Computer Architecture

Homework # 4 (Cache Replacement)

1 Objective

This project is to help you understand how Re-Reference Inverval Prediction (RRIP) cache replacement policy works. You are to implement the **Static** RRIP Hit Priority (SRRIP-HP) policy for shared L3 caches in zsim; and compare its performance to other policies.

2 System Requirement

Linux operating system and x84 processor is needed in order to use the pre-compiled x86_64 ISA SPEC CPU2006 and PARSEC binaries. **Do not use Cygwin**. If you don't have any linux machine, please use linux.cse.tamu.edu with your CSE account. If you don't have CSE account, contact HelpDesk located in the third floor.

3 Procedure

Implement a SRRIP-HP cache replacement policy in zsim. Run simulations to compare the performance to other policies using SPEC CPU2006 on single-core and PARSEC benchmarks on multicore processor. Follow exact same procedure as HW2 for setup.

3.1 Run the following benchmarks using ZSim

Table 1. SPEC CPU2006 Benchmarks (Single-threaded)

Integer	*bzip2, gcc, mcf, hmmer, sjeng, libquantum, xalancbmk*
Floating Point	*milc, cactusADM, leslie3d, namd, soplex, calculix, lbm*

Table 2. PARSEC Benchmarks (Multi-threaded)

blackschoels, bodytrack, canneal, dedup, fluidanimate
freqmine, streamcluster, swaptions, x264

You will run an representative simpoint of each SPEC CPU2006 benchmark for 100 million instructions. For PARSEC benchmarks, simulations will run the whole parallel phase.

1. Use `hw4runscript` to run the benchmarks

   Note: The config files and runscript must be in the `casim/zsim`.

   `$ ./hw4runscript <suite> <benchmark> <repl_policy>`

   Example: `$ ./hw4runscript SPEC bzip2 LRU`

2. Check the results in `outputs` directory (`zsim.out`). In `<benchmark>.log` you may see one of the following errors.

3. **Possible error:** A: Source/pin/injector_nonmac/auxvector.cpp: CopyAux: 291: unexpected AUX VEC type 26

- Add the following lines in file src/pin_cmd.cpp after line 52
- args.push_back("-injection");
- args.push_back("child");
- args.push_back("-ifeellucky");

4. **Possible error:** E: 4.4 is not a supported linux release
   - Add the following lines in file src/pin_cmd.cpp after line 52
   - args.push_back("-injection");
   - args.push_back("child");

If you are running zsim in linux.cse.tamu.edu, be sure you are not monopolizing computational resources on the machine. **Do not run more than 1 instance at a time in linux.cse.tamu.edu.** It is violation of section 3.3 of the Appropriate Use of Computer Science Computing Resources Policy, located here: https://engineering.tamu.edu/cse/cse-internal/computing-resources

Don't run more than one instance of any benchmark simultaneous in the same machine. It may cause errors. **Run one instance at a time per benchmark.**

4 Assignment

4.1 Reading

1. "High Performance Cache Replacement Using Re-Reference Interval Prediction (RRIP)", Aamer Jaleel, Kevin Theobald, Simon C. Steely Jr, and Joel Emer. *In International Symposium on Computer Architecture (ISCA)*, Saint-Malo, France, June 2010.

4.2 Guideline

RRIP finds a victim block that is not recently used in a set for cache replacement. When replace a new block in the cache, its recency value is not set to *near-immediate* re-reference interval (Most recently used) or *distant* re-reference interval (Least recently used). Instead, it is set to a long re-reference interval between the two extremes. Please refer to the paper for details.

4.3 Design and Implementation

In `casim/zsim/src/`, implement SRRIP-HP algorithm in `rrip_repl.h`. You can refer to LRU implementation in `casim/zsim/src/repl_policies.h`. After your implementation, add SRRIP instantiation in `casim/zsim/src/init.cpp` for system initialization. (search for SRRIP in the file). Re-compile the simulator and run simulations.

In `init.cpp`, we read `rpvMax` from the configuration file, which is the max value of RRPV. You need to pass this value to your SRRIP constructor. In this assignment, you need to simulate a 2-bit SRRIP-HP configuration, where `rpvMax = 3`.

4.4 Comparison

Compare the performance of three different cache replacement algorithms: LRU, LFU, and SRRIP on shared Last-Level Cache (LLC). Use `hw4runscript` to run different schemes and benchmarks.

We have provided the config files for LRU and SRRIP, check the difference between them. You need to create config files for LFU on your own. Copy the config files from `casim/configs/hw4/LRU` to `casim/configs/hw4/LFU` and change the parameters for `repl.type` and `outputDir`.

4.5 Results and Report

In this assignment, you will run SPEC benchmarks on a single-core processor and PARSEC benchmarks on multi-core processor. Read the configuration files for SPEC and PARSEC, describe the system configuration in your report for two systems, respectively.

In your report, you need to plot visualized results for Misses per Kilo Instructions (MPKI) of LLC and performance results. Also provide detail analysis on your results.

5 Tutorial

You can get some additional information regarding HW4 in tutorial slides:
http://zsim.csail.mit.edu/tutorial/slides/memory.pdf

6 Turning Instruction

1. Make all your files including modified source codes, simulation results and the report (`rrip_repl.h`, `init.cpp`, `casim/zsim/outputs/` directory and report) into one zipped file. **We accept zip files only.** If you send a different file format, you may receive 0 point for the assignment.

Your report must contain simulation results (You should include zsim log files in the zipped file, but **dont'** put the whole log in your report.) and analysis of them. Any result you consider important can be used. Only **PDF** is acceptable for the report.

2. Submit your zipped file through **CANVAS**. Your file name should be **hw4_#UIN.zip**.

3. Penalty of late submission: 5% deduction per day. Contact Teaching Assistant if you have any questions.