Project Report

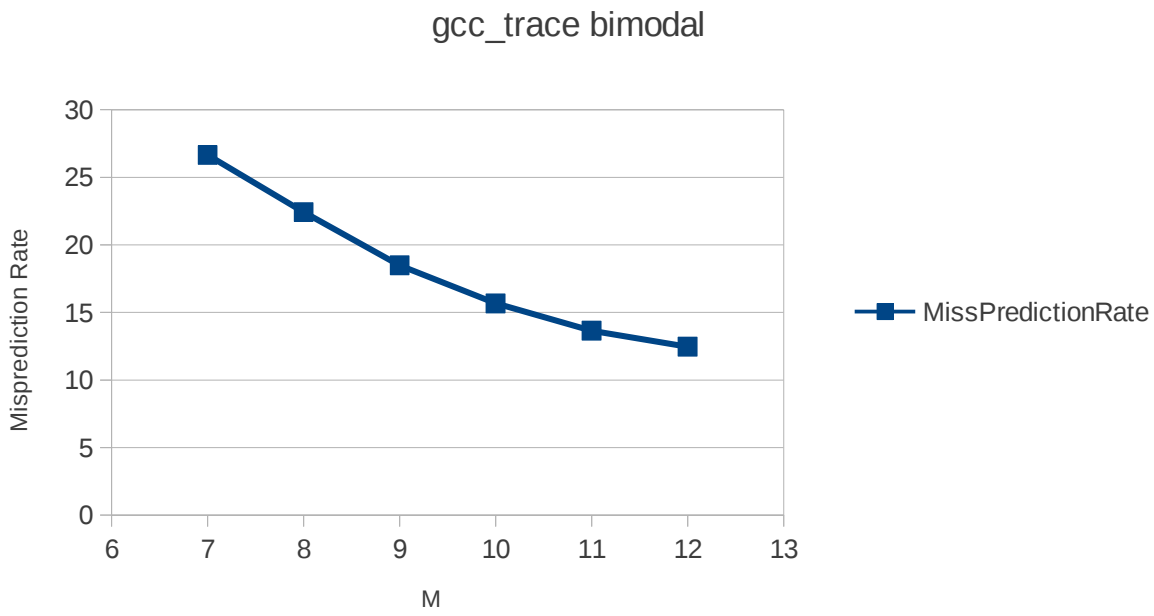
Branch Predictor Simulation

Munawira Kotyad mkotyad

Tasks:

PART 1: BIMODAL PREDICTOR

1) For gcc_trace:



The readings for gcc_trace are as follows:

M	Misprediction Rate
7	26.65
8	22.43
9	18.49
10	15.67
11	13.65
12	12.47

The readings show that as we increase the value of M, the Misprediction rate reduces. This is

because, as M increases , the number of collisions to the same index reduces and thus the mispredictions due to collisions reduces.

Design of an Optimized Branch Predictor:

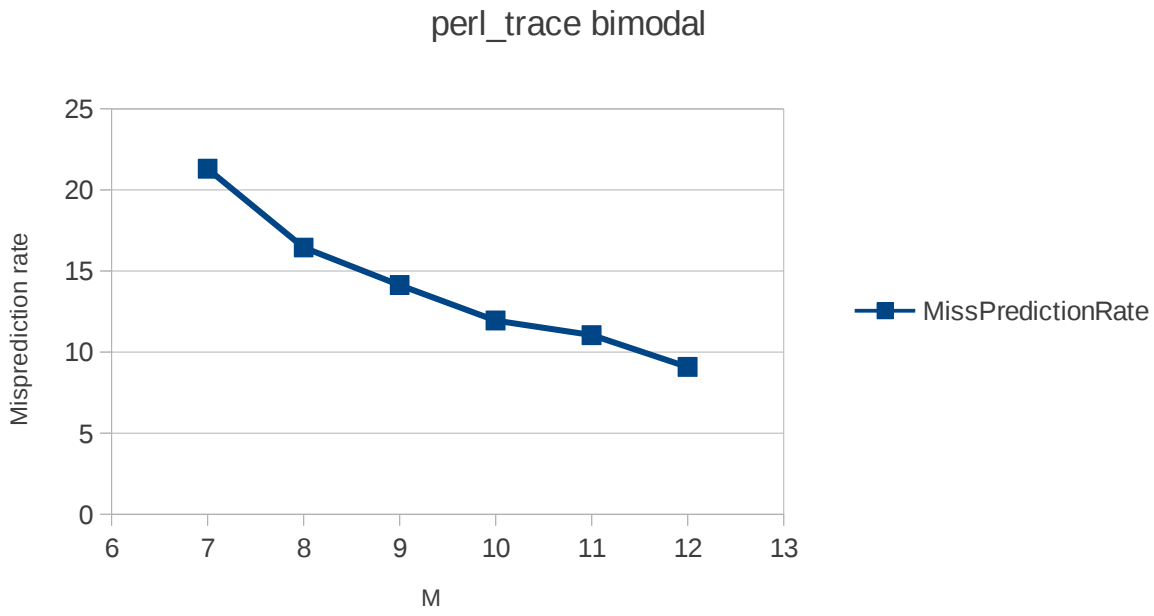
Each row in a branch predictor needs to store a 2 bit counter. This can be done using an int datatype which takes up 2 bytes. Thus the total storage for the table is 2^M * 2 bytes

