NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Spring 2012 (Rotenberg)

Project #2: Branch Prediction

by

<< SALIL KANITKAR >>

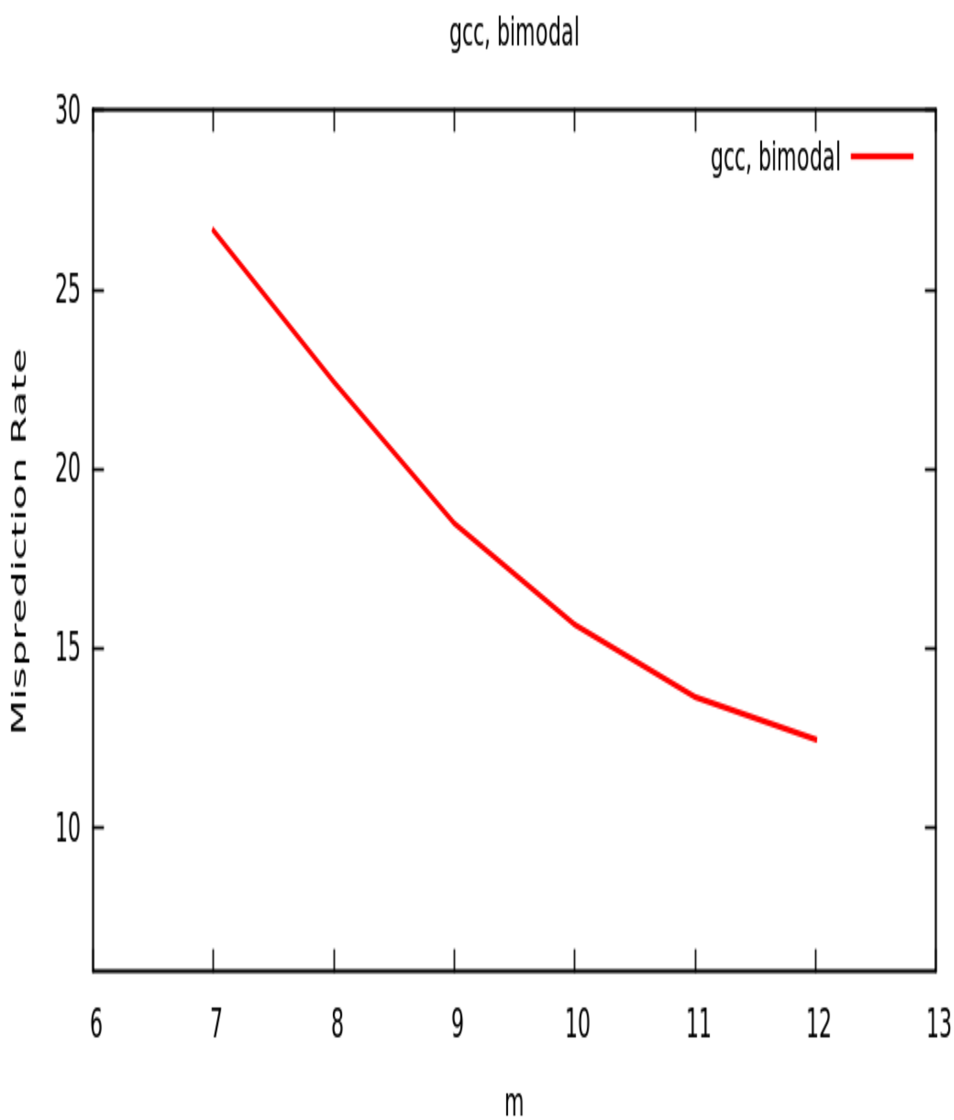
NCSU Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."

Student's electronic signature: _____Salil Kanitkar______
(sign by typing your name)

Course number: _______ 521 ________
(463 or 521 ?)

