Cache me outside, how 'bout dat!

David and Robbie

Description

Caching is the process by which data is stored in a cache so that the data can be served faster in the future. It is an integral part of modern computation and CPU specs often include the size of each level of cache. Our goal is to gain an understanding of how caching works and explore multiple caching implementations to evaluate their effectiveness. As our final deliverable, we will implement our own cache replacement algorithm for some specific task and analyze it using memory simulation software.

In the first phase, we will learn an overview of caching prior to diving into specific algorithms. This will involve understanding what architectural units decide on what to cache, what kind of memory is used for caching, what levels mean, and so on. We will also evaluate existing algorithms. We will focus on the more mathematically and algorithmically challenging ones, since some of the commonly used cache replacement policies such as LFU are very simple to understand and implement.

In the next phase, we will use the benchmark suite provided by ChampSim for the Cache Replacement Championship to implement our own cache policy and simulate its effectiveness. This will involve getting the suite to work, understanding the benchmarks, and learning how to write a policy. We will also add our own, specific test to the benchmarks and write a policy that we believe will work specifically for that test.

References

- Memory Systems: Cache, DRAM, Disk
- Wikipedia: Cache replacement policies
- Alasir: Functional Principles of Cache Memory
- http://crc2.ece.tamu.edu/ (Cache Replacement Championship website)

Deliverables

- A written report that conveys an understanding of the caching process
- Breakdown and analysis of the performance and tradeoffs of at least 3 common algorithms, with at least one of them being a complex non-intuitive one
- Create and implement our own caching policy and evaluate its performance using the simulation tools provided for the Cache Replacement Championship. Our caching policy might be a hybrid of existing ones or something entirely new and likely very bad. It might make sense and prove interesting to implement a policy that we believe is efficient for a niche purpose.

- A presentation that introduces the class to caching algorithms and includes a summary of our algorithm and simulation.

Work plan

- By Tuesday 11/28: Have an understanding of what caching is, what architectural units are responsible, what kind of memory it uses, and so on.
  (2 hours)
- By Friday 12/1: Read through README for the Cache Replacement Championship suite. Run our first benchmark simulation.
  (1.5 hours)
- By Monday 12/4: Have an understanding of at least 3 caching policies and their strengths and weaknesses.
  (2 hours)
- By Friday 12/8: Design our own caching policy (but do not implement).
  (2 hours)
- By Sunday 12/10: Implement non-intuitive caching policy (specific policy TBD) to better understand its implementation. Run benchmarks for evaluation. Compare the benchmark to how we expected this policy to perform.
  (2 hours)
- By Monday 12/11: Implement our own algorithm (time permitting) and flesh out any details we missed earlier. Run benchmarks for evaluation.
  (5 hours)
- By Tuesday 12/12: Write report, prepare presentation.
  (3 hours)