NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Fall 2015 (Rotenberg)

Project #2: Branch Prediction

By

<< FALAK VIJAY RAVANI >>

NCSU Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."

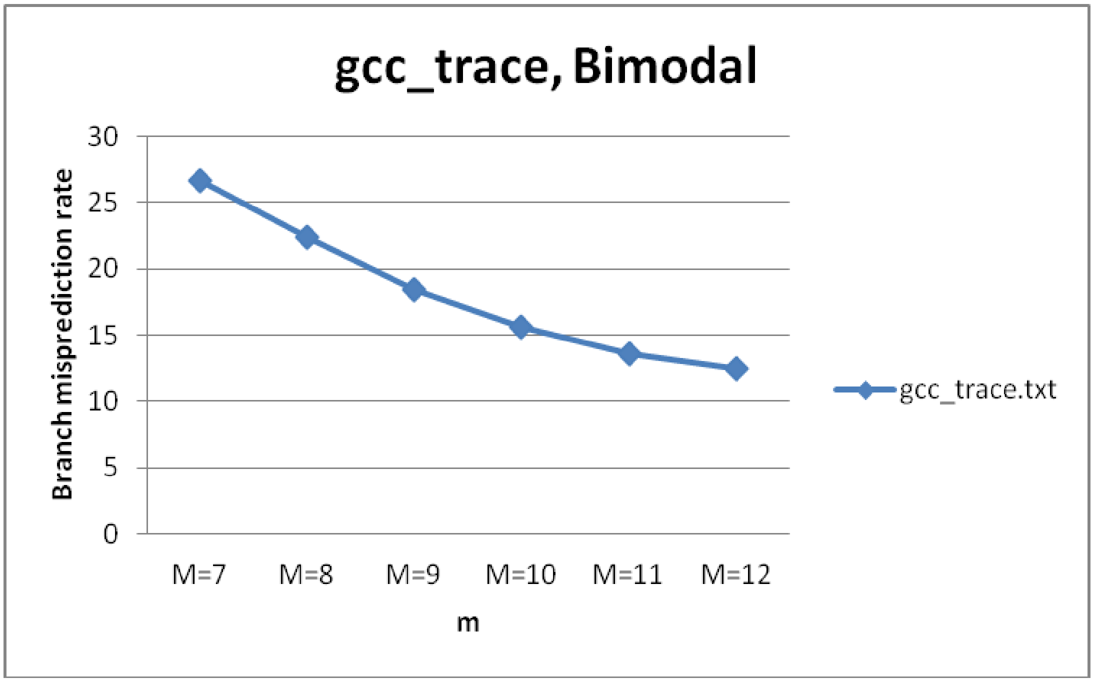
Student's electronic signature: __________Falak Vijay Ravani_____________________
(sign by typing your name)

Course number: ________521___________
(463 or 521 ?)

