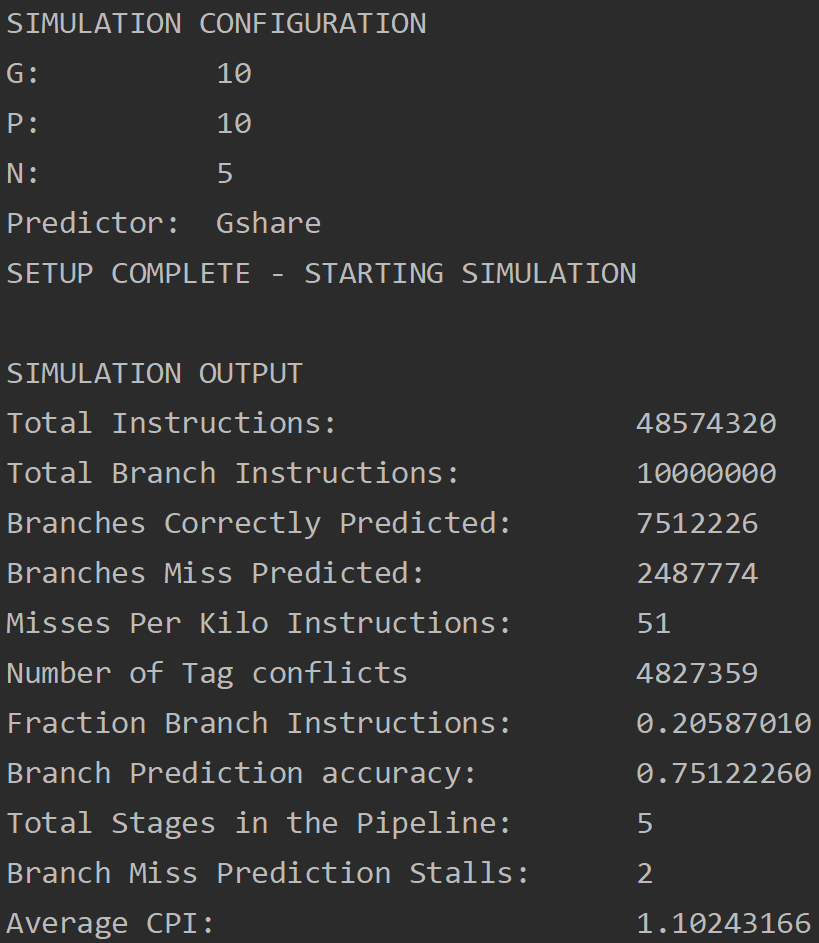
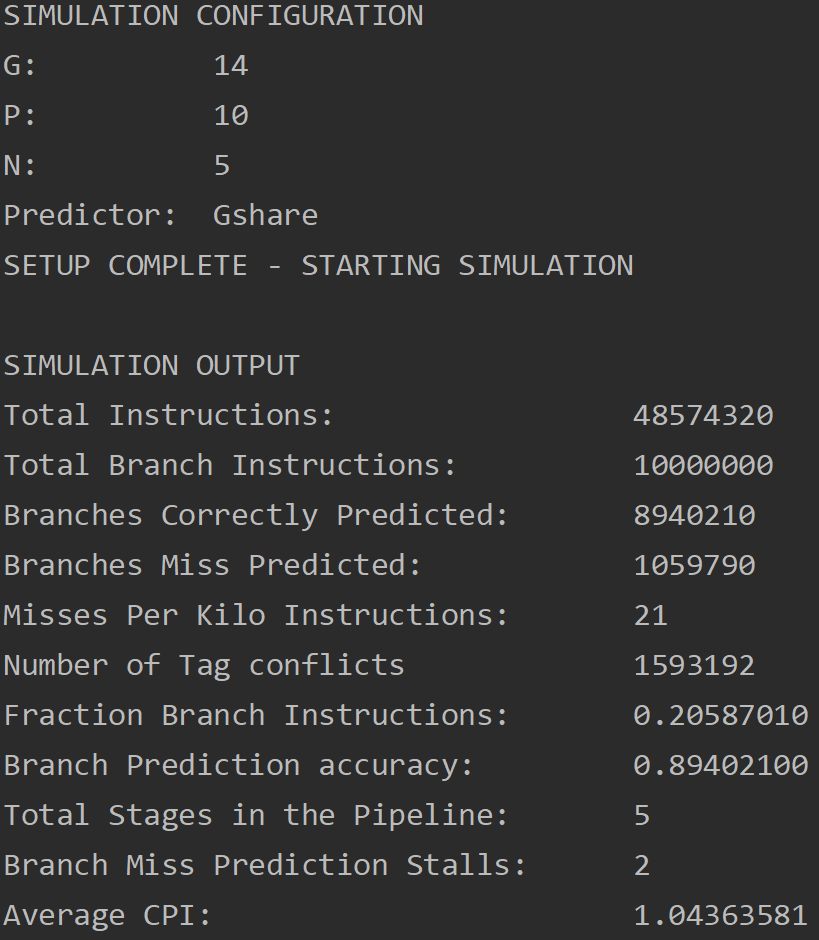
# Experiment Project 2

