Branch Prediction Simulation Results

Steve Jia

Purdue Northwest - Hammond

CS59000 – 10: Computer Architecture  
Fall Semester 2017

Professor: Dr. Ruijian Zhang

Abstract

This report presents the branch prediction simulation results and analysis based on said results. The simulation performed branch prediction using a static predictor and three two-level dynamic adaptive predictors (Pap, Gap, and GShare). Each predictor generated results based on different setup parameter values.

Branch Prediction Simulation Results

The branch prediction simulation project is mainly based on research done by Yeh and Patt in 1991. The simulation program mainly uses three different versions of the two-level dynamic adaptive training predictors, but also include a static predictor as a reference. The simulation program is written in C with the help of CodeLite IDE, and it is compiled and executed using a version of the GCC compiler running on a branch of Arch Linux operating system.

# Simulation Results

Predictor correctness will be discussed here, as well as cost for the three two-level branch predictors.

## Static Predictor

The static predictor used in this simulation always predicts “non-taken”. Out of 2,684,455 lines of branch instructions, the static predictor only predicted 48% correct. This means that more than half of the global branch results are “taken”.

## Two-Level Predictor: PAp

The PAp two-level predictor was simulated several times, each time with a different setup. The two setup parameters, number of entries in the Branch History Table (BHT) and number of bits for each Branch History Register (BHR), range from 512 entries to 8192 entries for the BHT, and from 2-bits to 8-bits for the BHR. The PAp predictor was implemented using the Least-Used algorithm when hashing the index of the BHT. If the current branch address is not found in the BHT, the program will find the least used entry and replace it with information for the current branch instruction.

Prediction accuracy ranges from 77% to 94%, and the cost ranges from 0.64Kbytes to 266Kbytes. With 2048 entries for the BHT and 4 bits for the BHR, the PAp predictor reached 91.985% correctness.

## Two-Level Predictor: GAp

The GAp two-level predictor was simulated several times, each time with a different setup. The two setup parameters, number of Pattern History Tables (PPHT) and number of bits for the Global Branch History Register (GBHR), range from 512 entries to 8192 entries for the PPHT, and from 2-bits to 8-bits for the GBHR. The index for the current branch instruction is simply hashed using the mod function and a simple replacement policy for collisions was used.

Prediction accuracy ranges from 69% to 93.9%, and the cost ranges from 0.51Kbytes to 262Kbytes. With 2048 entries for the BHT and 4 bits for the BHR, the PAp predictor reached 86.796% correctness.

## Two-Level Predictor: GShare

The GShare two-level predictor was simulated several times, each time with a different setup. The two setup parameters, number of Pattern History Tables (PPHT) and number of bits for the Global Branch History Register (GBHR), range from 512 entries to 8192 entries for the PPHT, and from 2-bits to 8-bits for the GBHR. The index for the current branch instruction is hashed by XOR’ing the current branch instruction address and the number of PPHT, and a simple replacement policy for collisions was used.

Prediction accuracy ranges from 65% to 95.5%, and the cost ranges from 0.51Kbytes to 262Kbytes. With 2048 entries for the BHT and 4 bits for the BHR, the PAp predictor reached 84.696% correctness.

# Result Analysis

Overall, the PAp two-level branch predictor is the best, in terms of prediction correctness with an affordable cost. With 2048 entries for the Branch History Table and 4-bits of Branch History Register, the PAp predictor can reach 91.98% correctness while the GAp and the GShare can only reach 86.80% and 84.70%, respectively. However, if cost is disregarded, the GShare predictor has the highest percentage correctness, at 95.53%.

Between the GAp and the GShare predictors, although the two predictors are very similar, the hashing function doesn’t seem to pay-off for the GShare predictor until the number of Pattern History Tables has increased above 2K entries. Therefore, with smaller entries sizes, the GAp predictor is better, and the GShare predictor is better for larger setups.

References

Yeh, Y. T., Patt, Y. N. (1991). Two-Level Adaptive Training Branch Prediction. *Proc. 24th Ann. Int'l Symp. Microarchitecture*, Pages 51-61.

