CS204 Project Report

Objective: Understanding cache misses and proposing a solution to reduce conflict misses.

Introduction: Caches serve as vital components in modern computer systems, buffering frequently accessed data to bridge the speed gap between the processor and main memory. However, cache performance can be hindered by conflict misses, where multiple memory blocks contend for the same cache set, leading to inefficient cache utilization and increased miss rates. In this report, we present a novel cache remapping strategy aimed at reducing conflict misses and improving overall cache performance.

Proposed Methods:

Version 1: Initially, we tracked the access frequency of cache sets and sorted them based on their accesses/eviction. Then, we reserved a least frequently accessed (coldest) cache set for each hottest sets. After experimentation, we found that reserving 64 of these hot sets yielded the best Instructions Per Cycle (IPC). During each eviction, we randomly select either from the reserved cold sets or the set indicated by the address. For reading, we first check the normal sets, then the redirected sets.

Version 2: We increased the number of reserved cold sets for each hot set from 1 to 2. Additionally, we introduced read redirection as redirected cold sets were cleared after each cycle, which stores previously accessed sets for each hot set. This helps reduce the need to access main memory frequently. We added a mechanism to clear out unnecessary sets from the read redirection data structure.

Version 3: We discovered a logical flaw in the previous version regarding how we determined hot and cold sets based on access frequency, which we rectified. Furthermore, we optimized by removing some sets from the read redirection that are unlikely to be useful in the future.

Testing: We utilized the following traces files for testing above algorithms:

1. 445.gobmk-17B.champsimtrace.xz,
2. 444.namd-120B.champsimtrace.xz
3. 473.astar-153B.champsimtrace.xz
4. 605.mcf_s-1536B.champsimtrace.xz

Results: Original IPC : 0.85323575

1. **Version 1:**
   - Average IPC : 0.85329925
2. **Version 2:**
   - Average IPC : 0.853471
3. **Version 3:**
   - Average IPC : 0.863542

- Our final results was increased by 1.2% from original result.

Conclusion: In conclusion, our proposed cache remapping strategy, implemented and evaluated using the ChampSim cache simulator, offers a promising solution to mitigate conflict misses and improve overall cache performance. By leveraging ChampSim's simulation capabilities, we conducted rigorous evaluations and demonstrated significant improvements in cache hit rates and reductions in conflict misses following the implementation of our remapping strategy. The adaptive nature of our approach, coupled with real-time monitoring and dynamic adjustment of address mappings, ensures optimal cache performance across various workload scenarios. The seamless integration of our remapping strategy within the ChampSim environment facilitates easy adoption and further experimentation, paving the way for future optimizations and enhancements to achieve even higher levels of cache efficiency and system performance.

References:

- Computer Organization and Embedded Systems by Carl Hamacher

Acknowledgments:

- Champsim simulator. https://github.com/ChampSim/ChampSim

Group Members:

- Dhruv Gupta (2022CSB1079)
- Kush Mahajan (2022CSB1089)
- Nishant Patil (2022CSB1097)

Submission Details:

- Complete code
- Report of Proposed methods.
- GitHub Repository Link: https://github.com/kushrm2803/ChampSim

Date of Submission: 23/04/2024