## Introduction

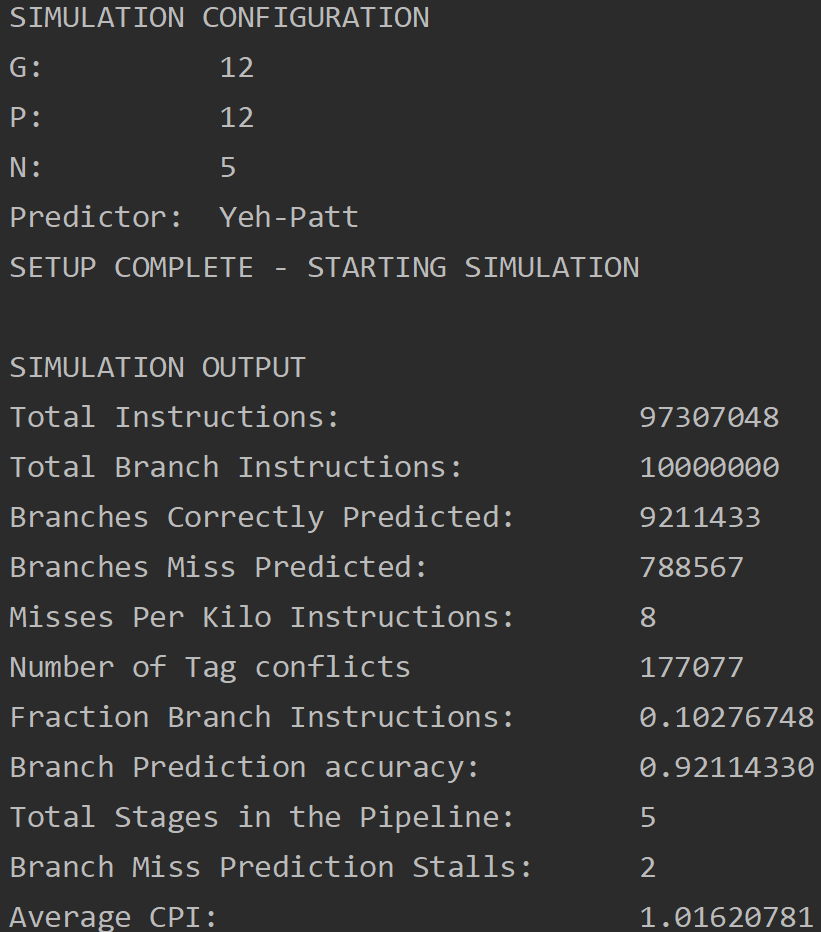
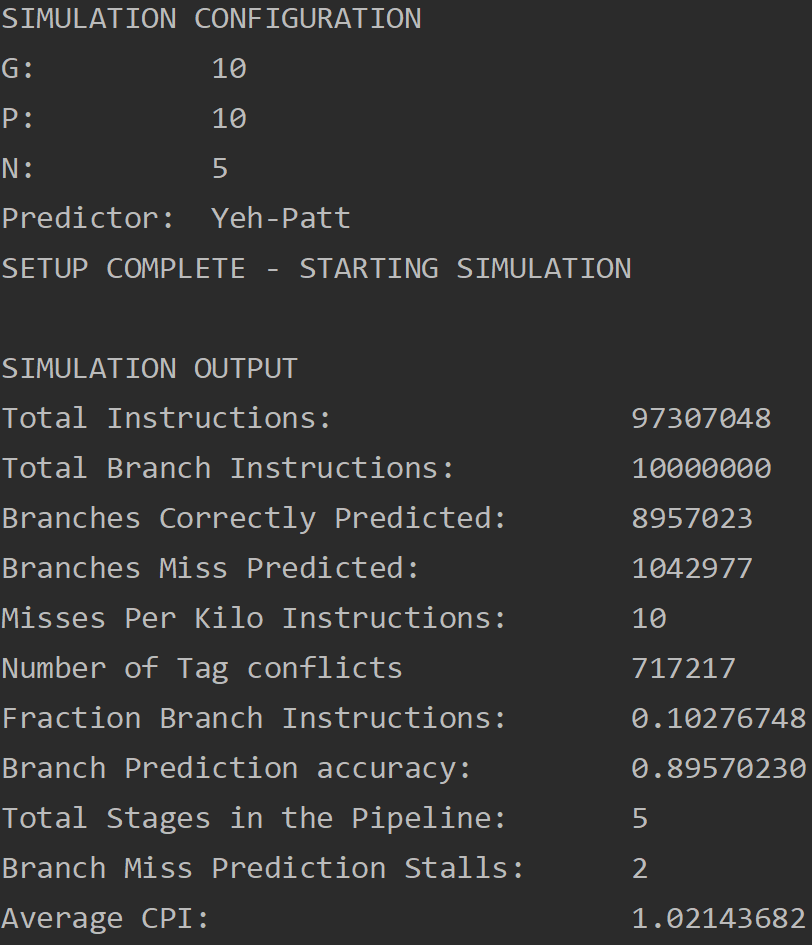
In this part of the project, we are supposed to find the best accuracy of predictors for different workload. To find the optimal configuration, we have to use the tricks the same way we did for experiment 1.

