Assignment 1 - 2019CS10399

1 Predictor

For the 2400, 6400, 9999 bit budgets, a tournament predictor was implemented. The two predictors used were a `PAp` predictor and a `GAp` predictor. The choice predictor was a saturating counter which increments or decrements based on which predictor is correct and which is incorrect. For all three of the predictors, instead of concatenating n bits of the address with the k bit `GHR`, a mangling function was used as follows:

$$\left(\frac{(big + small) \times (big + small + 1)}{2} + small\right) \&((1 << numBits) - 1)$$

For different power budgets, different table sizes were used. Average accuracy obtained was as follows (m is the number of bits used to address the `GHR` table, k is the number of previous branches stored, n is the number of bits used to address the `PHT`, g is the number of bits used to address the `GPT`, `c` is the number of bits used the address the `CPT`):

1. 2400 bit budget: 2356 bits were used to obtain an (avg) accuracy of 96.98580124895354%
   - Parameters used are: m = 5, mOffset = 3, k = 5, n = 10, pht_bits = 2, g = 4, gpt_bits = 3, c = 5, cpt_bits = 3
2. 6400 bit budget: 6152 bits were used to obtain an (avg) accuracy of 97.41073325632129%
   - Parameters used are: m = 8, mOffset = 5, k = 6, n = 10, pht_bits = 3, g = 8, gpt_bits = 3, c = 8, cpt_bits = 3
3. 9999 bit budget: 9865 bits were used to obtain an (avg) accuracy of 97.63366804224047%
   - Parameters used are: m = 7, mOffset = 4, k = 5, n = 11, pht_bits = 3, g = 9, gpt_bits = 3, c = 9, cpt_bits = 3

1.1 32000 bit Budget

For the 32000 bit budget, a perceptron predictor was implemented with the modification that a `GPT` was used to retrieve the k bit `GHR` instead of using a completely global `GHR`. The `PerceptronTable` stores the weights for each of the that are multiplied with the history to make a prediction.

The parameters (weight_bits is the number of bits used for each weight value, threshold is the value that is used to decide whether further training is needed or not [based on perceptron output]) used

are m = 3, mOffset = 12, k = 60, n = 6, nOffset = 3, weight_bits = 8, pht_bits = weight_bits * (k + 1), threshold = ((1 « (weight_bits - 1)) * (k + 1)) * 0.0135.

The average accuracy obtained is 98.61938734273128% using 31720 bits.

2 Machine Learning Predictor

A perceptron based classification algorithm was used. For most of the traces, predictions are skewed towards not taken (which is the default bias for any branch prediction). Following accuracies and confusion matrices were obtained for each of the 5 traces:

2.1 Trace 1

Accuracy = 86.71140678862687%

	Predicted 1	Predicted 0
Actual 1	257086	151049
Actual 0	139605	1637797

2.2 Trace 2

Accuracy = 87.91069628086742%

	Predicted 1	Predicted 0
Actual 1	252316	111787
Actual 0	103196	1310992

2.3 Trace 3

Accuracy = 97.88601128897839%

	Predicted 1	Predicted 0
Actual 1	1330039	6910
Actual 0	25254	159281

2.4 Trace 4

Accuracy = 95.81091670010764%

	Predicted 1	Predicted 0
Actual 1	790762	11317
Actual 0	26159	66373

2.5 Trace 5

Accuracy = 57.30949291281886%

	Predicted 1	Predicted 0
Actual 1	836908	544078
Actual 0	478538	535931

2.6 Average Accuracy

The average accuracy of the ML algorithm is 85.125704794%

3 Working of Tejas

There is a shared memory channel between the PIN execution of the executable and Tejas, from which Tejas collects relevant information. The PIN has some events defined which on triggering cause corresponding simulator functions to be called which then log relevant information.

3.1 Information Stored by Tejas

3.1.1 Basic Information

1. Memory configuration
2. Row buffering policy used
3. Scheduling policy used
4. Queuing structure used

3.1.2 Core Specific Information

1. IPC
2. Number of micro-ops
3. Number of CISC instructions
4. Number of branches
5. Predictor accuracy (TAGE predictor is used by default)
6. Memory Related-
    a. Number of memory requests
    b. Number of loads
    c. Number of stores

d. LSQ forwardings
e. Instruction TLB hits, misses, accesses, hit-rate, miss-rate
f. Data TLB hits, misses, accesses, hit-rate, miss-rate
g. Instruction cache misses, hits, accesses, hit-rate, miss-rate (for I1 cache)
h. Data cache misses, hits, accesses, hit-rate, miss-rate (for L1, L2 cache)
i. Shared (data) cache misses, hits, accesses, hit-rate, miss-rate (L3 cache)

3.1.3 RAM Information

Infomation about number of reads and writes for each channel, its ranks and the banks of each ranks are given.

3.1.4 Energy Information

The leakage energy consumed, dynamic energy consumed and the number of dynamic accesses for each component of each core and the shared components is given.