Advanced Multiprocessor Architecture

Programming Assignment 1

Due on Monday, Esfand 18th, 1399

Overview

In this assignment, you will write a cache simulator and evaluate the LRU replacement policy using traces of CloudSuite, PARSEC, and SPEC workloads. You are free to choose the programming language of your choice among C/C++, Java, Perl, PHP, or Python. You are advised to start early.

Programming Exercise

Code a generic cache component which is parametrized by (1) Cache Block Size, (2) Cache Associativity, and (3) Cache Size. The cache component, among other things, should have a 'lookup' method which gets an address and searches the cache to find the address. If the address is in the cache, this lookup is a hit. If the address is not in the cache, it is a miss. In this case, the block that contains the address should be inserted into the cache. If all of the frames in the set that the block address maps to are occupied, the cache should pick a victim according to the replacement policy, evict the victim block, and insert the block that holds the address.

Write an interface that reads addresses from the input trace files and apply them to the cache using the 'lookup' method.

Note 1: You can find the format of the trace files in the 'readme.txt' file which is delivered to you along with the trace files (in 'traces/readme.txt').

Note 2: Please ignore the PC field of trace files in this programming assignment.

Milestone

For every workload, compare the LRU replacement policy against Belady's optimal replacement policy for a 2 MB, 16-way associative last-level cache with a 64-byte block size.

Note: You can find useful information about any of the mentioned replacement policies on the Web (Google is your best friend!).

Deliverable

Hand in the code and a short report (PDF) that describes the replacement policies and their observed performance (i.e., hit ratio). How much better is the best replacement policy as compared to the best practical one?

Please put your code and report in separate folders named *code* and *report*, respectively, place both folders in another folder named *pa1_firstname_lastname*, zip the upper folder and upload it into LMS.