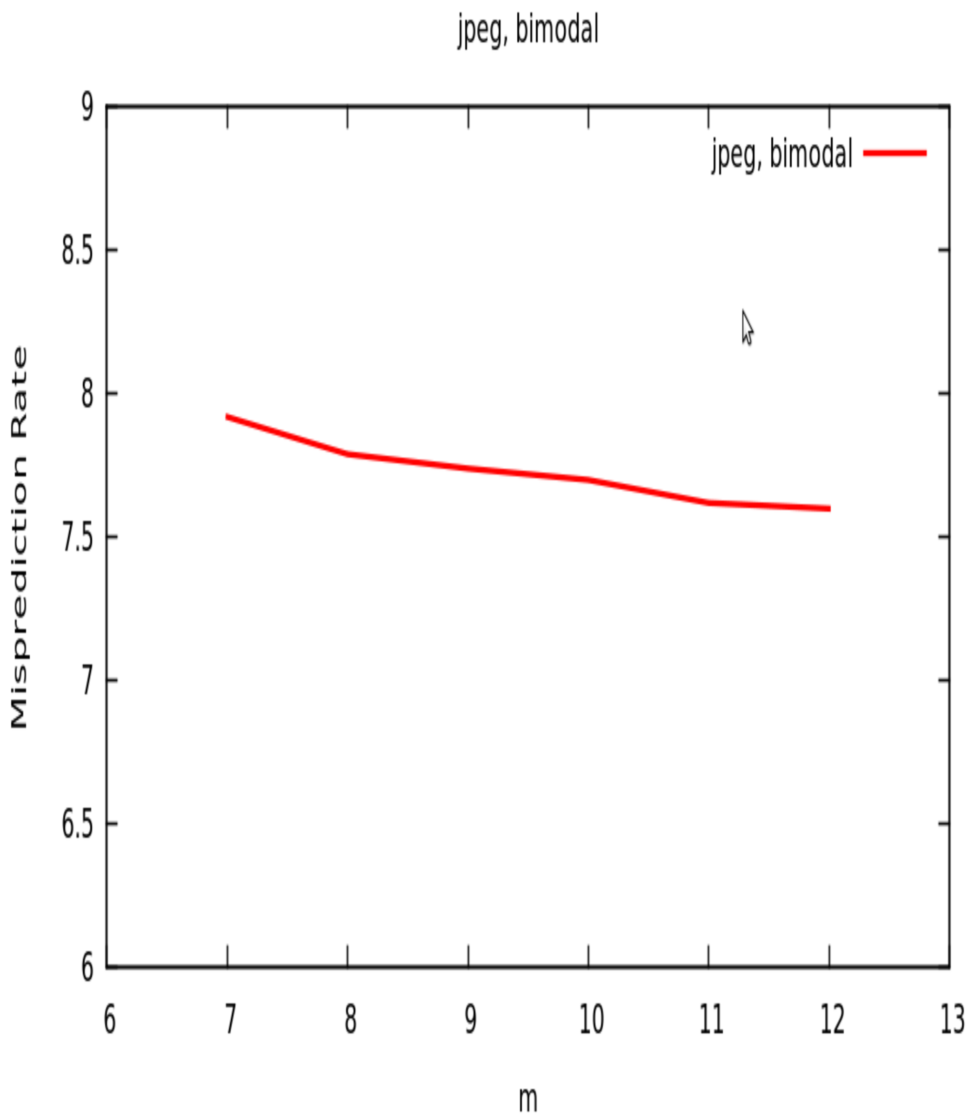
Bimodal Predictor

Graph1: gcc Benchmark:



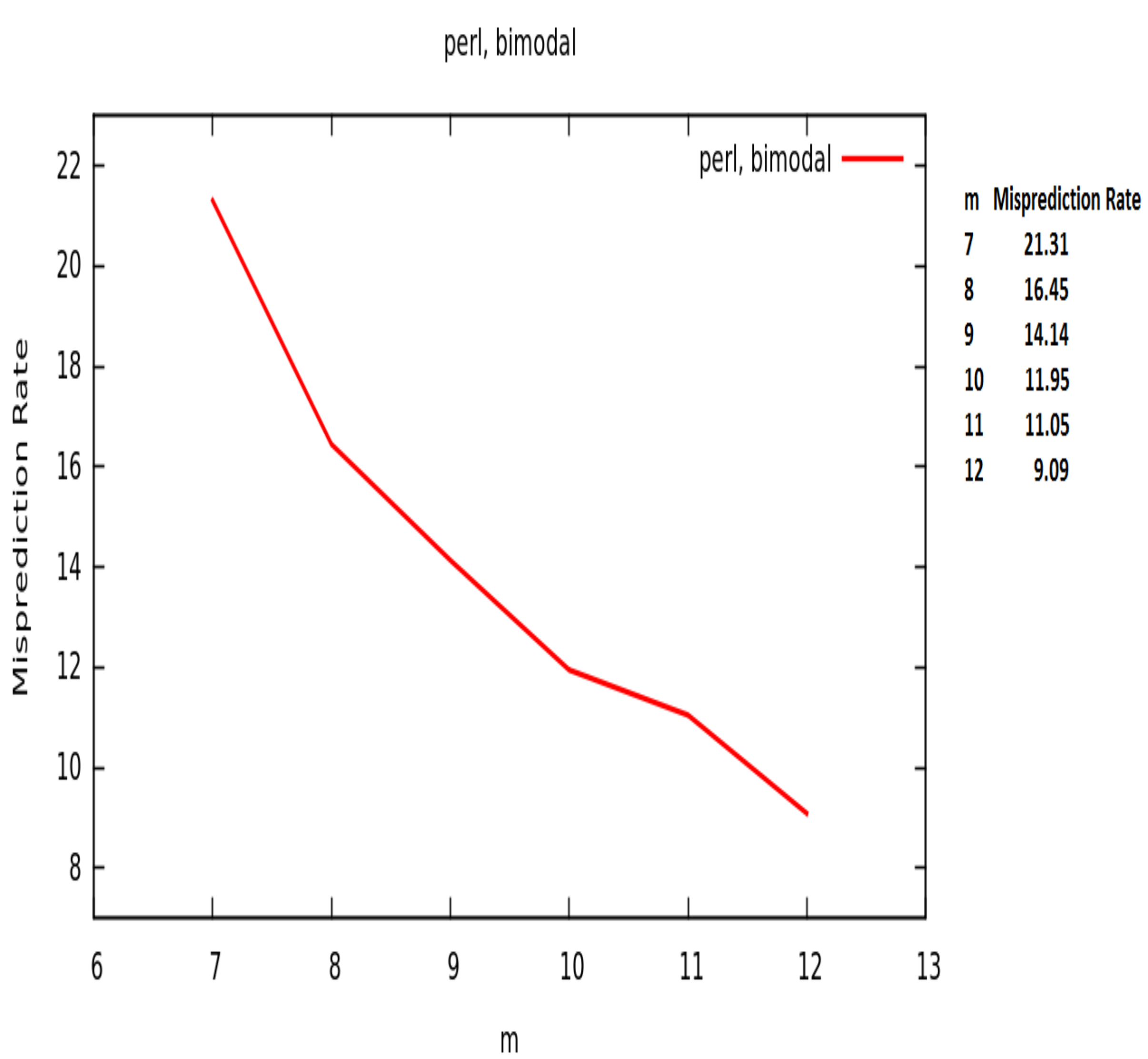
m	Misprediction rate
7	26.65
8	22.43
9	18.49
10	15.67
11	13.65
12	12.47

Graph2: jpeg Benchmark:



m	Misprediction Rate
7	7.92
8	7.79
9	7.74
10	7.70
11	7.62
12	7.60

Graph3: perl Benchmark:



Analysis:

*** Trends and Conclusions from Graphs:**

1. The value of Misprediction Rate goes on decreasing as the value of "M2" increases i.e. as the size of Prediction Table increases, the Misprediction Rate decreases.
2. However, the absolute values of Misprediction Rate for same values of M2 and the rate of decrease of Misprediction Rate is different for different benchmarks.
3. The maximum and minimum values for the three benchmarks are as below - (maximum at m2=7 and minimum at m2=12)
   gcc: For m2=7, misprediction rate = 26.65 & For m2=12, misprediction rate = 12.47
   jpeg: For m2=7, misprediction rate = 7.92 & For m2=12, misprediction rate = 7.60
   perl: For m2=7, misprediction rate = 21.31 & For m2=12, misprediction rate = 9.09
   As can be seen from above data, the average Misprediction Rate is the least for "jpeg benchmark".
4. The difference between highest and lowest values for the three benchmarks are as follows -
   gcc = 14.18 ; jpeg = 0.32 ; perl = 12.22

*** Similarities between benchmarks:**

1. As the size of the prediction table increases, the misprediction rate decreases.
2. The curve for all the cases goes towards attaining a saturation line. For jpeg, the saturation line seems to be reached already.
   For gcc, its nearing attaining saturation at 12, for perl, it seems the saturation line might be starting off from 12.

