Department of Electrical and Computer Engineering

The University of Texas at Austin

EE 382N.1, Fall 2018

Instructor: Dam Sunwoo

Teaching Assistant: Pritesh Vinod Chhajed

Graduate Lab Assignment

Due: Dec 10th @ 11:59 pm [No slack days for this lab]

You may do this lab in teams of 2 and we recommend you take advantage of this.

Introduction

For this assignment, you will evaluate the Re-Reference Interval Prediction (RRIP) cache replacement policy. The experiments will be done with ChampSim, a trace-based simulator.

RRIP

The common LRU replacement policy performs well for a near immediate re-reference interval on cache hits and misses. Therefore, workloads with distant re-reference interval exhibit poor performance under LRU. Jaleel et al. [1] proposed cache replacement using Re-reference Interval Prediction (RRIP) to improve the performance of such workloads.

In this lab assignment, you will evaluate the **Static and Dynamic RRIP (SRRIP / DRRIP)** cache replacement policies using gem5. The RRIP paper can be found here.

[1] A. Jaleel, K. Theobald, S. C. Steely, and J. Emer, “High Performance Cache Replacement Using Re-Reference Interval Prediction (RRIP),” in ISCA-32, 2010.

ChampSim

ChampSim is a trace-based microarchitecture simulator and was the simulator platform of choice for the 2nd Cache Replacement Championship. You can find the source code of the simulator here: https://github.com/ChampSim/ChampSim

Read the README page to get familiar with how to build and run the simulator. ChampSim will use traces that are generated with Pin (see below).

Evaluation

1. Build SPEC Benchmarks

You will use the SPEC CPU 2006 benchmark suite to evaluate the effect of the RRIP cache replacement policies and compare against LRU. Specifically, you will be running the following five benchmarks as used in the RRIP paper: **bzip2**, **cactusADM**, **hmmer**, **mcf**, and **sphinx3**.

You may download the SPEC CPU benchmarks from Canvas (look for “SPEC2006_cd.tar.gz” in Files). Instructions for building the SPEC benchmarks can be found one the official SPEC CPU 2006 website (note that the SPEC CPU version we provide is v1.1). For this assignment, you will use the x86 architecture. Use the **Reference input sets** for the benchmarks. For benchmarks with multiple input sets (bzip2 and hmmer), it is okay to choose one. Rough instruction counts for each benchmark and input can be found in Figure 1 in this paper. Use this information as guide to select inputs and to estimate SimPoint and trace generation time. To use each benchmark in the following flows (SimPoint and trace generation), you will need to extract the command lines and input files to run the benchmarks.

2. Generate SimPoints and Traces

In order to save time, we suggest that you use the simulation points technique instead of running the full SPEC benchmarks. SimPoint [2] is a statistical method to identify representative segments of an application. In a paper titled Simulation points for SPECCPU 2006, Nair et al. [3] show that applying this methodology to the SPEC CPU 2006 benchmarks, they can save time by collecting data from representative segments of the workload instead of executing the whole workload.

Use PinPoints to generate SimPoints for your SPEC benchmarks. PinPoints can be found here: https://software.intel.com/en-us/articles/pin-a-binary-instrumentation-tool-pinpoints. README.PinPoints has some examples on generating SimPoints using PinPoints.

Once SimPoints have been identified, use the ChampSim tracer to generate traces at those SimPoints. See “How to create traces” under the ChampSim documentation. Make sure you generate them using the right length (as specified in your SimPoint flow).

To save time for this lab, you are allowed to use as few as 3 SimPoints per benchmark and each SimPoint can be as short as 30 million instructions. Adjust the “slice_size” and “maxk” parameters to set your SimPoint length and count. More and longer SimPoints will increase your accuracy. Feel free to try more and longer SimPoints if you have time and resources. Please be aware that SimPoint and trace generation can take a long time (several days depending on benchmark).

[2] Sherwood, T., Perelman, E., Hamerly, G., and Calder, B. 2002. Automatically characterizing large scale program behavior. In Proceedings of the 10th international Conference on Architectural Support For Programming Languages and Operating Systems (San Jose, California, October 05 - 09, 2002). ASPLOS-X. ACM, New York, NY, 45-57.

[3] Nair AA, John LK, "Simulation points for SPECCPU 2006", Computer Design 2008, ICCD 2008, Pgs 397-403.

3. Run traces in ChampSim

ChampSim already has implementations for LRU, SRRIP, and DRRIP. After running the traces, the simulator will produce some basic statistics. Compare the performance across different replace policies and different configurations. Be sure to include the metrics you used and your rationale for including them in the slides. Feel free to add more statistics that can help with your analysis.

Submission

You will submit a set of PowerPoint slides of your study.

Your slides should include the following information:

- Brief summary of the RRIP cache replacement policy
- SimPoint setup
  - Number of SimPoints, length of SimPoints used
  - List of identified SimPoints and weights for each benchmark
- Quantitative evaluation
  - Important metrics for comparing cache replacement policies & rationale
  - Methodology (system configurations, how you collected data)

    - Your evaluation data for RRIP vs. LRU
        - different cache sizes
        - different associativity
        - s-curve (refer to Figure 9 in the RRIP paper)
        - effect of prefetch vs. no-prefetch
- Analysis
    - Discuss your observations, insights from the data
    - Conclude and make recommendations

The slides should be self-explanatory -- enough text to understand, but still look like slides (no paragraphs and no fonts smaller than 16pt (ideally no smaller than 20pt).

Submission format

You will submit your slides on Canvas:

- A ppt or pdf file containing your slides (ee382n_gradlab_eid1_eid2.pdf)