Yeh, Y. T., Patt, Y. N. (1992). *Alternative Implementations of Two-Level Adaptive Branch Prediction.* *19th Ann. Int'l Symp. on Computer Architecture*, Pages 124-134.

Yeh, Y. T., Patt, Y. N. (1993). A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History. *Proc. 20th Ann. Int'l Symp. On Computer Architecture*, Pages 257-267.

Tables

Table 1

PAp Predictor Percentage Accuracy

|  |  |  |  |
| --- | --- | --- | --- |
| BHT\BHR | 2 | 4 | 8 |
| 512 | 77.244 | 78.960 | 81.081 |
| 1024 | 87.349 | 89.867 | 92.818 |
| 2048 | 89.417 | 91.985 | 94.975 |
| 4096 | 89.417 | 91.985 | 94.975 |
| 8192 | 89.417 | 91.985 | n/a |

Note: this table and the following tables contain data for two parameters that were used to setup each branch predictor. Column with values like 512, 1024, etc. are number of entries for the Branch History Table or Number of Pattern History Tables, and the row with values like 2, 4, 8, are number of bits used for the Branch History Register. Simulation with 8192 entries for BHT and 8 bits for BHR cannot be performed due to computer restrictions.

Table 2

PAp Predictor Cost in Kbytes

|  |  |  |  |
| --- | --- | --- | --- |
| BHT\BHR | 2 | 4 | 8 |
| 512 | 0.640 | 2.304 | 33.280 |
| 1024 | 1.280 | 4.608 | 66.560 |
| 2048 | 2.560 | 9.216 | 133.120 |
| 4096 | 5.120 | 18.432 | 266.240 |
| 8192 | 10.240 | 36.864 |  |

Table 3

GAp Predictor Percentage Accuracy

|  |  |  |  |
| --- | --- | --- | --- |
| BHT\BHR | 2 | 4 | 8 |
| 512 | 69.583 | 78.323 | 92.055 |
| 1024 | 76.014 | 84.060 | 93.127 |
| 2048 | 80.100 | 86.796 | 93.773 |
| 4096 | 83.359 | 88.465 | 93.932 |
| 8192 | 85.233 | 89.388 |  |

Table 4

GAp Predictor Cost in Kbytes

|  |  |  |  |
| --- | --- | --- | --- |
| BHT\BHR | 2 | 4 | 8 |
| 512 | 0.512 | 2.049 | 32.769 |
| 1024 | 1.024 | 4.097 | 65.537 |
| 2048 | 2.048 | 8.193 | 131.073 |
| 4096 | 4.096 | 16.385 | 262.145 |
| 8192 | 8.192 | 32.769 |  |

Table 5

GShare Predictor Percentage Accuracy

|  |  |  |  |
| --- | --- | --- | --- |
| BHT\BHR | 2 | 4 | 8 |
| 512 | 65.960 | 72.103 | 91.912 |
| 1024 | 71.772 | 79.112 | 93.713 |
| 2048 | 77.808 | 84.696 | 94.754 |
| 4096 | 84.475 | 89.308 | 95.530 |
| 8192 | 89.135 | 92.373 |  |

Table 6

GShare Predictor Cost in Kbytes

|  |  |  |  |
| --- | --- | --- | --- |
| BHT\BHR | 2 | 4 | 8 |
| 512 | 0.512 | 2.049 | 32.769 |
| 1024 | 1.024 | 4.097 | 65.537 |
| 2048 | 2.048 | 8.193 | 131.073 |
| 4096 | 4.096 | 16.385 | 262.145 |
| 8192 | 8.192 | 32.769 |  |

Figures

Figure 1: PAp Predictor Percentage Accuracy

Figure 2: PAp Predictor - Percentage Accuracy per Byte

Figure 3: GAp Predictor Percentage Accuracy

Figure 4: GAp Predictor - Percentage Accuracy per Byte

Figure 5: GShare Predictor Percentage Accuracy

Figure 6: GShare Predictor - Percentage Accuracy per Byte

Program Output

Source Code