15-418/618 FINAL PROJECT PROPOSAL: CacheM ALL Sim
Cache Memory Access Load Latency Simulator

Claudia Kho: **ckho**
Shivani Prasad: **sprasad1**

URL: https://metafly.github.io/CacheM_ALL/

SUMMARY

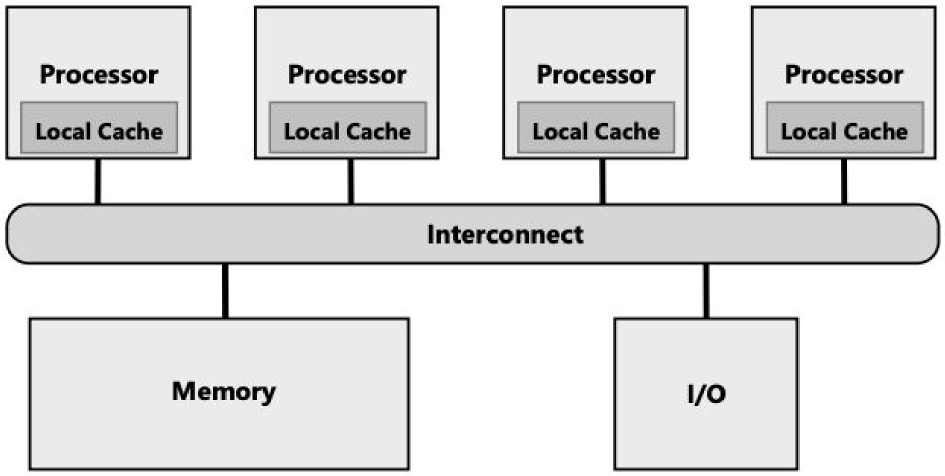
We plan on taking a previous 15-418 project, 418CacheSim, and modifying it by adding support for NUMA and mixed architectures. Our goal is to study the performance of different cache coherence protocols on different Memory Access layouts.

BACKGROUND

In computer architecture, cache coherence is the uniformity of shared resource data that ends up stored in the multiple local caches. These caches can be connected to memory in various ways:

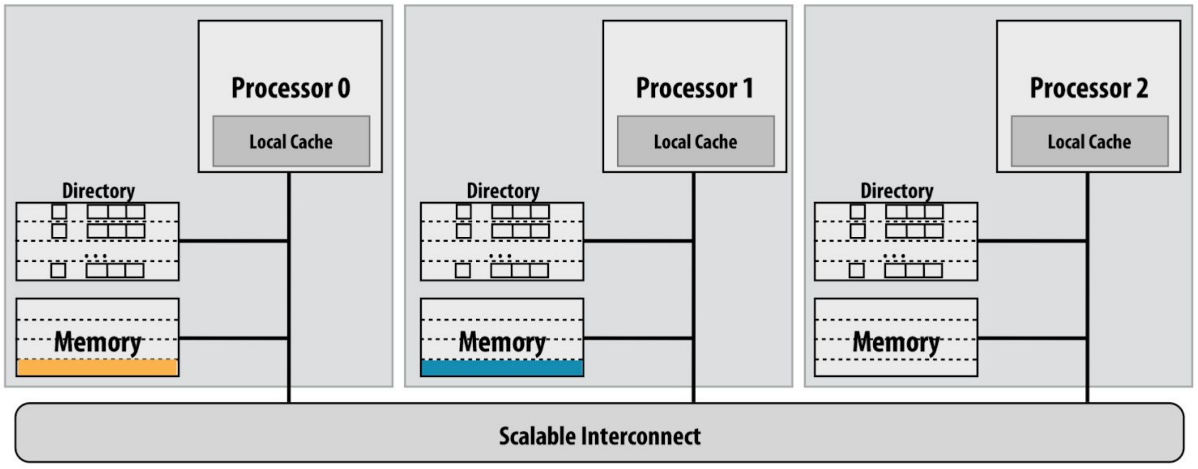
1) Uniform Memory Access (UMA)

In this form of connection, each processor in the model shares the physical memory uniformly. Hence access time to a memory location is independent of which processor makes a request

