Gianluca Bonanno

Taken goes through the input and compares to “T” every time. When there is a match it increments the total number of taken branches.

Not taken goes through the input and checks if the input is anything that isn't “T”. When there is a mismatch, it increments the total number of not taken branches.

Bimodal has counters for page sizes of 16,32,128,256,512,1024, and 2048. It also has arrays of integers for each page size. The array holds a 0 for not taken and 1 for taken. If we index into the array by using the address module the page size and find a 1 and the input file is taken, then we increment the correct predictions by 1. If we find a 0 in the array and the input file says it is not taken, we also increment the correct predictions by 1. We also update the array at the current index with the value 1 if it was taken and 0 if it was not taken.

Bimodal 2 bit functions in the same way as bimodal except that the array now stores a Predictor struct instead of an array. Rather than being 1 or 0, the array now stores a Predictor struct which has 2 int values. Taken and Strong. Taken = 1 when taken and taken = 0 when not taken. Strong = 1 if it is a strong prediction and Strong = 0 when it is a weak prediction.

The tournament predictor provided the best performance. It had the benefits of both gshare and bimodal.

The optimal configuration of the best performing predictor should have a smaller gshare history but larger array sizes, as close to the number of instructions in the program as possible. That way every branch has its own index in the arrays. That way you won't have misses when indexing into the array.