correctness of our MESIF and MOESI implementations. Moreover, we need to analyze the performance of different protocols under different memory access patterns after we wisely choose metrics to collect.

Resources

We plan to make use of existing mature frameworks and tools in the field of computer architecture. We decide to use SPEC as our benchmark suite (it would require access to SPEC). We decide to use Intel's Pin as our tool for instrumentation, or more specifically collecting memory access traces of programs. We will use ZSim to simulate the cache subsystem, implementing MESIF and MEOSI protocols on top of existing MESI protocol implementations.

References of Pin, ZSim, and SPEC are here:

https://software.intel.com/sites/landingpage/pintool/docs/98484/Pin/html/index.html

https://github.com/s5z/ZSim

https://www.spec.org/benchmarks.html#cpu

Goals and Deliverables

We have the following deliverables we plan to achieve:

1. The source code of a modified version of ZSim where MESI protocol is replaced by MESIF protocol.
2. The source code of a modified version of ZSim where MESI protocol is replaced by MOESI protocol.
3. A collection of input, output, and analysis that validate the correctness of implementations of MESIF and MOESI protocols.
4. A collection of input, output, and analysis that compare the performance of different protocols under different use patterns. (Validate that the 5-state protocols are more efficient than MESI in certain use patterns.)
5. A report on the project.

We have the following deliverables we hope to achieve:

1. A collection of input, output, and analysis that is related to some further exploration using the cache simulator. (We have not decided yet what to analyze beyond what is planned. We will modify this section and be clear about what extra analysis we can do after we successfully finish deliverables 1,2, and 3.)

Potential Demonstration Materials:

1. Metrics and analysis to demonstrate that our MESIF and MOESI implementations are correct.
2. Graphs and metrics to show different performance of different protocols under different use patterns.

The demonstration materials have to be easy to understand, supported by reliable data, and matching expectations and theoretical analysis. A good example of what kind of materials we want to show are those beautiful performance metric histograms in lecture slides.

Platform

We want to perform a realistic analysis between different cache coherence protocols. Therefore, using existing mature frameworks and tools is the best option. We find ZSim to be a powerful simulator that meets our need for this project and ZSim recommends to work with Pin. That's how we decide on our "platform" for this project.

Besides, as the team members use PCs with different architectures, it might be hard to configure the same virtual machine environment for development. Therefore, we decide to run our applications on AWS.

GHC and PSC machines are not considered right now as the team members do not have root access on those machines. This choice of running platform may change depending on the development progress.

Schedule

Date at End of Week	**Weekly Goal**
11.16	Study the ZSim and Pin tools, learn how to use them, and play with some inputs on the default MESI cache simulator
11.23	Study source code of ZSim related to cache coherence protocol; start to modify the MESI protocol to one of MESIF and MOESI
11.30 (Milestone)	Finish our implementation of MESIF and MOESI protocols. Find a naive method to validate those implementations
12.7	Collect inputs to formally validate our implementations of MESIF and MOESI
12.14	Collect inputs to do the performance comparison analysis
12.18	Do the additional analysis, prepare for demonstration, and finish up the project

Note:

We observe that the weekly goal from 11.23 to 11.30 might be the most difficult one to achieve. Given that the goals from 12.7 to 12.18 might not be time-consuming (at this moment, we have no accurate estimate of how long it will take to collect inputs to do all those validations and performance tasks). There is room for an adjustment: we may only finish the implementation of one 5-state protocol by 11.30.