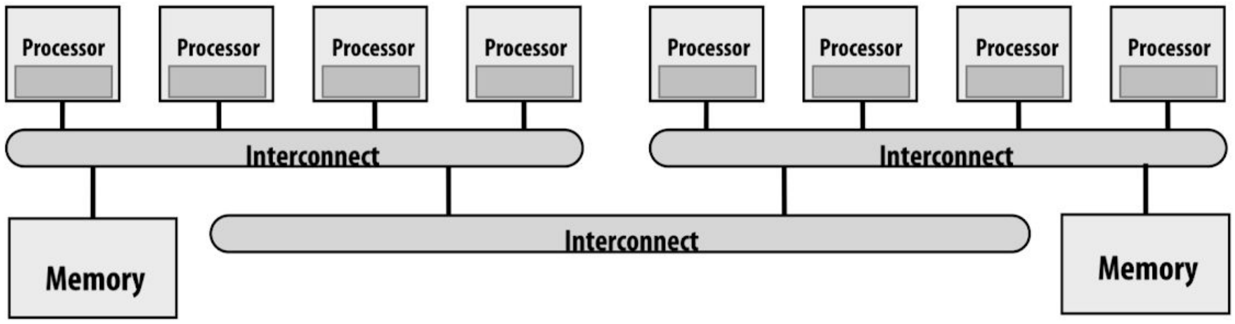


2) Non-Uniform Memory Access (NUMA)

In this form of connection, each processor's access time depends on the memory's location relative to the processor. Under this, a processor can access local memory faster than non-local memory.



3) **NUMA and UMA structures are often combined** to make scalable architectures. In this, processors in a Node have uniform access time to their local memory and have non-uniform access time to memories outside of their node depending on the distance.



We are trying to create a comparative study of cache coherence in different memory access layouts of nodes.

THE CHALLENGE

This project will have several challenges for us:

1) We need to understand how the 418CacheSim Repository works, and see whether we can just build a different memory access layout on that, or will we have to redefine how the simulator addresses various aspects of cache coherence.

2) We believe that timing is an important factor for cache coherence across NUMA. So, an important part of the project will be to decide a method to be able to calculate clock ticks. Or some other similar metric to calculate time. For example, measuring how much time does it take to service a request internally vs externally.

3) Deciding on what other metrics that should be compared across these architectures.

RESOURCES

We will be building off of a previous 15-418 project:

- 418CacheSim: https://kshitizdange.github.io/418CacheSim/proposal-page

The project provides a code base for snooping-based cache coherence protocols on basic UMA architecture.

Since we will be running the simulations with traces, we will not need any additional machines.

GOALS AND DELIVERABLES

Goals:

- Implement memory access models drawn in the background section: NUMA, and UMA+NUMA
- Add support for full bit vector directory-based cache coherence
- Take Memory Traces and run them with our simulator to generate statistics like avg service request time, internal service request time vs external request time, number of cache to cache transfers, number of snooping bus transactions for each bus, etc.

Performance:

- A number of bus transactions, per bus.
- Graphs comparing actual performance with theoretical performance.
- Avg. memory request serving time
- Memory contention

Extra stuff if we have time:

- Add different sparse directory-based protocols to the simulator
- Find additional metrics to compare things.

PLATFORM CHOICE

We will be using C++, since the project we are referring to was in C++. It also works well for reading memory traces and outputting the simulation results.

SCHEDULE

Week 1: October 29th - November 4th	● Meet Course Instructors Tuesday(10/29, 10:15 am) ● Decide on our project and submit a proposal (10/30, 23:59 hrs) ● Study the starter codebase
Week 2: November 5th - November 11th	● Implement NUMA ● Create way to calculate clock ticks
Week 3: November 12th- November 18th	● Implement UMA+NUMA architecture
PROJECT CHECKPOINT: November 18th	
Week 4: November 19th - November 25th	● Finish remaining implementations ● Generate memory traces
Week 5: November 26th - December 2nd	● Run traces and analyze performance results
Week 6: December 3rd - December 9th	● Work on the final report and project presentation
FINAL PROJECT DUE: December 9th	
PROJECT POSTER SESSION: December 10th	