NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2019 (Dr. Huiyang Zhou)

Project #2: Branch Prediction

By

_____Krishna Dheeraj Putrevu (ID:kputrev@nccsu.edu)_____

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

Student's Electronic Signature: _____Krishna Dheeraj Putrevu
(Sign by typing your name)

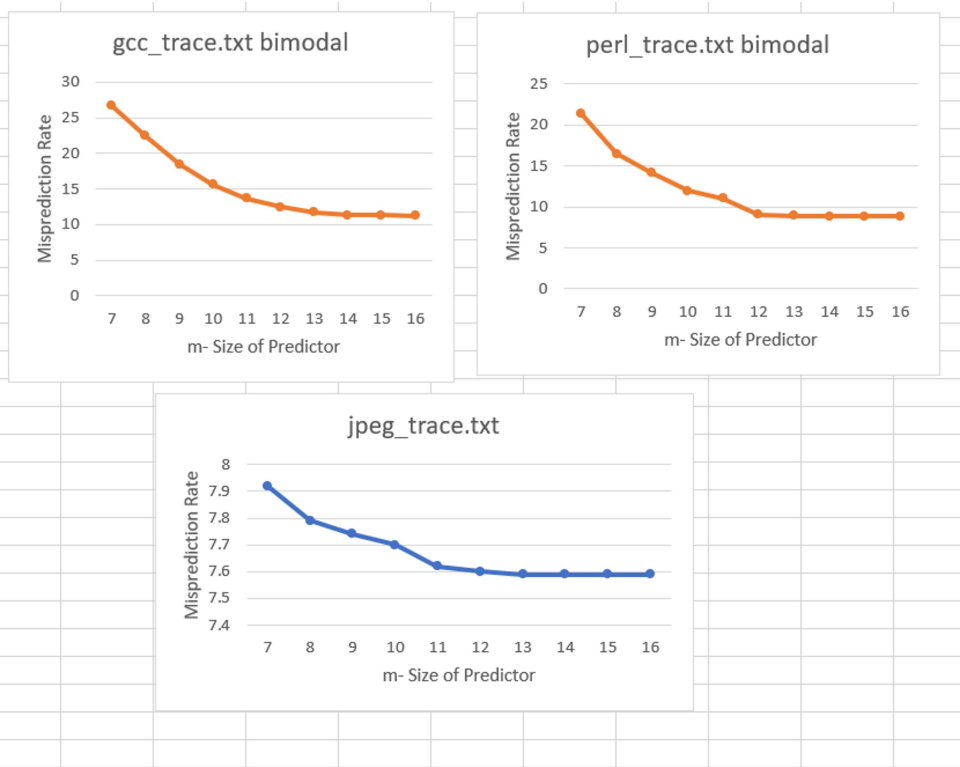
Course Number: ____563____

1) BIMODAL PREDICTOR:

Misprediction Rate behavior for different traces:
gcc_trace:
Saturates at around m=14.
perl_trace:
Saturates at around m=12
jpeg_trace:
Saturates at around m=11,12

All the traces show similar behavior for an increasing size.

Figure: Misprediction Rate vs Size of the Predictor



Design:
Assuming the predictor is implemented as a SRAM memory whose cost per bit around 2010(data from the internet) is 5000 dollars per GB. 5 dollars per kb. It can be substantial for a consumer computer but not so much for a high performance computer.

