**Project 2 Report:**

**gshare Predictor**

By: Joshua Nichols

Computer Architecture

EEL 4768-003

12/01/2019

**Objective**

To see the impact of changing the Global Branch History (GBH) size and the number of lower program counter bits on the misprediction rate of the gshare branch predictor that we made. When the lower program counter bits parameter is set to zero, the gshare acts as a simple bimodal branch predictor.

**Process**

* Run each experiment using the setup mentioned for each.
* Due to not being able to run this in Eustis (as it was down at the time of writing this report), the following method was used to test the program in command prompt:
  + gcc -o gshare gshare.c
  + gshare.exe [ M ] [ N ] "C:\Users\Josh\Desktop\[ T ]"
  + Where:
    - M is the number of lower program counter bits.
    - N is the number of bits in the GBH register.
    - T is the trace file name.

**Global Branch History Analysis**

The following settings were maintained for the predictor as instructed:

* M is set to 4 bits.

|  |  |  |
| --- | --- | --- |
| **N** | **Misprediction Rate (mcf\_trace)** | **Misprediction Rate (gobmk\_trace)** |
| 1 | 24.71% | 0.77% |
| 2 | 26.86% | 0.87% |
| 3 | 29.36% | 0.86% |
| 4 | 31.72% | 0.82% |

Table 1: Misprediction rates for the given trace files vs GBH Register Size.

Graph 1: Misprediction Rate vs GBH Register Size

Having a larger number of GBH entries makes it more likely that branches will map to the same two-bit predictor (called aliasing). This interference causes an increase in the misprediction rate.

**Lower Program Counter Bits Analysis**

The following settings were maintained for the predictor as instructed:

* N is set to 4 bits.

|  |  |  |
| --- | --- | --- |
| **M** | **Misprediction Rate (mcf\_trace)** | **Misprediction Rate (gobmk\_trace)** |
| 4 | 31.72% | 0.82% |
| 5 | 26.56% | 0.67% |
| 6 | 19.81% | 0.60% |
| 7 | 12.40% | 0.58% |

Table 2: Misprediction rates for the given trace files vs the number of saved lower program counter bits using the gshare predictor scheme.

Graph 2: Misprediction Rate vs Saved Lower Program Counter Bits (gshare)

Saving more of the lower bits of the program counter generally caused a decrease in the misprediction rate. This is because with more bits the likelihood of unique two bit predictors for addresses increased and increases the accuracy of the predictor overall.

**Simple Bimodal Branch Predictor**

The following settings were maintained for the predictor as instructed:

* N is set to 0 bits.

|  |  |  |
| --- | --- | --- |
| **M** | **Misprediction Rate (mcf\_trace)** | **Misprediction Rate (gobmk\_trace)** |
| 4 | 23.76% | 0.69% |
| 5 | 20.83% | 0.66% |
| 6 | 15.07% | 0.67% |
| 7 | 10.63% | 0.60% |

Table 3: Misprediction rates for the given trace files vs the number of saved lower program counter bits using the simple bimodal predictor scheme.

Graph 3: Misprediction Rate vs Saved Lower Program Counter Bits (simple bimodal)

Both the simple bimodal and gshare branch predictors showed a decrease in the misprediction rate when the number of lower program counter bits utilized increases. This is because the gshare predictor scheme acts similar to the bimodal prediction scheme and avoids the aliasing problem mentioned in the global branch history analysis.

**Conclusion**

From this project and the experiments performed above, we have seen how both the size of the global branch history and the number of saved program counter bits influences the misprediction rate of the gshare branch predictor. Changing the global branch history size caused a general increase in the misprediction rate due to overlapping mappings, called aliasing. This problem is reduced when more of the lower program counter bits are retained, and the addresses become more unique. We have also seen how the behaviors of both the simple bimodal and gshare predictors are similar.