*** Differences between benchmarks -**

1. The minimum and maximum values which are at m=7 and m=12 respectively differ considerably for 3 benchmarks.
   For jpeg benchmark, the difference between the two values is very less.
   This means that for jpeg benchmark, the decrease in Misprediction rate gained by decreasing M is not as much as compared to others.
2. Considering the size of the Prediction table to be a major cost contributor, keeping M in lower values will be suitable for jpeg benchmark.
   For other benchmarks however, lowering M values helps in decreasing the misprediction rate a lot. For gcc and perl, the misprediction rate at m=12 is considerably less than the misprediction rate value at m=7.
3. The slope of the curve is highest for perl, then for gcc and lowest for jpeg.

Design:

1. It is given that there is a limit of 16KB of storage space for Prediction Table.

Assuming that each counter value takes 2 bits, the total number of unique counter entries that can fit in 16KB of storage is -
2^14 * 2^3 / 2 = 2^16 entries.
Therefore, we can have 2^16 different 2 bit entries in a 16KBytes storage.
To index these 2^16 entries, we need M=16bits.

2. Therefore, we will find values from M=7 to M=16 - for each benchmark.
   And also, for each benchmark, we will find values of Misprediction Rate till M=16

gcc benchmark:

1. Following are the values of m and the corresponding misprediction rate values. (M=13 to M=16 since M=7 to M=12 values are already explored and plotted in graph)

m	Misprediction Rate
13	11.72
14	11.37
15	11.30
16	11.21
17	11.19
18	11.17

2. Since for M>16, we overshoot the 16KBytes requirement, we will only base our decision on Misprediction Rate values till M=16.
3. From M=13 to M=16, the Misprediction rate decreases, but by a small amount.
4. The reason for this is that with increased size in Prediction Table, every static branch encountered gets a dedicated counter (no conflicts among branch PCs), thus making the 1-bit counter algorithms the accuracy limiter and not the table size.
5. **Optimal Size: M=16.**

jpeg benchmark:

1. Following are the values of m and the corresponding misprediction rate values. (M=13 to M=16 since M=7 to M=12 values are already explored and plotted in graph)

m	Misprediction Rate
13	7.59
14	7.59
15	7.59
16	7.59
17	7.59
18	7.59

2. As can be clearly seen here, there is no point in increasing M beyond 13 since the value of the Misprediction Rate remains the same.
3. This value indicates that every branch instruction has got a dedicated entry in the prediction table. So increasing size of prediction table will not reduce the

misprediction rate. We might have to modify the 2-bit counter algorithm to reduce misprediction rate further.

4. **Optimal Design: M=13**

perl benchmark:

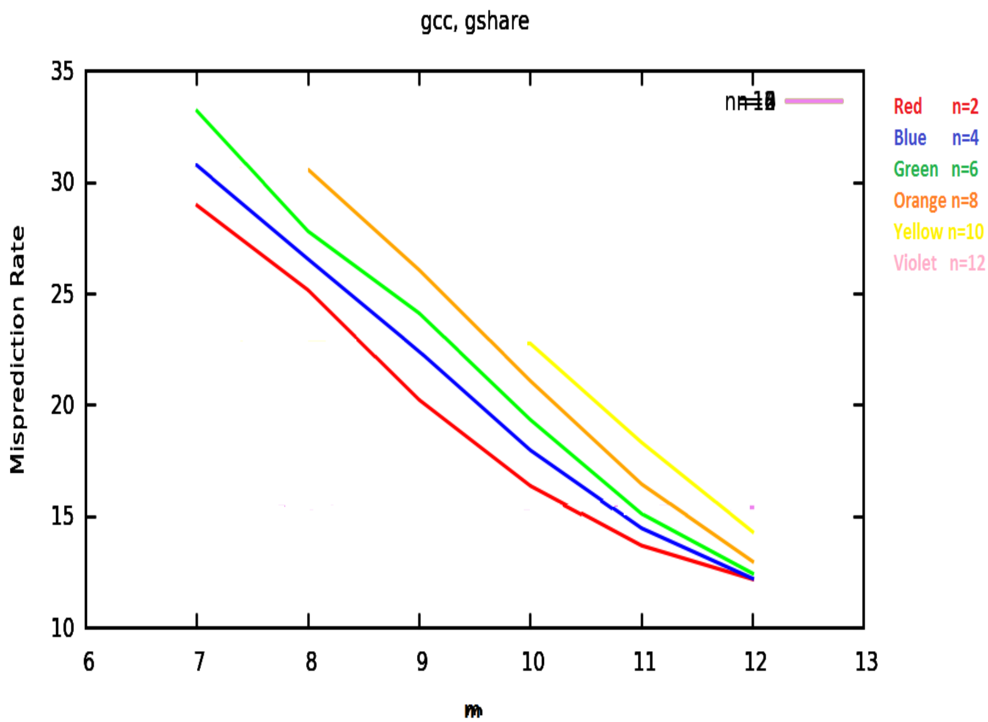
1. Following are the values of m and the corresponding misprediction rate values. (M=13 to M=16 since M=7 to M=12 values are already explored and plotted in graph)