## Process

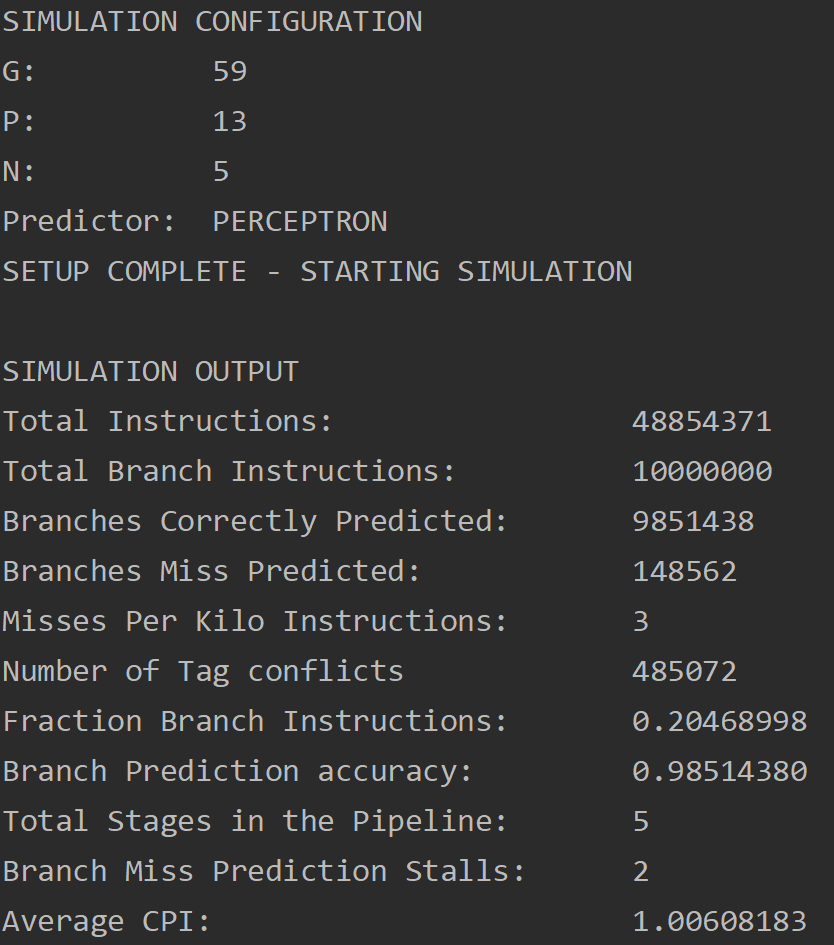
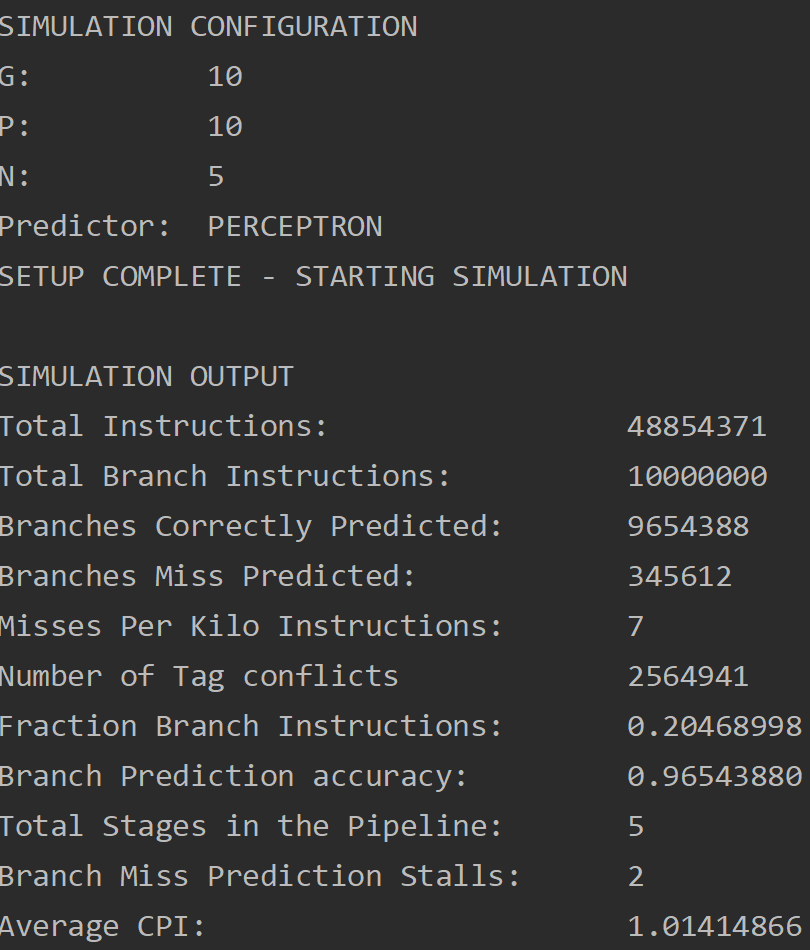
TO start, I choose the basic configuration provided by the constructor. The accuracy that I got was about 79% which petty high for the >1Kb bit budget (10 + 2^ 11 = 2058 bits). So, second thing was to use this and see if increasing the bit budget for GShare improves the accuracy or not. After careful calculation I realize that I can only increase the GShare bit budget to max of 4KB(it goes little over if I went 8KB) which is G= 14, and it improve the accuracy greatly. The average accuracy that I got was about 93%.



Then I tried similar pattern for yeh pat predictor. So, I tried the biggest possible configuration available to us. G = 12 and P= 12 (bits = (P\*2^G) +(2^P+1)), which adds up to total of 57344 bits or 7KB which improved accuracy litter bit. On average I got 94% as my accuracy for the workloads. Which is not so much, considering the amount of bits that we are using.

Similarly, I tried maximum capacity configuration for perceptron, G=59 P=13 (total bits = 7\*2^13 + GHR) which is approximately 7 kb. This is the maximum size available in all three predictors. The average accuracy that I got form these huge weights are about 97% which is way higher than the yeh pat. On similar note, perceptron works way better than any predictor on basic settings.

## Results

As suspected, bigger the predictor better its performance. Also. The perceptron performance way better than other two algorithm in a small budget. So, I choose this configuration for the given workloads. Since we have 8KB of bit budget.

**G = 59 , P=13, N=5, predictor = perceptron**