CSE220 Lab 3 Check-In

Andrew Rivero, Sean Li

November 2024

Overview of Progress

Our progress in implementing this branch prediction algorithm consists of: Modifying graphing function to enable graphing of benchmark tests against each other, creating new flags for running specific bp algorithm, creating rough bp algorithm files, and verifying algorithm functionality.

In-Depth Analysis

To implement this branch prediction, we will reference the algorithm described in *Two-Level Adaptive Training Branch Prediction*.

Following this paper, we created the branch prediction algorithm using one of the algorithms referenced. For future optimizations, we would like to integrate multiple algorithms, and have them choose an ideal prediction based on majority rule. Additionally, we would like to optimize each algorithm to run faster, as at this stage they run un-optimally.

To implement the algorithm in Scarab, we referenced cbp files, as they implement different bp algorithms. Following their format, we modified cbp_table.def and cbp_to_scarab.cc to run our algorithm using the ”–twolevel 1” flag in our .json file.

The benchmark algorithms we will be testing against–branch always taken and never taken–were generated in an identical manner to our implementation of two-level bp. However, when creating these algorithms, we noticed that the number of BTB_ONPATH_MISS' were nearly identical between them, only differing by ∼5. From this revelation, we opted to choose only one of the algorithms to test against, to demonstrate that it has less overall MISSES.

The graphing function was additionally modified to showcase multiple different algorithm results. This is done by searching through different folders for algorithm results, then plotting them. The graphing function was also modified such that differences in values were more visible. This was achieved by setting the yscale to logarithmic.