m	Misprediction Rate
13	8.92
14	8.82
15	8.82
16	8.83
17	8.83
18	8.83

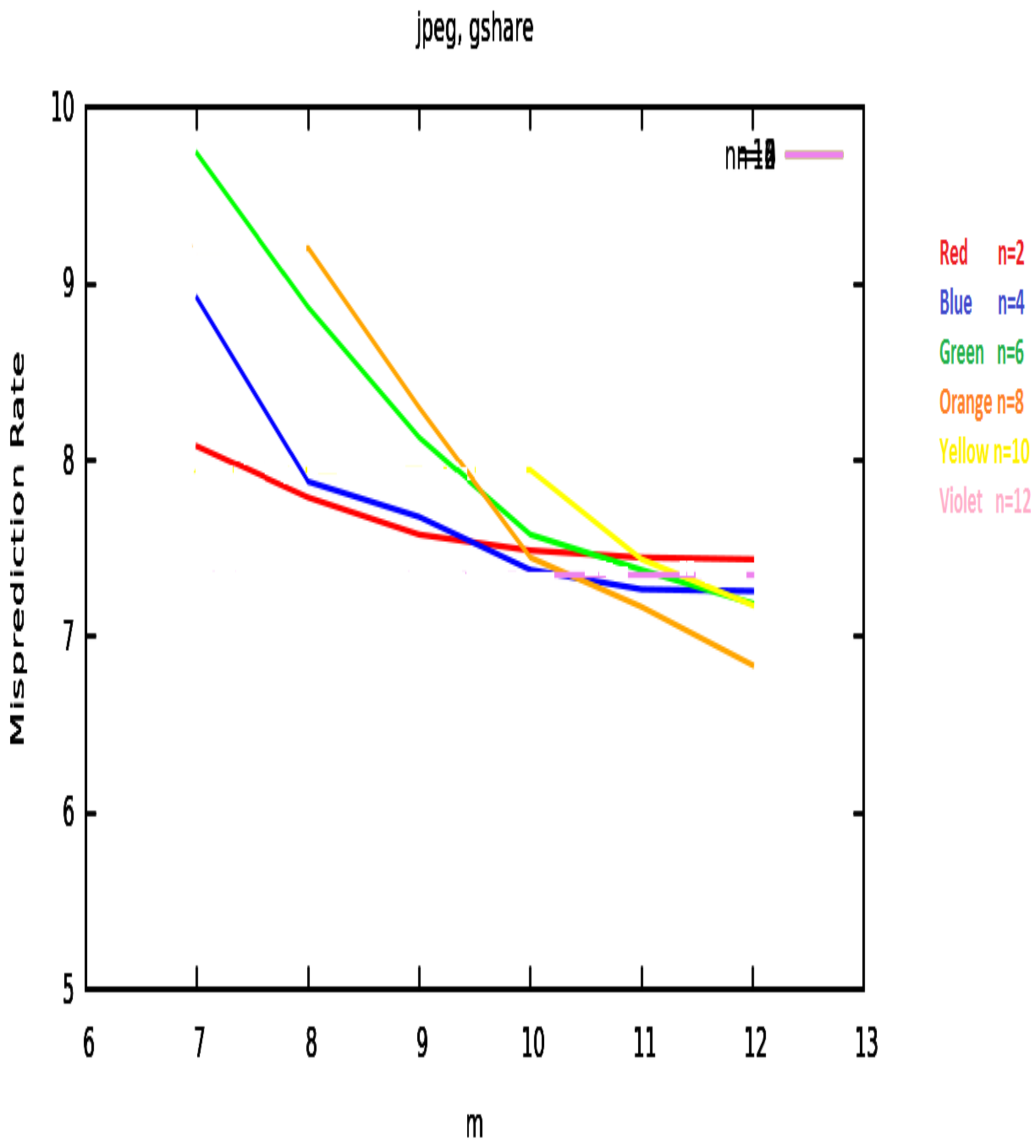
2. Here, the optimal design is when M=14.
3. **Optimal Design: M=14**
4. For perl benchmark, the misprediction value is slightly more for M=16 than for M=14 and 15.

