CS 6354: Graduate Computer Architecture Fall 2022

Championship Value Prediction

Begins: Mon, Sep. 26 **Ends:** Wed, Oct. 26 *(45 points)*

Introduction

This contest is based on the Championship Value Prediction held in conjunction with ISCA 2018. The goal of the championship is to design a value predictor that provides the best improvement in IPC (Instructions Per Cycle). You will be provided with a set of execution traces (generated from real workloads) and a trace-based simulator with an interface to plug in your value predictor. The trace-based simulator will take the execution trace as input and simulate an out-of-order superscalar processor with a 3-level cache hierarchy. With the value predictor enabled, you should be able to observe a speedup (improvement in IPC) at the end of the simulation.

Setup Instructions

The skeleton code and the execution traces we will use to evaluate your predictor can be found on the CS department server `portal.cs.virginia.edu` at `/p/csd/CVP.tar.gz`. The only two files that you will need to modify and turn-in are `predictor.h` and `predictor.cc`. While you may test your predictor on Gradescope, it is much faster and more convenient to test it on a local machine. You may compile the simulator using the Makefile provided as part of the tarball for compiling the simulator. The simulator itself can be run using the following command line.

```
./cvp -v -t 1 <trace-name>
```

Scope of the Championship

- You're only required to predict the results of *load* instructions.
- The predictor will also have a budget, namely, 16 KB (including all predictor tables and history registers). That may not be the amount of storage your trace-based simulator uses, however – for example, you may implement 2-bit predictors with a table of ints, in which case the simulator will use more memory – that's okay, we're only concerned about the memory used by the simulated value predictor.

Turn-in Instructions

You will turn-in only `predictor.h` and `predictor.cc` to Gradescope, which will feature a leaderboard that will be updated periodically. All predictors will be ranked based on the amount of IPC improvement they provide against a baseline that features an EVES predictor with an 8KB fixed budget. You are also required to turn in a report, no longer than 2 pages. The report **must** include the following, in addition to a description of the design details of your predictor.

- A detailed block diagram of your predictor such as Figure 3 from the H3VP paper.
- A detailed budget justification table such as Table 2 and Table 3 from the H3VP paper.

Grading Criteria

1. 36 points for beating the baseline (8KB EVES)
2. 42 points for beating our implementation of a 16KB EVES predictor
3. Upto 6 additional points will be allocated based on your standing on the leaderboard (reaching a maximum total of 48/45). The exact formula for points allocation will be decided at the end of the championship, based on how the class has been performing.
4. There will also be a penalty of upto 6 points if your predictor exceeds the budget of 16 KB

Collaboration and Plagiarism Policy

You are free to use and tweak code of the winning predictors from CVP-1 (e.g., EVES and H3VP). You are also free to brainstorm ideas outside of your group. However, when it comes to coding, please make sure to restrict all collaboration only within your team.