For M=10, the storage is 2048 bytes = 2KBytes.

M=10 gives a misprediction rate of13.65% which is significantly lower than with M=9. M=11 doesn't give any significant improvement over M=11 and takes up 2^12 * 2 = 8 KBytes of storage.

Thus M=10 gives an optimum performance. Without much storage overhead

2) For perl_trace:



M	Misprediction Rate
7	21.31
8	16.45
9	14.14
10	11.95
11	11.05
12	9.09

The readings show that as we increase the value of M, the Misprediction rate reduces. This is because, as M increases , the number of collisions to the same index reduces and thus the

mispredictions due to collisions reduces.

Design of an Optimized Branch Predictor:

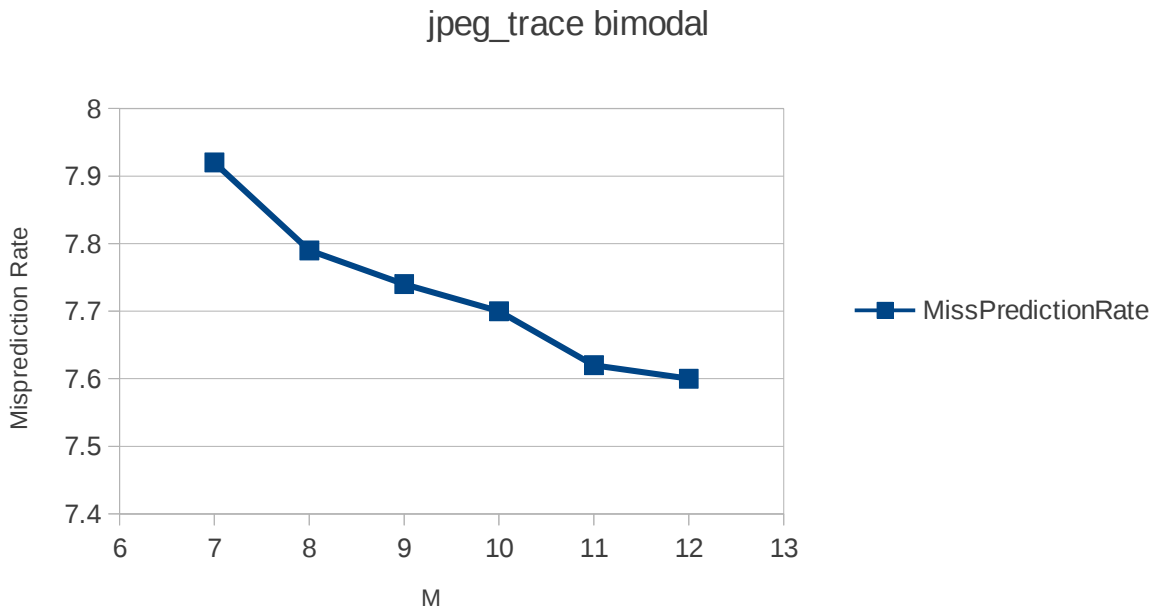
Each row in a branch predictor needs to store a 2 bit counter. This can be done using an int datatype which takes up 2 bytes. Thus the total storage for the table is 2^M * 2 bytes

For M=10, the storage is 2048 bytes = 2KBytes.

