

Seung Woo Lee

Project 2 Experiment Report

CS 4290

03.19.23

Our goal for the experiment was to find the optimum (highest accuracy) Tournament branch predictor. Misprediction rate is an important factor in computing misprediction stalls as the rate is a scalar factor for the number of misdirection stalls per instruction which can be seen in this formula:

$$\text{misdirection stalls} = \text{branch misprediction rate} * \text{number of branch instructions} * \text{branch misprediction penalty}$$

Since misprediction rate is $1 - \text{branch prediction accuracy}$, we can use this number to analyze how “accurate” the Tournament branch predictor is. Also, as everything in the expression is a formula is a scalar and not in a rate in time, we can assume that we can use arithmetic means for the misprediction rate. Even though the misprediction rate is $1 - \text{branch prediction accuracy}$, it is still legal to use the arithmetic mean as rearrangement of the expression would be safe due to the linear characteristic of arithmetic mean.

The predictor also had several constraints, where it came down to the total size of both Yeh-Patt and Perceptron be at least 6KiB while not exceeding 24 KiB total. From this constraint, there were 1192 possible configurations of the Tournament table. As we had four different benchmarks, we found the optimum Tournament branch predictor by obtaining the misprediction rate for each benchmarks for each configuration, getting an arithmetic mean out of them, and selecting the configuration with the lowest mean misprediction rate. The configuration with P: 11, L: 12, N: 9, G: 35, C: 1 had the lowest mean misprediction rate, which was 0.02873015, and thus was the optimum Tournament branch predictor.

Some interesting finding was that the optimum branch predictor for each benchmark was slightly different. The configuration with P: 11, L: 12, N: 9, G: 35, C: 1, which was our chosen optimum branch predictor overall had the lowest misprediction rate for GCC and Perl, but configurations with P: 14, L: 12, N: 8, G: 35, C: 1, and P: 15, L: 11, N: 8, G: 47, C: 1 had the lowest misprediction rate for MCF and x264 respectively. However, when we look at other benchmark values of the two other configurations, there were considerably more difference in the Perl misprediction rate than the overall optimum branch predictor. For the MCF benchmark, the optimum configuration had $\sim 2 * 10^{-5}$ lower misprediction rate than the configuration with P:14, and for the x264 benchmark, the optimum configuration had $\sim 9 * 10^{-3}$ lower misprediction rate than the configuration with P:15. However, the Perl benchmark values were lower than the optimum configuration by $\sim 1 * 10^{-2}$ for the configuration with P:14 and up to $\sim 1.5 * 10^{-2}$ for the configuration with P:15. These differences made the optimum configuration to have the lowest misprediction rate arithmetic mean for all four benchmarks.