Milestone Report

Haoyang Wang (haoyang4)
Zhuoyu Ji (zhuoyuj)

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

Completed Works

So far, we have checked the zSim and Pin tools in general. We have studied the zSim source code closely, especially the parts related to cache coherence implementations. We have started the implementation of the 5-stage protocols by modifying the zSim source code. Yet, we have made an important conclusion that implementing MESIF will be ineffective under zSim's cache architecture (explained below). At the same time, our implementation of MOESI is still in progress.

At this moment, there isn't any intermediate result to demonstrate.

Changes from Original Plan

The team finished goals for the first two weeks in the original schedule. But, for the third week, the team was behind schedule firstly due to some challenges in this phase and secondly due to workloads of other courses. (Original schedule is attached at the end.)

Besides, there was a major shift in plan as an outcome of the second week. As mentioned, after studying the zSim source code, we conclude that implementing MESIF under zSim will be ineffective. MESIF improves from MESI by reducing the coherence traffic when reading a shared cache line; however, this improvement is not effective if the cache architecture is not snooping-based; zSim's cache architecture is directory-based, which means the request to read will not be broadcasted globally; instead, there will be a recursive search performed within the zSim tree-like cache structure until one of the shared lines is found (assuming reading a shared line); in this way, implementing MESIF is basically meaningless and could only cause extra unnecessary works by introducing and constantly updating the F state. As a note, we have brainstormed how MESIF could fit under a directory-based cache but couldn't figure it out.

Given this conclusion, we have changed our goals and decided to only aim for MOESI implementation. Updated deliverables and a new detailed half-week schedule are shown below.

Challenges

The team didn't finish the MOESI implementation because of several challenges in the source code. To be honest, we are unsure if the MOESI implementation would be effective. We are questioning whether the given MESI implementation under zSim's cache architecture already has the major effect of MOESI. Yet, we are not confident enough to reach a conclusion with our current knowledge since we have not fully understood how zSim simulates internal data transfer such as memory flush or cache line forwarding.

As to other challenges, the team expects to need help in getting access to certain benchmarking tools for experimental evaluations. But as the team has not yet progressed to this stage, those challenges are not very clear right now.

Fortunately, we have reached out to Ziqi Wang. With the help of someone who knows better about the tools we are using, we expect to resolve some challenges after a meeting.

However, we have to mention that there is a possibility that our current doubt turns out to be valid, which means implementing MOESI is also ineffective under zSim. If this is real, the team will meet with the instructors as soon as possible to discuss alternatives as all original goals would be not achievable. A backup plan is to write a detailed report on zSim's cache architecture and analyze why implementing any of the 5-stage protocols is ineffective; in addition, experimental evaluations should still be performed to verify that existing zSim already shows performance with effects of the MOESI or MESIF protocol. We could also attempt other optimization directions that are more effective under a directory-based cache architecture such as optimizing the critical path.

Whatever the case is, the team is confident to finish a complete project, as members will be able to fully commit to this project starting from 12.9.

Note that the updated goals and new schedule below assume implementing MOESI is still effective.

Updated Goals and Deliverables

We will have the following deliverables:

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

We will demonstrate those in the poster session:

1. Metrics and analysis to prove that our MOESI protocol implementation is correct.
2. Graphs and metrics to show the performance differences.

The graphs we want to show would be similar to those performance metric histograms in lecture slides.

New Schedule

Date By	**Tasks**
12.3	Both: meet with Ziqi Wang on 12.2; resolve existing issues on source code

12.7	Both: finishing implementation of MOESI protocol
12.11	Both: fix apparent bugs in the source code if there is any Haoyang: collect inputs for correctness verification Zhuoyu: collect inputs for performance evaluation
12.14	Both: fix hidden bugs in the source code if verification test fails Haoyang: do correctness verification and collect results Zhuoyu: do performance evaluation and collect results
12.17	Both: finish up the project, write the report, prepare for poster session
12.18	Poster Session

Original Schedule for Reference

Date By	**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