CS204: Project

Objective:

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

Release date: March 27, 2024
Preliminary submission: April 16, 2024
Final Deadline: April 23, 2024, EOD

Points: 100
Weightage 10%
Group size: 3. Group can be different or same as mini-project group

Problem statement:

Given a fixed configuration of last level cache (2MB, 16 ways, 64 byte block size) and for input traces, find a suitable remapping of address to line such that conflict misses could be reduced. A reduction in conflict misses will improve hit rate, therefore, AMAT, and therefore will improve IPC and performance.

It is observed that fixed address to line mapping using modulo operator, can cause under-utilization of cache and since some sets are less utilized while other sets are heavily utilized. One possible solution of this problem is to create an intermediate data-structure that helps us to remap an address to a new index, if required. For example, a two dimensional array ([#Sets][#ways+1] can be used to represent a remap table, initialized with default set values. Example of 4 ways 8 set cache is shown below:

```
{{ 0  0  0  0  0},
 { 1  1  1  1  1},
 { 2  2  2  2  2},
 { 3  3  3  3  3},
 { 4  4  4  4  4},
 { 5  5  5  5  5},
 { 6  6  6  6  6},
 { 7  7  7  7  7} }
```

Let's say set 1 is over utilized and set 6 is under utilized, then a few blocks of set 1 can reside in set 6, remap table can look like

```
{{ 0  0  0  0  0},
 { 1  1  1  1  6},
 { 2  2  2  2  2},
 { 3  3  3  3  3},
 { 4  4  4  4  4},
 { 5  5  5  5  5},
 { 6  6  1  1  6},
 { 7  7  7  7  7} }
```

Classifying cache sets as Hot or Cold based on access and eviction counts is crucial for optimizing cache performance. After completing execution for a given trace, you can categorize sets into Very Hot, Hot, Very Cold, and Cold.

For each set, you are expected to track the number of accesses, evictions, and assign a category based on these metrics:

1. Very Hot Sets: Sets with exceptionally high access and eviction counts.
2. Hot Sets: Sets with high access and eviction counts.
3. Very Cold Sets: Sets with exceptionally low access and eviction counts.
4. Cold Sets: Sets with low access and eviction counts.

Note:

1. **You're expected to set the thresholds for above categorization of sets based on your observations of accesses and evictions per set.**
2. **The simulation has to be done with Warmup and Simulation for 100 Million instructions each.**

Stage 1 (Preliminary submission):

Preliminary submission consists of your novel way of remapping blocks.
Performance will be monitored by IPC values.
Overhead will be measured by the average number of tag comparisons for one memory search.
Submission: Complete code base as zip file. Average IPC and overhead for four given traces.

Stage 2: IPC challenge competition

Second phase will be an open competition where all the groups will compete for best IPC and least overhead.

1. Name your team.

2. Submit your IPC score in the below google form:
3. Every time you are able to upgrade your IPC score, you need to submit this score again, and maintain this version
4.

Submit your IPC score in the following google form:

https://forms.gle/QLUUGyFBQmqHXvTs6

Final Submission: A zip file that contains

- A report of your proposed method
- All versions of your work, along with results (if you have submitted five times on google form, there has to be five versions in zip file

Scoring:

- 10% for initial + correct submission
- 70% for final submission
  - Individual marks for individual team members
- 20% would depend on
  - IPC achieved by you
  - If you are in top performing teams
  - Number of attempts/submissions in google form
  - Overhead
- Negative marking for false claims
- Negative marking for plagiarism

Traces for evaluation:

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

Note: The traces have to be used in compress format only

Traces download link:
https://drive.google.com/drive/folders/1IxbyLJ30uXWaxK7uy_84BHyr5WjIC8Qu?usp=sharing

Champsim Simulator:

1. Create a new folder Name it: **dpc3_traces** and put the downloaded trace files in it.

2. For granting permissions, run for both **build_champsim.sh** and **run_champsim.sh**

    a. chmod +x script.sh example : **chmod +x build_champsim.sh**
    b. sed -i -e 's/\r$/\n/' script.sh example: **sed -i -e 's/\r$/\n/' build_champsim.sh**

3. For building: **./build_champsim.sh bimodal no no no no lru 1**

4. For running: **./run_champsim.sh bimodal-no-no-no-no-lru-1core 1 10 gcc_13B.trace.xz**

5. For code change,
    1. Look at replacement folder
    2. Look at src folder: mainly **cache.cc**
    3. Look at inc folder: mainly **cache.h, block.h**

Champsim download link:
https://drive.google.com/file/d/1RgT3E2AUDhLu2mxse0W6U7rG4hcGFfCx/view?usp=sharing

FAQ

We will add FAQs as we get queries related to project.