CSE 520 Computer Architecture -- Spring 2019
Programming Assignment 2 (100 points)

In this assignment, you will implement a cache replacement policy, i.e., SHiP (Signature-based Hit Predictor), for the last level cache, and compare its performance with the built-in LRU and RRIP cache replacement policies of gem5.

Similar to RRIP (or static RRIP), SHiP considers re-reference interval and makes prediction for each cache line when it will be re-referenced in the future. It learns the re-reference behavior of cache lines belonging to each signature of memory region or program counter. Please read the papers about RRIP [1] and SHiP [2] to understand the idea of re-reference interval based cache replacement.

The SHiP policy to be implemented should be based on memory region signature, i.e., the most significant bits of the data reference address, and should be applied to RRIP policy. The SHiP mechanism is equipped with a 16K Signature History Counter Table (SHCT) of 3-bit saturating counters to learn the re-reference behavior of a signature.

You will implement the replacement policy in a patched gem5. The modification in the patch is to pass data reference address in the object of "ReplacementData". Then, you can access Signature History Counter Table based on reference addresses. We will post a link soon for downloading this patched release.

To debug your implementation, you can utilize the debug flags support of gem5. Please refer to http://learning.gem5.org/book/part2/debugging.html for details.

To expose the cache replacement policies as command line option, you should follow the steps described in perf_gem5_instruction.pdf document. The calling name of each policies in gem5 are strictly defined in the parenthesis. That is, when configuring gem5 to use SHiP replacement policy in L2 cache, you will give the flags as --l2_rpp="SHiPRP()". In your implementation of SHiP, the class name of the replacement policy should be set to "SHiPRP()".

For performance comparison, you need to consider the following cache replacement policies for L2 cache.

- LRU (LRU, built-in)
- RRIP (Re-Reference Interval Prediction) [1]
- SHiP (Signature-based Hit Predictor) using the SRRIP [2]

The following CPU configuration in gem5 is used during this assignment.

- AtomicSimpleCPU
- L1 cache
  - 32 KB for instruction, 32 KB for data
- L2 cache
  - 256 KB, 1M, and 4M
  - 16-way set associative
  - Replacement Policies LRU, RRIP, and SHiP
- All other settings remain in default value.

You will run your gem5 with two workloads (BFS_opt and MST_opt) for studying different L2 cache replacement policies. The inputs to the benchmark programs are in benchmark_assign2.zip where you can find two graph inputs for BFS and other two for MST. So, in total, you need to report 36 cache miss

ratios, i.e. from 2 benchmark programs x 2 inputs x 3 replacement policies x 3 l2 cache sizes. The estimated simulation time for each run should be less than 30 minutes. Note that you may choose some small graphs from the benchmark inputs of assignment 1 when testing your replacement policy.

In the report, please give a brief introduction to all 3 cache replacement policies and their key difference, followed by the table and graph showing **overall miss rate of L2 cache** collected from your simulation results. Then, you will provide your observation and thought on how these cache replacement policies affect the performance in various workloads.

[1] Aamer Jaleel, Kevin B. Theobald, Simon C. Steely, Jr., and Joel Emer. 2010. "High performance cache replacement using re-reference interval prediction (RRIP)", ISCA 2010, https://people.csail.mit.edu/emer/papers/2010.06.isca.rrip.pdf.

[2] Carole-Jean Wu, Aamer Jaleel, Will Hasenplaugh, Margaret Martonosi, Simon C. Steely, Jr., and Joel Emer. 2011. "SHiP: signature-based hit predictor for high performance caching", MICRO-44 (http://mrmgroup.cs.princeton.edu/papers/MICRO11_SHiP_Wu_Final.pdf).

Due Date

The due date of the assignment is 11:59pm, Feb. 27.

What to Turn in for Grading

1. Create a working directory, named "cse520-assgn02-LastName_FirstInitial", for the assignment to include
    a. A pdf document with all tables, graphs, and answers from this assignment. Don't forget to add your name and ASU id in the document.
    b. A patch of all your changes in gem5's cache replacement policy directory gem5/src/mem/cache/replacement_policies/. It is likely that you will add source code files for SHiP replacement policy, and modify the files Tags.py and SConscript.
    c. A readme text file with all the commands you use. Alternatively, you could have a script file with comments inside on how to use it.
2. Compress the directory into a zip archive file named cse520-assgn02-LastName_FirstInitial.zip. Note that any object code or temporary files should not be included in the submission. Submit the zip archive to the course Canvas by the due date and time.
3. There will be 20 points penalty per day if the submission is late. Note that submissions are time stamped by Canvas. If you have multiple submissions, only the newest one will be graded. If needed, you can send an email to the instructor and TA to drop a submission.
4. The assignment must be done individually. No collaboration is allowed, except the open discussion in the forum on Canvas.
5. ASU Academic Integrity Policy (http://provost.asu.edu/academicintegrity), and FSE Honor Code (http://engineering.asu.edu/integrity) are strictly enforced and followed.