Jared Raiola

CS4290

6/2/20

Project 1 Design Results

The purpose of the second half of project 1 was to find the best branch predictor for each trace in the traces directory following the constraints of having a total budget of 4096 bits and achieving the lowest possible misprediction rate. The way I approached this problem was using the brute force method of testing every predictor by hand and performing manual changes to each parameter. While this is not the most time efficient method, I chose this direction because I believed I could better determine trends between the predictors parameters and how they affected misprediction rate for a greater overall understanding of the branch prediction concept and it allowed me the widest range of motion to manipulate the parameters to any value efficiently.

Methodology:

My plan was simple, to test each predictor in order with the highest parameter combinations available to use up the most storage overhead available and then decrease parameters as necessary. The logic behind this was the idea that the more a predictor could store, the higher its accuracy rate would be because it would have the ability to choose from a variety of possibilities. From that point, I picked the lowest misprediction rate, and began decreasing parameters in favor of increasing num_entries to find the lowest misprediction rate possible. This process is complete because the higher the storage overhead, the more information being stored, the larger the history available to predict the next branch, meaning we do not have to test all possibilities, just the ones that provide the most information turnover.

Design Results:

Listed below are the highest values that I was able to find per trace for every predictor. Number 1 for each trace file is the highest performing predictor for that trace file.

- h264ref.trc:
  1. G-Share with 1024 num_entries, 2 counter_bits and 10 history_bits returned a misprediction rate of 5.0543% and storage overhead of 3082 bits.
  2. Bimodal with 2048 num_entries and 2 counter_bits returned a misprediction rate of 5.1041% and storage overhead of 4096 bits.
  3. Local History with 512 num_entries, 3 counter_bits and 1 history_bits returned a misprediction rate of 5.5842% and a storage overhead of 3584 bits.
  4. Two-Level Adaptive with 512 num_entries, 4 counter_bits, 4 history bits returned a misprediction rate of 6.4950% and a storage overhead of 2112 bits.
- bzip2.trc:
  1. Bimodal with 1024 num_entries and 4 counter_bits returned a misprediction rate of 3.5179 % and storage overhead of 4096 bits.
  2. G-Share with 1024 num_entries, 3 counter_bits and 10 history_bits returned a misprediction rate of 3.5327 % and storage overhead of 3082 bits.
  3. Local History with 512 num_entries, 3 counter_bits and 1 history_bits returned a misprediction rate of 3.56293 % and storage overhead of 3584 bits.
  4. Two-Level Adaptive with 512 num_entries, 5 counter_bits and 4 history_bits returned a misprediction rate of 4.0667 % and storage overhead of 2128 bits.
- gcc.trc:
  1. G-Share with 2048 num_entries, 2 counter_bits and 11 history_bits returned a misprediction rate of 5.7527% and storage overhead of 2058 bits.
  2. Bimodal with 2048 num_entries and 2 counter_bits returned a misprediction rate of 5.8324% and storage overhead of 4096 bits.
  3. Local History with 256 num_entries, 2 counter_bits and 2 history_bits returned a misprediction rate of 5.8735 % and storage overhead of 2560 bits.
  4. Two-Level Adaptive with 256 num_entries, 3 counter_bits and 4 history_bits returned a misprediction rate of 6.4892% and storage overhead of 1072 bits.