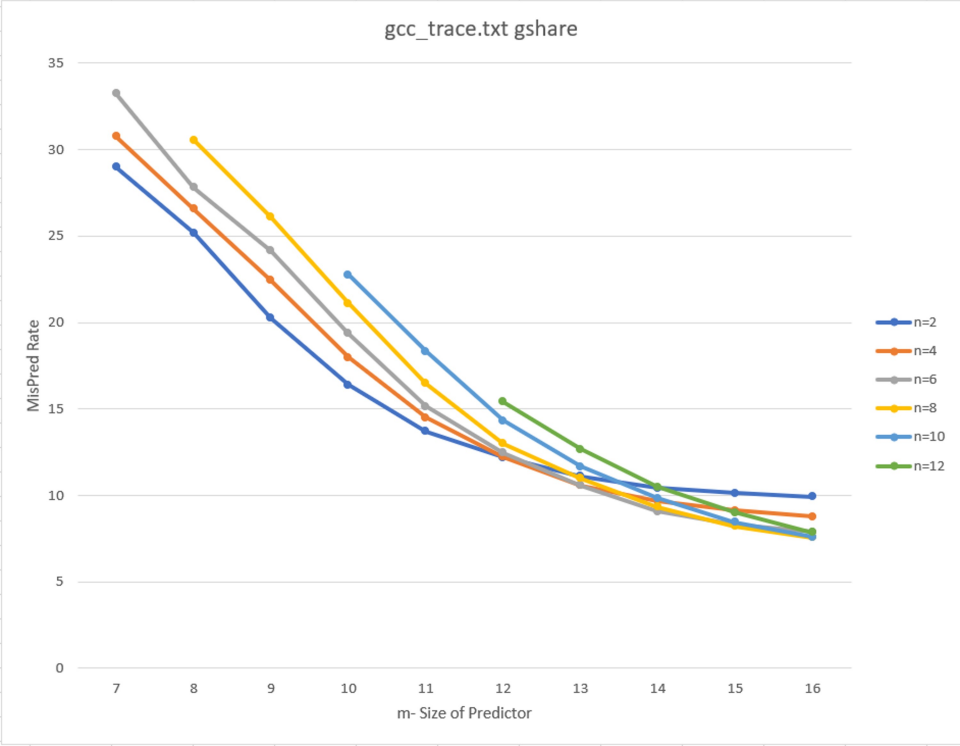
For gcc_trace:
M=13 has a missrate of 11.72% costing 2kb
M=14 has 11.37% costing 4kb after which its minimal.
Both these look optimum. Depending on which application these computers are used for, these values look like a good choice.

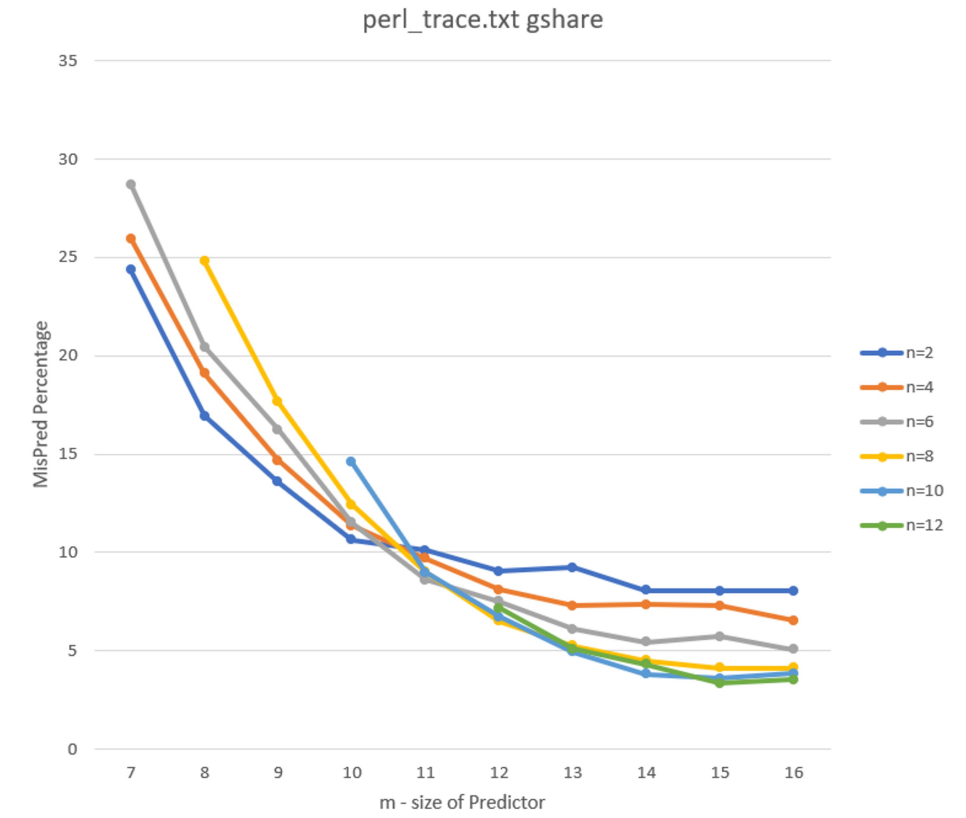
For perl_trace:

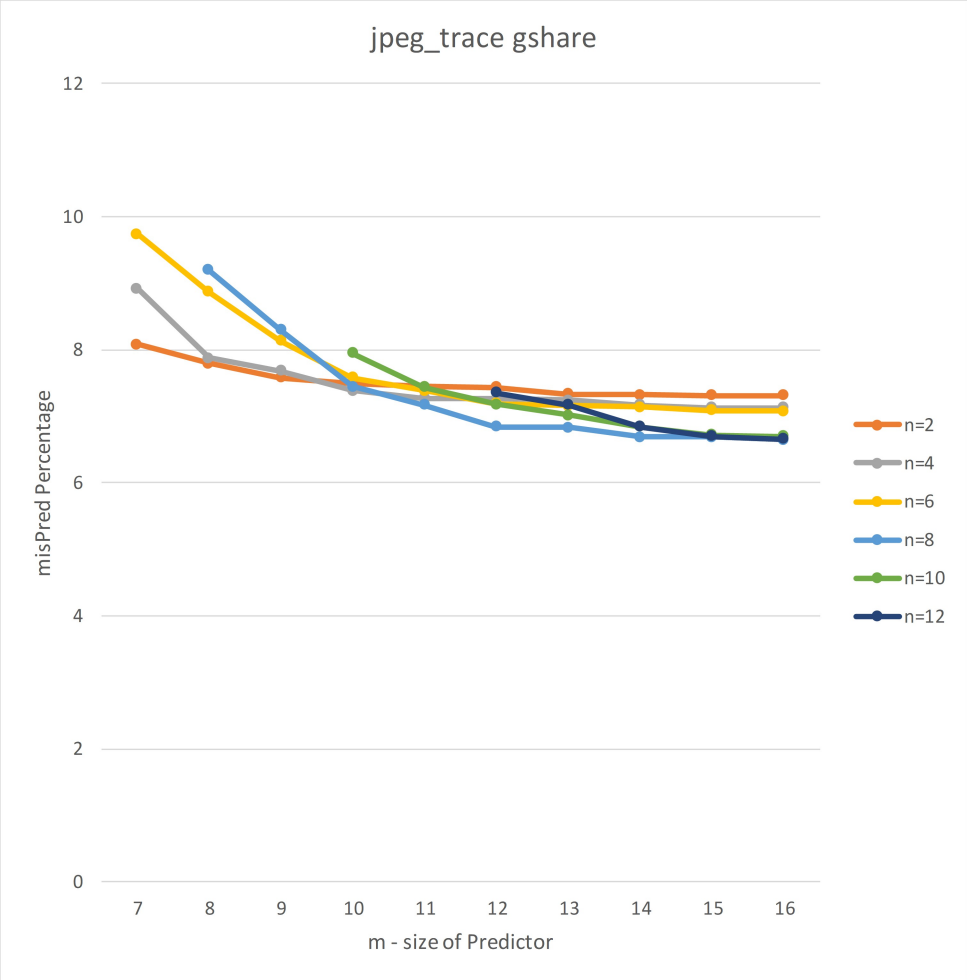
M=12 having a miss rate of 9.09% seems optimum. M=13 can also be another choice. Further its very sub-optimal gain vs cost improvement.

For jpeg_trace:
M=11 looks optimum with a miss rate of 7.62%

2) GSHARE PREDICTOR




perl_trace.txt gshare
MisPred Percentage
35
30
25
20
15
10
5
0
7
8
9
10
11
12
13
14
15
16
m - size of Predictor
n=2
n=4
n=6
n=8
n=10
n=12



Discussion:

- All traces generally have a trend of improved performance with an increase in the

Value of n. So a Branch history register tends to improve performance to some extent until it saturates.

- While obsesrving the graphs its worth noting that the graphs with a relatively falttened out curve have reached their saturation in terms of optimisation. Gcc_trace is a case in point.

Design:

Gcc_trace:

The least misprediction percenetage from the graph is seen for m=16 at n=8 with 7.57%. Second best being m=16 with n=10 at 7.61%

I think the performance can be improved with a bigger budget because the curves don't saturate at m= 16.

Perl_trace:

M=16 n=16 gives better rate at 2.91%. A smaller size at m=15 and n=12 gives a 3.35% which can be good option if cost seems to be a deciding point and we don't want to use the whole budget for whatever reason. Increasing the budged may not necessarily improve performance though because of the steep fall and plateauing of the graph.

Jpeg_trace:

Best performance comes at m=16 n=14 with misprediction rate of 6.57%. But m=14 n=12 gives a rate of 6.84% with 1/4th the cost so that would be a prudent selection.