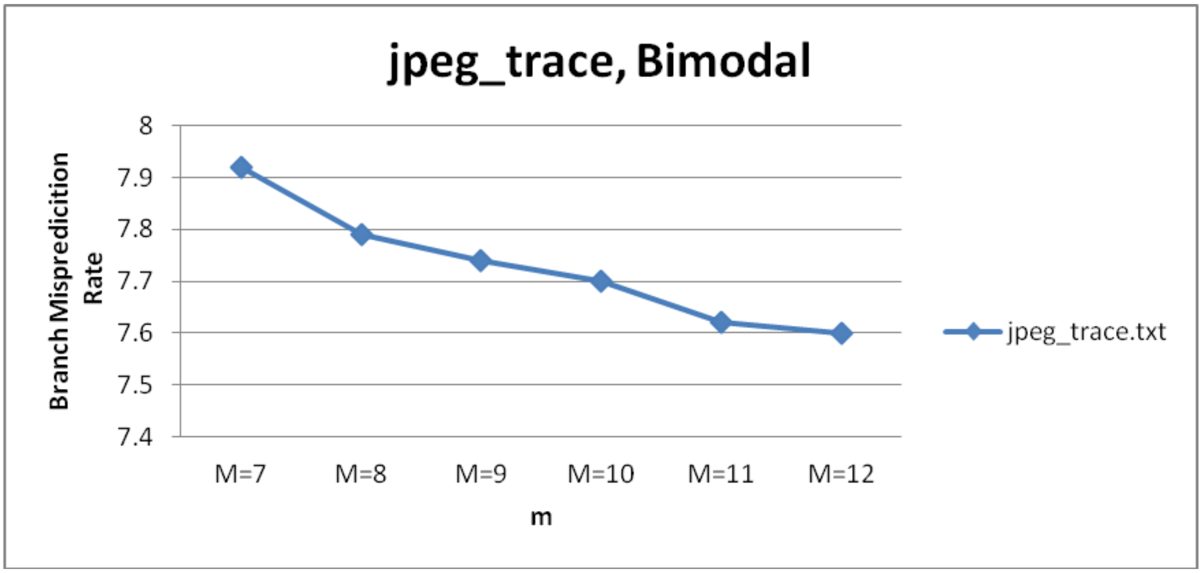
Part 1: BIMODAL PREDICTOR

Graphs:

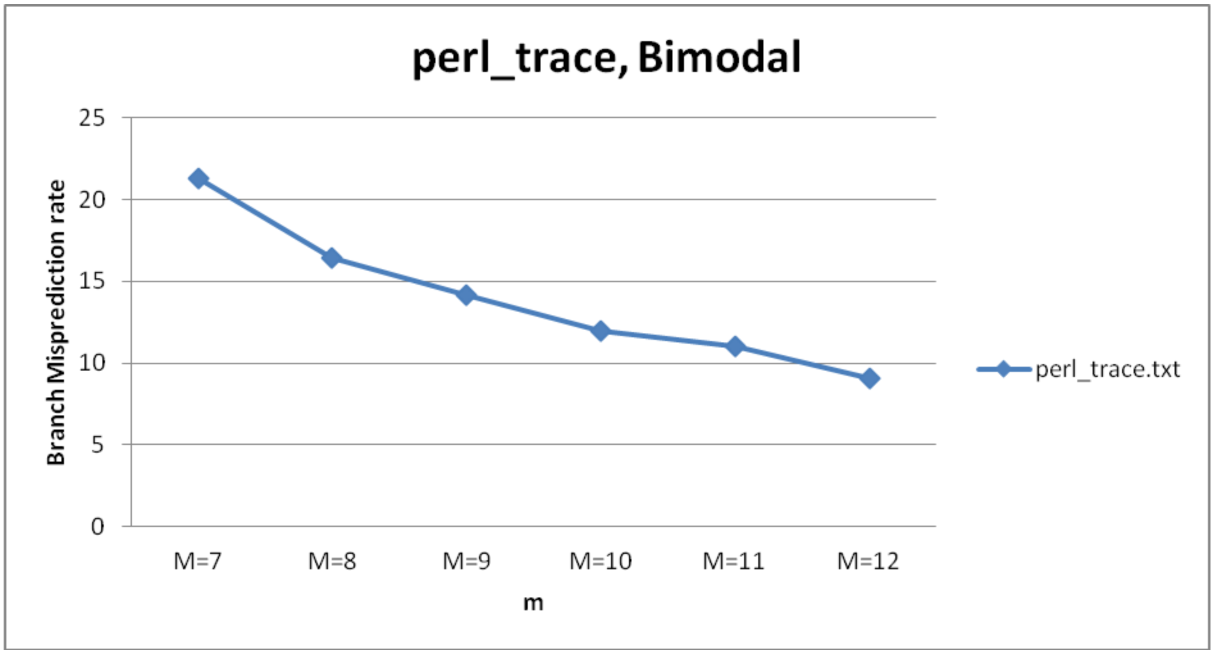
a) gcc_trace, bimodal



b) jpeg_trace, bimodal



c) perl_trace, bimodal



Analysis:

As m goes on increasing, the misprediction rate goes on decreasing. As we increase m, less branches index the same entry in the prediction table i.e. interference goes on decreasing. Thus, accuracy increases as number of counters increase.

Similarities: We observe that all the graphs are decreasing and the rate of decrement is decreasing as well. This implies that at one point, the misprediction rate will saturate and further increase in m won't result in significant change in the misprediction rate. This point in the graph is called as "point of diminishing returns."

Differences: The graphs differ in their starting rates and their slopes. The slopes of gcc, jpeg and perl traces are 53.20, 4.04 and 57.34 respectively. This shows that the rate of misprediction is decreasing more rapidly with increase in m for perl_trace as compared to gcc_trace and jpeg_trace.