M=10 gives a misprediction rate of 11.95% which is significantly lower than with M=9 (14.14). M=11 and M=12 doesn't give any significant improvement over M=10 and takes up 2^12 * 2 = 8 KBytes of storage.

Thus M=10 gives an optimum performance. Without much storage overhead

3) For jpeg_trace:



M	Misprediction Rate
7	7.92
8	7.79
9	7.74
10	7.7
11	7.62
12	7.6

The readings show that as we increase the value of M, the Misprediction rate reduces. This is because, as M increases , the number of collisions to the same index reduces and thus the mispredictions due to collisions reduces.

Design of an Optimized Branch Predictor:

Each row in a branch predictor needs to store a 2 bit counter. This can be done using an int datatype which takes up 2 bytes. Thus the total storage for the table is 2^M * 2 bytes

For M=8, the storage is 512bytes .

M=8 gives a misprediction rate of 7.79% which is lower than with M=9 (7.74). M=9 and M=10 doesn't give any significant improvement over M=8 and takes up 2^9 * 2 = 1 KBytes of storage.

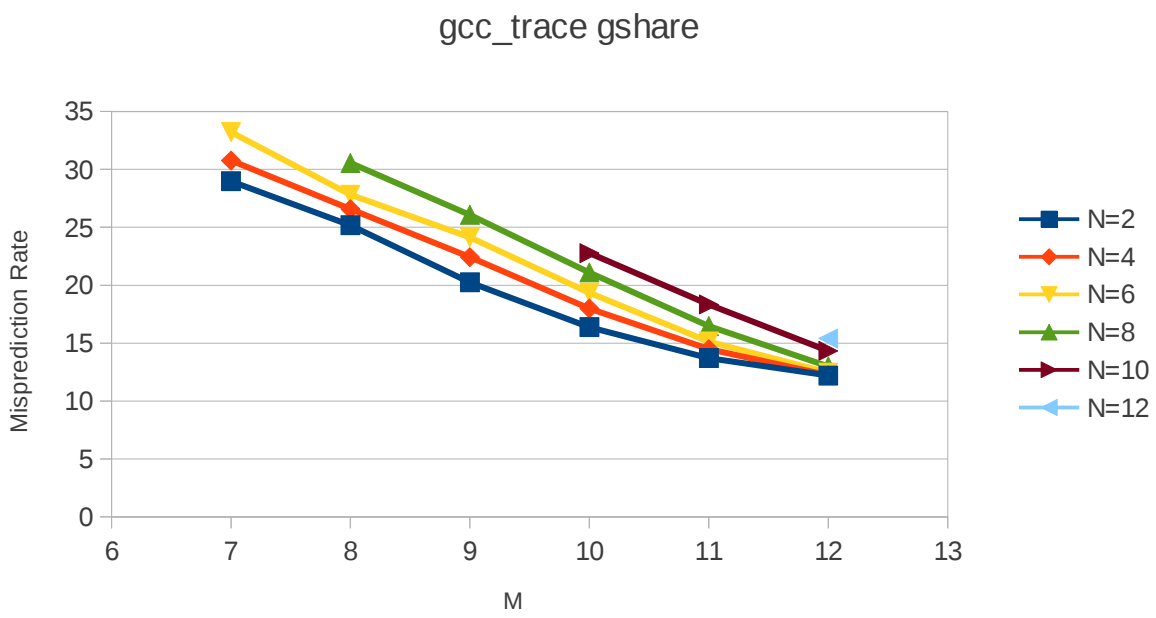
Thus M=8 gives an optimum performance. Without much storage overhead.

Comparison for gcc, jpeg and perl traces for Bimodal:

The misprediction rate is lesser for perl and jpeg as compared to gcc trace. This may be because, in the jpeg and perl traces, the number of branch collisions are lesser than that of the gcc trace. This leads to lesser mispredictions due to collisions.