GShare Predictor

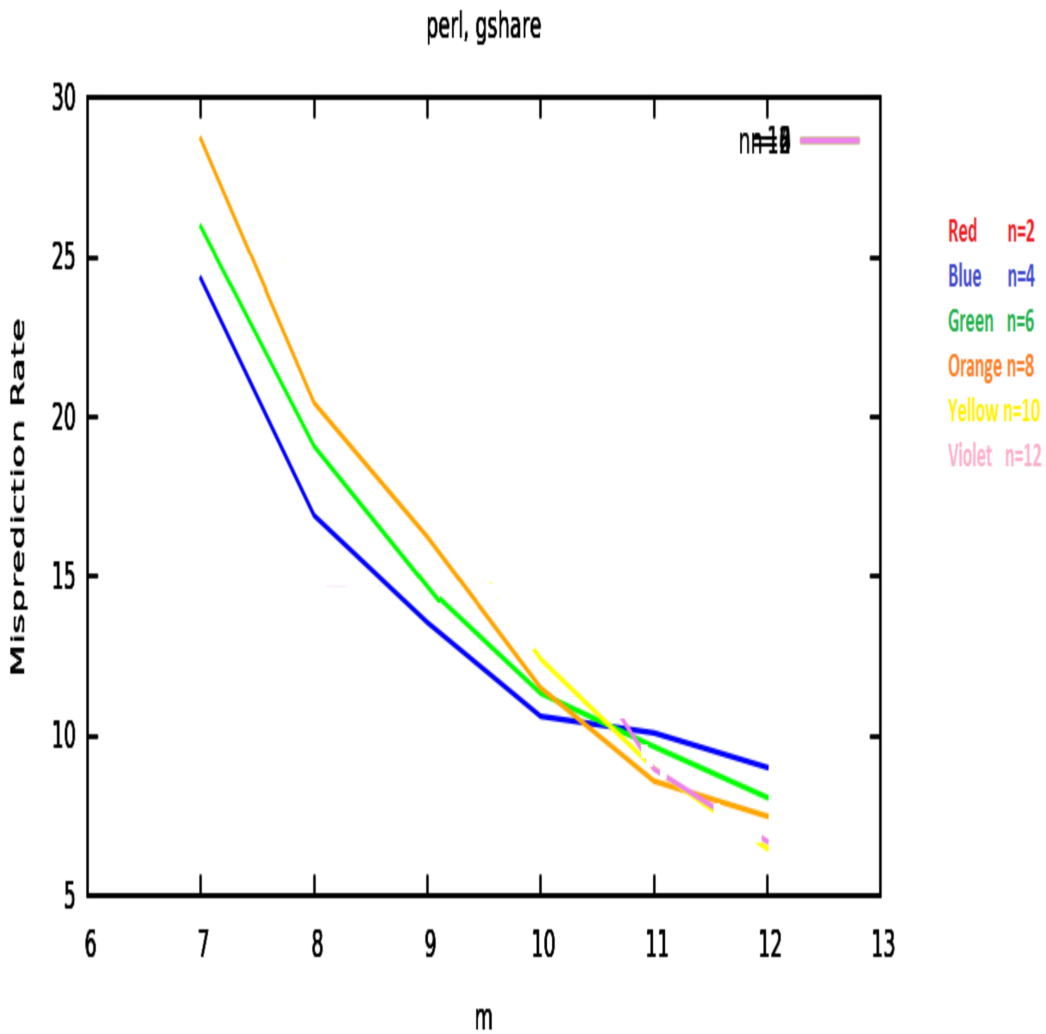
Graph1: gcc Benchmark:



Graph2: jpeg Benchmark:



Graph3: perl Benchmark:



Analysis:

*** Trends and Conclusions from Graphs:**

1. For given value of n, as m increases, the misprediction rate value decreases. In other words, keeping the size of BHR (Branch History Register) same, as the size of Prediction Table increases, the Misprediction Rate value decreases.
2. For given m, as n increases, the Misprediction Rate value also increases.
   In other words, keeping the prediction table size the same, as the size of BHR (Branch History Register) increases, the Misprediction Rate value increases.
3. Just like the Bimodal case, for each benchmark, each configuration tries to reach the saturation point where the misprediction rate will be minimum.
   The minimum value of misprediction rate in such cases will be determined by the counter algorithm since every branch will be assigned a dedicated entry in the prediction table. Thus, the optimal design should be the one where the misprediction rate is independent of the size of the prediction table.

*** Similarities between benchmarks-**

1. For given n, as m increases, the misprediction rate decreases.
2. For given m, as n increases, the misprediction rate increases.
3. The slope of all the curves (with each corresponding to a particular value of n is almost the same.

*** Differences between benchmarks-**

1. The jpeg benchmark hits the saturation case early i.e. for lesser values of m.
2. The difference between largest and smallest values of misprediction rate is different for different benchmarks. Its the least for jpeg and maximum for perl.

Design:

1. It is given that there is a limit of 16KB of storage space for Prediction Table.
2. Assuming that each counter value takes 2 bits, the total number of unique counter entries that can fit in 16KB of storage is -
   2^14 * 2^3 / 2 = 2^16 entries.
   Therefore, we can have 2^16 different 2 bit entries in a 16KBytes storage.
   To index these 2^16 entries, we need M=16bits.
   Therefore, we will find values from M=7 to M=16 - for each benchmark.
   And also, for each benchmark, we will find values of Misprediction Rate till M=16.
3. Since the size of BHR is negligible in comparison to size of the Prediction Table, we will not factor that in while

gcc benchmark:

m	n=2	n=4	n=6	n=8	n=10	n=12	n=14	n=16
13	11.11	10.57	10.59	11.00	11.68	12.68	-	-
14	10.42	09.69	09.69	09.34	09.83	09.48	11.13	-
15	10.13	09.13	08.30	08.22	08.46	09.01	09.48	-
16	09.93	08.77	07.89	07.57	07.61	07.86	08.34	08.75

Observed Trend: As n increases, for given m, sometimes, the value of Misprediction rate is decreasing. This is different than when the Sample space was for m=7 to m=12.

Optimal Design: M=16 & n=8 => Misprediction Rate = 7.57%

jpeg benchmark:

m	n=2	n=4	n=6	n=8	n=10	n=12	n=14	n=16
13	07.33	07.24	07.16	06.83	07.02	07.17	-	-
14	07.32	07.17	07.14	06.69	06.84	06.84	06.93	-
15	07.31	07.13	07.09	06.69	06.72	06.70	06.67	-
16	07.31	07.13	07.08	06.65	06.70	06.66	06.57	06.68

Observed Trend: As n increases, for given m, sometimes, the value of Misprediction rate is decreasing. This is different than when the Sample space was for m=7 to m=12.

Optimal Design: M=16 & n=8 => Misprediction rate = 6.65%

perl benchmark:

m	n=2	n=4	n=6	n=8	n=10	n=12	n=14	n=16
13	09.23	07.27	06.09	05.26	04.92	05.09	-	-
14	08.07	07.35	05.43	04.51	03.80	04.30	03.75	-
15	08.02	07.28	05.71	04.13	03.58	03.35	03.58	-
16	08.04	06.54	05.07	04.12	03.84	03.53	03.01	02.91

Observed Trend: As n increases, for given m, sometimes, the value of Misprediction rate is decreasing. This is different than when the Sample space was for m=7 to m=12.

Optimal Design: M=16 & N=16 => Misprediction Rate = 2.91%