Report

Group Number: 23

Hemlata Gautam-2022csb1084

Smriti Gupta-2022csb1128

ChampSim Project

ChampSim is a microarchitecture simulator designed to facilitate the exploration of CPU cache hierarchies and memory systems. It allows us to simulate multiple levels of caches (L1, L2, L3, LLC) with varying configurations. We can set cache sizes, associativities, block sizes, replacement policies, and more with the help of this simulator.

OBJECTIVES:

* Reducing the conflict misses to improve hit rate, therefore, AMAT, and therefore improving IPC and performance.
* To modify the allocation process of the sets to the data by the processor.
* To implement lru and srrip replacement policies.

STEPS TO EXECUTE THE PROGRAM-

1. Download the zip files, the user is requested to add a folder named “dpc3\_traces” in ChampSim and add traces in it in zipped from only through the link:

https://drive.google.com/file/d/1\_jnGbQwFSXyVrh5qkXTMtMCOB5DRwA3f/view?usp=sharing

2. To set the permissions. Write the following commands:

--> chmod +x build\_champsim.sh

--> sed -i -e 's/\r$/\n/' build\_champsim.sh

-->chmod +x run\_champsim.sh

3. Running commands-

Using lru replacement policy:-

./build\_champsim.sh bimodal no no no no lru 1

./run\_champsim.sh bimodal-no-no-no-no-lru-1core 1 10 nameoftrace.xz

Using srrip replacement policy:-

./build\_champsim.sh bimodal no no no no srrip 1

./run\_champsim.sh bimodal-no-no-no-no-srrip-1core 100 100 nameoftrace.xz

in command 2,

"1" stands for first 1 Million instructions of trace file are for warmup,

"10" stands for next 10 Million instructions of trace file are for simulation

We can change 1 and 10 to any number, only thing is it will be in "millions" of instructions.

CHANGES MADE:

* We have created a map of int and vector<int> called as remapped.

>> map<int, vector<int>> remapped;

It stores the sets mapped to each set in a vector corresponding to each set number.

* A header file named “modifications.h” is made which contains the declarations of all the new variables used.

OUR SOLUTION:

* To define the thresholds for a set to be categorised as very hot, hot, cold and very cold, we have created a vector named “heat”.

>> vector<int> heat[4];

* In the functions, “handle\_fill()”, “handle\_writeback()” and “handle\_read()” we have updated the values of “set\_Misses\_LLC” and “set\_Hits \_LLC” of each set whenever any hit or miss occurs.
* For all these functions, we have created new utility functions which provide the correct set and way number from the sets mapped to the current set.
* These new functions return a pair of set and way number.
* A new function is created named “find\_hot\_cold\_sets()” which categorises the sets as very hot, hot, cold, and very cold using a vector of tuple. First, the tuples are sorted by the number of misses, then these are categorised into two parts and again sorted by the number of hits.
* Finally, the sets are categorised as follows:

- (high miss and low hit)- very hot;

- (high miss and high hit)- hot;

- (low miss, low hit)- very cold;

- (low miss, high hit)- cold.

* A function named “remap\_LLC()” is made which remaps the most used (very hot and hot) sets to the lesser used (cold and very cold) sets.
* The remapping of sets will be started after simulation of (no. of simulations/5) instructions.
* “find\_hot\_cold\_sets()” and “remap\_LLC()” functions are called in “operate()” function to categorise the sets and remap them according to the procedure.
* In the file “llc\_replacement.cc”, “llc\_find\_victim()” is made to return the victim set and way in a pair.
* In the file “base\_replacement.cc” a new function named “lru\_victim\_llc()” is introduced to find the victim according to the newly mapped sets as now the required data can be in any of the mapped sets.

CONCLUSION-

The average IPC values obtained after remapping are:

* Preliminary – 1M warmup and 10M simulations : 0.81103225
* IPC challenge – 100M warmup and 100M simulations: 0.863253