PART 2: GSHARE PREDICTOR

1) gcc_trace:



N	M=7	M=8	M=9	M=10	M=11	M=12
2	28.98	25.18	20.25	16.39	13.71	12.2
4	30.76	26.57	22.43	17.99	14.49	12.23
6	33.22	27.82	24.14	19.36	15.14	12.46
8		30.56	26.08	21.1	16.47	13.00
10				22.77	18.34	14.33
12						15.4

The readings show that as we increase the value of M, the Misprediction rate reduces. This is because, as M increases , the number of collisions to the same index reduces and thus the mispredictions due to collisions reduces.

Also for a given value of M, as N is increased, the misprediction rate increases. This is because, due to collisions, the contents of the global history register are not always accurate for every program counter. Thus a value of BHR that could avoid misprediction for one PC may lead to an increase in misprediction. Thus an increase in number of bits in BHR while keeping M constant leads to an increase in the misprediction rate.

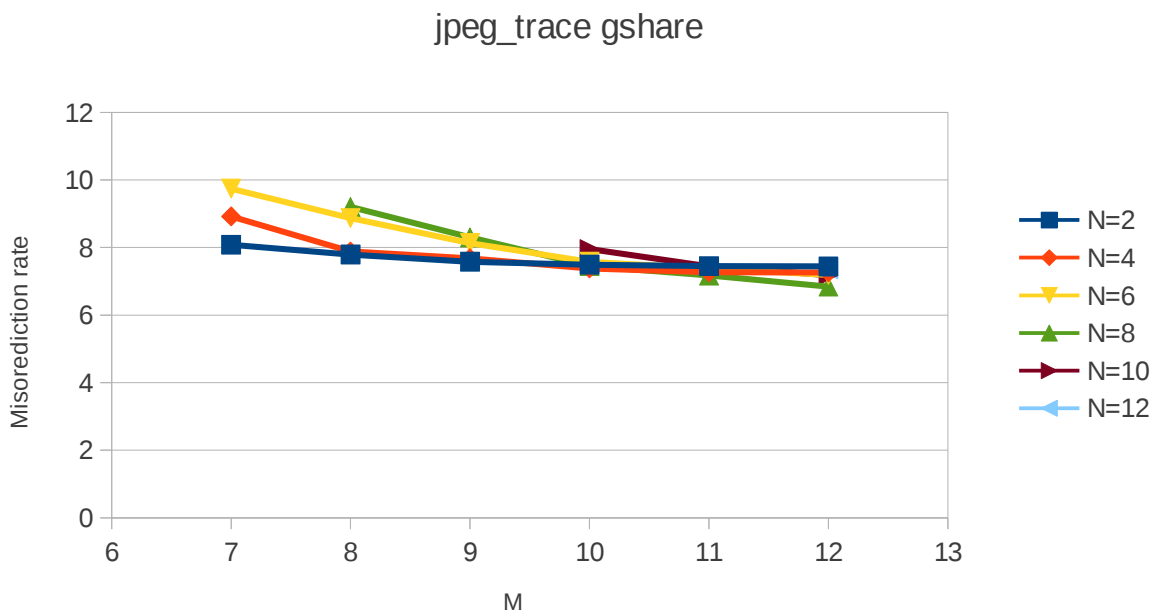
On the contrary, if we increase N and M , the misprediction rate reduces significantly. The decrease in misprediction rate due to increase in M is much more prominent than the decrease in misprediction rate due to increase in M and N.

Design of an Optimized Branch Predictor:

An optimized Branch Predictor would be with M=12 and N=2. This configuration gives a considerably lesser misprediction rate than M=11 with slightly greater storage overhead.

The Miss rate with this configuration is : 12.2%
The Storage required is 8KBytes plus 2 bits for the BHR.

2) jpeg_trace:



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

The readings show that as we increase the value of M, the Misprediction rate reduces. This is because, as M increases , the number of collisions to the same index reduces and thus the mispredictions due to collisions reduces.

Also for a given value of M, as N is increased, the misprediction rate increases. This is because, due to collisions, the contents of the global history register are not always accurate for every program counter. Thus a value of BHR that could avoid misprediction for one PC may lead to an increase in misprediction. Thus an increase in number of bits in BHR while keeping M constant leads to an increase in the misprediction rate.

On the contrary, if we increase N and M , the misprediction rate reduces significantly.