Design:

gcc_trace	**Misprediction Rate**	**Percentage Improvement**
m=7	26.65	
m=8	22.43	15.83%
m=9	18.49	17.56%
m=10	15.67	15.25%
m=11	13.65	12.89%
m=12	12.47	8.64%
m=13	11.72	6.01%

jpeg_trace	**Misprediction Rate**	**Percentage Improvement**
m=7	7.92	
m=8	7.79	1.64%
m=9	7.74	0.64%
m=10	7.7	0.51%
m=11	7.62	1.03%
m=12	7.6	0.26%
m=13	7.59	0.13%

perl_trace	**Misprediction Rate**	**Percentage Improvement**
m=7	21.31	
m=8	16.45	22.80%
m=9	14.14	14.04%
m=10	11.95	15.48%
m=11	11.05	7.53%
m=12	9.09	17.73%
m=13	8.92	1.87%

The above tables show improvements in misprediction rates as m goes on increasing for gcc, jpeg and perl traces. Since our storage budget is of 16 kilobytes, misprediction rate for m=13 is also considered **[2^13**(Number of entries in the prediction table) *** 2**(no. of bits for each counter) **= 16kB]**.

We will select the design for which improvement in the misprediction rate has been more than 1%.

Gcc_trace: For gcc_trace, the lowest misprediction rate is observed for m=13 and the improvement from m=12 is 6.01%. This satisfies our criterion. Hence, best performance predictor for gcc_trace is m=13 with a misprediction rate of 11.72% and predictor cost equal to 16kB.

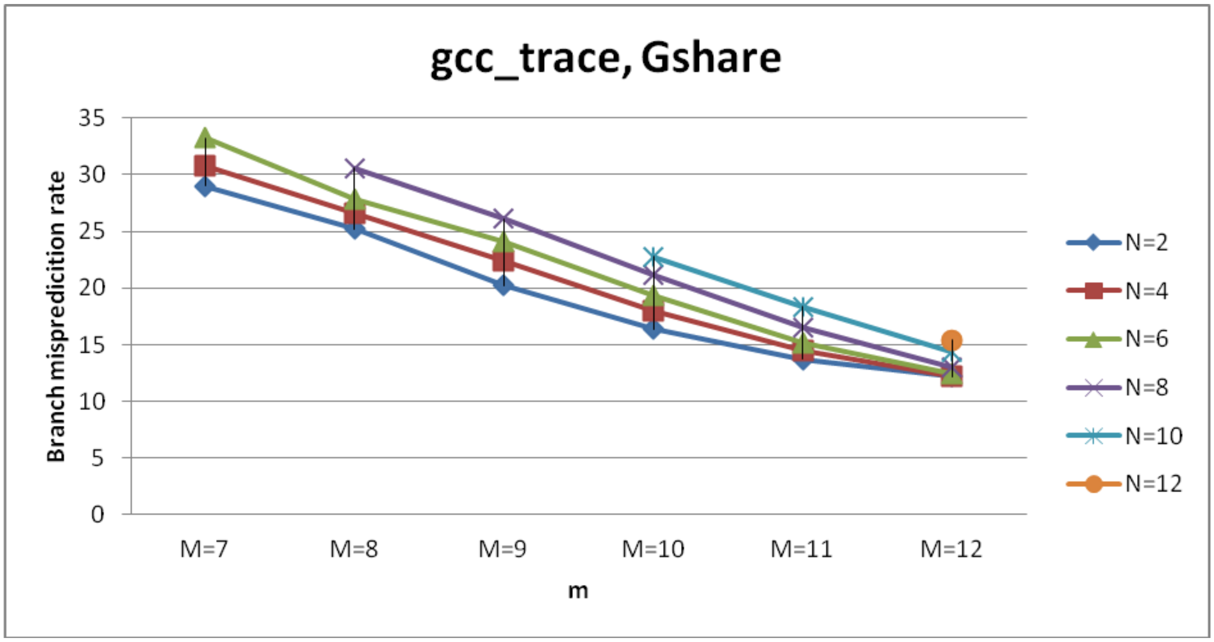
Jpeg_trace: For jpeg_trace, the lowest misprediction rate is observed for m=13. However, there isn’t a significant improvement in its performance as compared to m=12. The next best predictor is m=12 but even its improvement is less than 1%. Hence, we select a predictor with m=11 for jpeg_trace as its improvement from m=10 is 1.03% (>1%). This satisfies our criterion. The predictor cost for m=11 is 4kB.

