CSE 240A Project 1 – Branch Predictor Contest

Yihong Zhang

PID:A53097346

The predictor is based on the model of piecewise linear, which unifies the perceptron-based and path-based neural predictor.

The algorithm has two components: a prediction function and an update procedure. The following variables are used by the algorithm:

W: a three-dimensional array of integers. The indices of this array are the branch address, the address of a branch in the path history, and the position in the history. W keeps track of correlations for every branch in the program. W[B,0,0] is the weight that keeps track of the tendency of branch B to be taken. This weight is the bias weight for B. Addition and subtraction on elements of W saturate at +127 and -128. The dimensions of the array are arbitrarily large, i.e., large enough to accommodate any access that might be made during the algorithm.

The indices of the W array must be limited to keep them from exceeding the practical bounds of an implementable branch predictor. I limit the first two indices by taking them modulo two integers n and m. In a realistic implementation, n and m would be chosen as powers of 2 to make the modulo operation a simple mask. We limit the third index by choosing an appropriate value h for the history length. Thus, W becomes an n*m*(h+1) 3-dimensional array of 8-bit weights. For the tuning of predictor, when n=m, it has better results.

h: The global history length. This is a small integer.

GHR: The global history register. This vector of bits accumulates the outcomes of branches as they are executed. Branch outcomes are shifted into the first position of the vector.

GA: An array of addresses. As branches are executed, their addresses are shifted into the first position of this array. Taken together, GHR and GA give the path history for the current branch to be predicted.

output: An integer. This integer is the value of the linear function computed to predict the current branch.

Overall, in my program, I choose h=15, n=m=16. The optimal threshold is based on “Piecewise Linear Branch Prediction”[3], which is 2.14(h+1)+20.58. In my test cases, when threshold=33 the predictor has the optimal accuracy. The total size of tables I use is under the hardware budget, which is:

h+log2(m)*h+n*m*(h+1)*8=32843(bits) < 33024(32K+256)(bits).

I compare the piecewise linear predictor with Alpha 21264[4], 2-level local prediction, global/local perceptron (GLP) and global perceptron (GP). From the following result it is clear to conclude that the piecewise linear predictor has the optimal average predict accuracy.

Trace	Alpha21264	2-level	GLP	GP	PWL
FP-1	3.357	4.147	2.023	2.454	3.071
FP-2	1.315	2.311	1.14	1.12	1.074
INT-1	8.587	14.08	6.27	7.451	6.244
INT-2	11.865	13.619	11.171	10.327	7.683
MM-1	8.918	10.813	7.7	7.68	7.434
MM-2	10.851	14.404	9.927	10.047	9.065
SERV-1	9.775	10.06	8.764	7.789	6.678
SERV-2	10.138	10.153	9.207	8.32	7.202
AVG	8.10075	9.948375	7.02525	6.8985	6.056375

References.

[1] D. Jimenez, C. Lin, Dynamic Branch Predictor With Perceptrons, HPCA, 2001

[2] D. Jimenez, C. Lin, Neural Methods for Dynamic Branch Prediction in ACM TOCS, 2002

[3] D. Jimenez, Piecewise Linear Branch Prediction, ISCA, 2005

[4] Kessler, The Alpha 21264 Microprocessor, IEEE Micro, 1999.