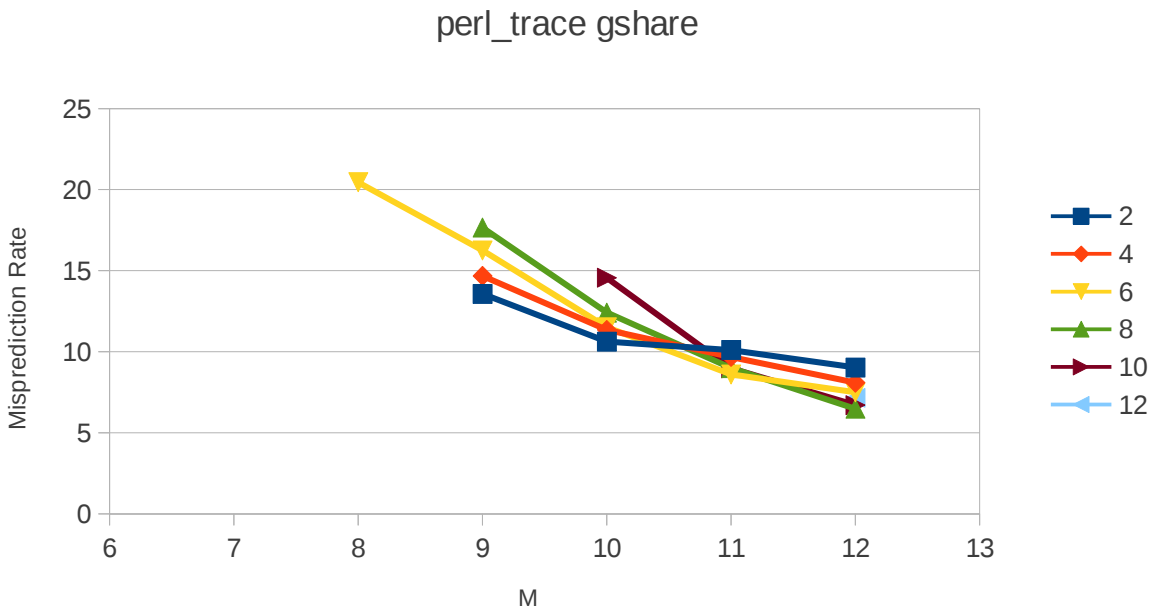
As, M is increased, there is a change in the trend and misprediction rates also decrease with the increase in the value of N. Thus increasing the value of N with the value of M leads to a reduction in the misprediction rate.

Design of an Optimized Branch Predictor:

An optimized Branch Predictor would be with M=10 and N=4. This configuration gives a considerably lesser misprediction rate than M=9 with lesser storage overhead.

The Miss rate with this configuration is : 7.38%
The Storage required is 2KBytes plus 4bits for the BHR.

3) perl_trace:



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

The readings show that as we increase the value of M, the Misprediction rate reduces. This is because, as M increases , the number of collisions to the same index reduces and thus the mispredictions due to collisions reduces.

Also for a given value of M, as N is increased, the misprediction rate increases. This is because, due to collisions, the contents of the global history register are not always accurate for every program counter. Thus a value of BHR that could avoid misprediction for one PC may lead to an increase in misprediction. Thus an increase in number of bits in BHR while keeping M constant leads to an increase in the misprediction rate.

On the contrary, if we increase N and M , the misprediction rate reduces significantly. T

As, M is increased, there is a change in the trend and misprediction rates also decrease with the increase in the value of N. Thus increasing the value of N with the value of M leads to a reduction in the misprediction rate.

Design of an Optimized Branch Predictor:

An optimized Branch Predictor would be with M=11 and N=4. This configuration gives a considerably lesser misprediction rate than M=10 with greater storage overhead.

The Miss rate with this configuration is : 11.35%
The Storage required is 4KBytes plus 4bits for the BHR.

Comparison for gcc, jpeg and perl traces for Gshare:

The misprediction rate is lesser for perl and jpeg as compared to gcc trace. This may be because, in the jpeg and perl traces, the number of branch collisions are lesser than that of the gcc trace. This leads to lesser mispredictions due to collisions.

Since the number of collision decreases, the increase in the number of bits of the BHR will help reduce the misprediction rate even further. The BHR will provide the history of the correct program counter due to lesser collisions and thus an improvement in misprediction rate is noticed.