Perl_trace: For perl_trace, the lowest misprediction rate is observed for m=13 and its improvement from m=12 is 1.87% which is greater than 1%. This satisfies our criterion. Hence, we choose a predictor with m=13 and a predictor cost of 16kB for perl_trace.

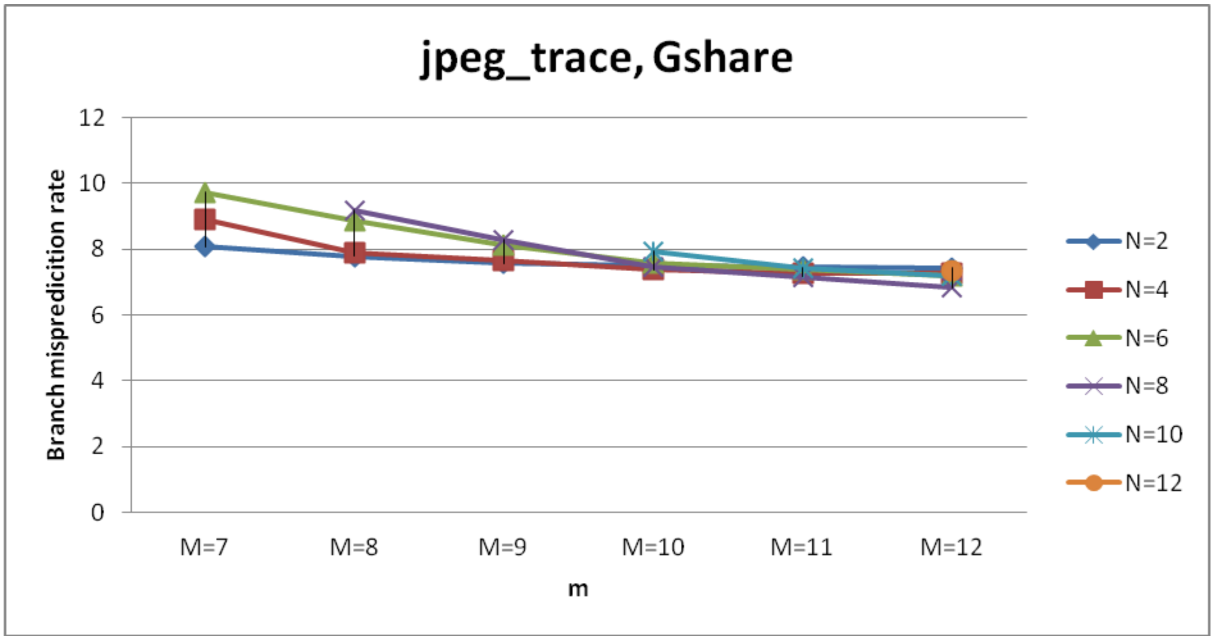
Part 2: GSHARE PREDICTOR

Graphs:

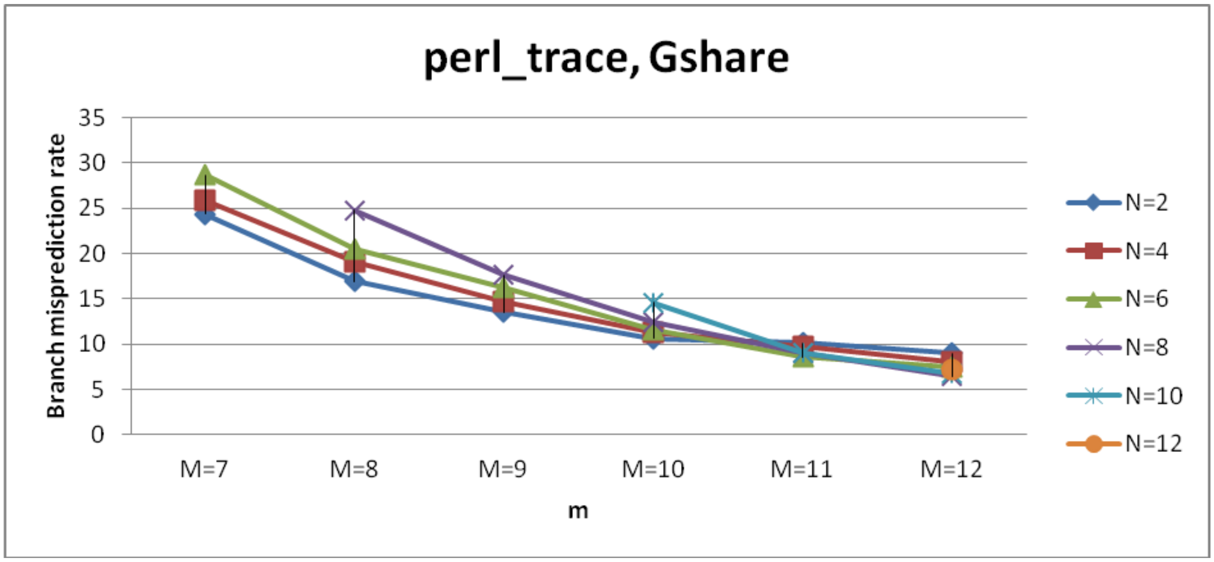
a) gcc_trace.txt



b) jpeg_trace.txt



c) perl_trace.txt



Analysis:

As the value of m increases, for a fixed value of n, the misprediction rate goes on decreasing. As we increase m, less branches index the same entry in the prediction table i.e. interference goes on decreasing. Thus, accuracy increases as number of counters increase.

Also, for a fixed value of m, the misprediction rate goes on decreasing as value of n increases. Thus, we take into consideration two factors –

1) Global Branch History register
2) Previous history of a particular branch

When we hash branch address and global history by “Exclusivly ORing” them, we expect to store more information in the result as compared to individual factors. Hence, our prediction may be more accurate.

Similarities: We observe that all the graphs are decreasing and the rate of decrement is decreasing as well. This implies that at one point, the misprediction rate will saturate and further increase in m won’t result in significant change in the misprediction rate. This point in the graph is called as “point of diminishing returns.” Also, we observe that lower value of n saturates faster than higher values of n.

Differences: The jpeg_trace saturates much faster than other two traces.

Design:

gcc_trace.txt	M=7	M=8	M=9	M=10	M=11	M=12
N=2	28.98	25.18	20.25	16.39	13.71	12.2
N=4	30.76	26.57	22.43	17.99	14.49	12.23
N=6	33.22	27.82	24.14	19.36	15.14	12.46
N=8		30.56	26.08	21.1	16.47	13
N=10				22.77	18.34	14.33
N=12						15.4

jpeg_trace.txt	M=7	M=8	M=9	M=10	M=11	M=12
N=2	8.08	7.79	7.58	7.49	7.45	7.44
N=4	8.92	7.88	7.68	7.38	7.27	7.26
N=6	9.74	8.87	8.13	7.58	7.38	7.19
N=8		9.2	8.3	7.45	7.17	**6.84**
N=10				7.95	7.44	7.18
N=12						7.35

perl_trace.txt	M=7	M=8	M=9	M=10	M=11	M=12
N=2	24.34	16.92	13.57	10.63	10.11	9.03
N=4	25.96	19.09	14.68	11.35	9.68	8.09
N=6	28.71	20.45	16.25	11.52	8.6	7.5
N=8		24.79	17.66	12.42	9	**6.49**
N=10				14.57	8.98	6.71
N=12						7.16

The above tables show the misprediction rates for various values of m and n. For a fixed value of n, the misprediction rate goes on decreasing as value of m goes on increasing.

Gcc_trace: For gcc_trace, the lowest misprediction rate is observed for m=12 and n=2. The cost of predictor is 8kB which is within our budget of 16 kB storage. Thus, we choose this gshare predictor for gcc_trace.

Jpeg_trace: The lowest misprediction rate for jpeg_trace is observed for m=12 and n=8. The cost of this predictor is 8kB as well which is within our storage budget.

Perl_trace: For perl_trace, the lowest misprediction rate is observed for m=12 and n=8. Hence, we choose this gshare predictor (m=12, n=8) for perl_trace as its cost is 8kB(<16kB).