NC State University

Department of Electrical and Computer Engineering

ECE 563: Fall 2017

Project #1: Cache Design, Memory Hierarchy Design

by

SOPAN PATRA

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

Student's electronic signature: _____SOPAN PATRA____
(sign by typing your name)

Course number: _____563_____
(463 or 563?)

9.1.4 Compare and contrast different benchmarks

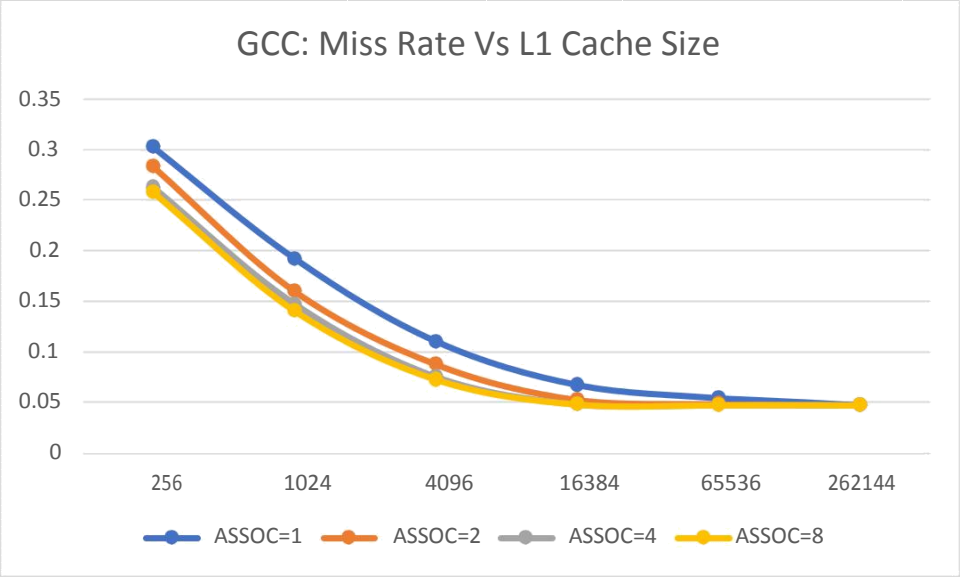
Simulation points are:

- Area budget for L1: 256KB
- Area budget for VC: 64KB
- Area budget for L2: 256KB
- Replacement Policy: LRU
- Block size: 16B

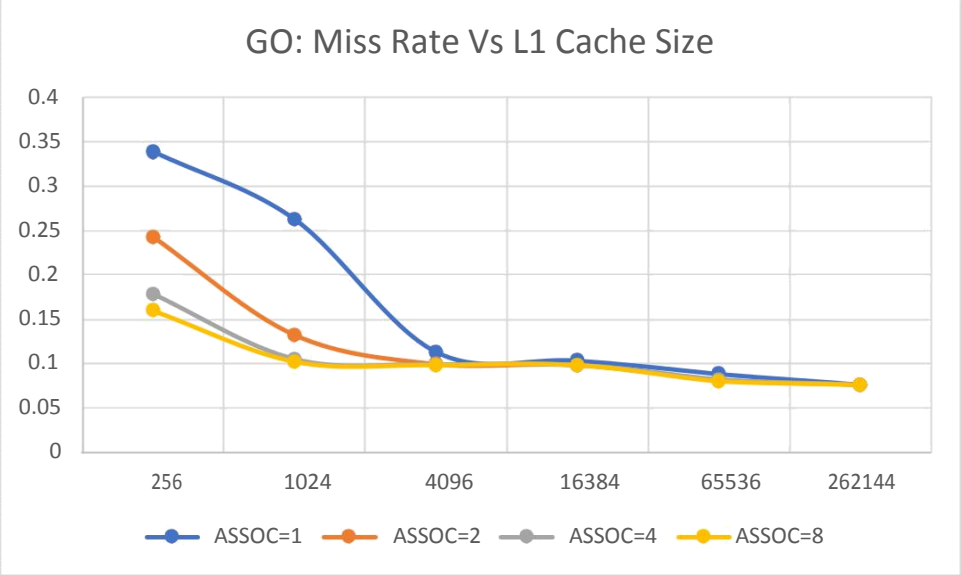
9.1.1 Exploring the effect of following parameters on overall performance of cache

1. L1 Cache Configuration Vs Miss Rate (Without L2 & VC)

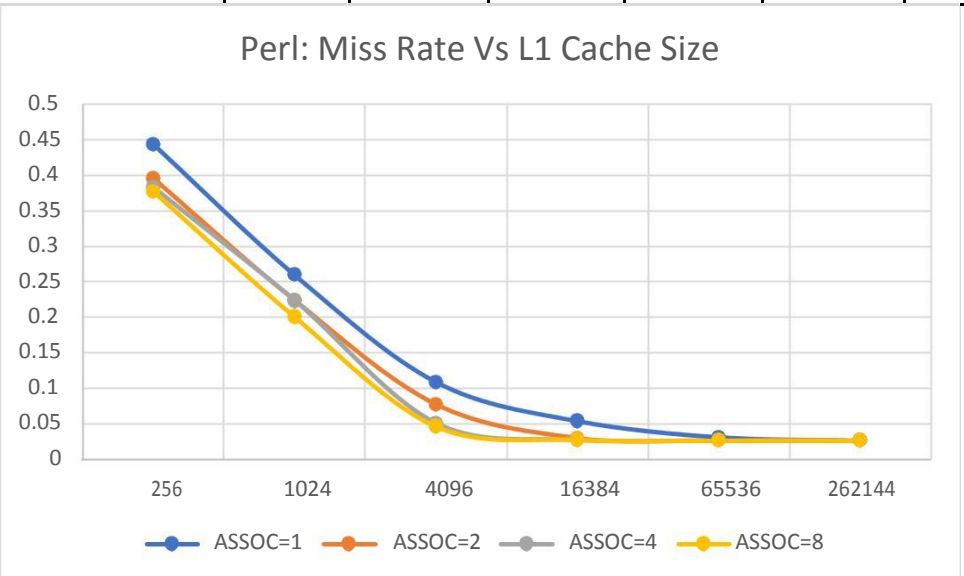
MISS RATE (gcc)						
L1: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.3026	0.1922	0.1102	0.0672	0.0546	0.0471
2	0.2829	0.1600	0.0875	0.0525	0.0485	0.0471
4	0.2627	0.1473	0.0755	0.0482	0.0472	0.0471
8	0.2577	0.1401	0.0727	0.0477	0.0472	0.0471



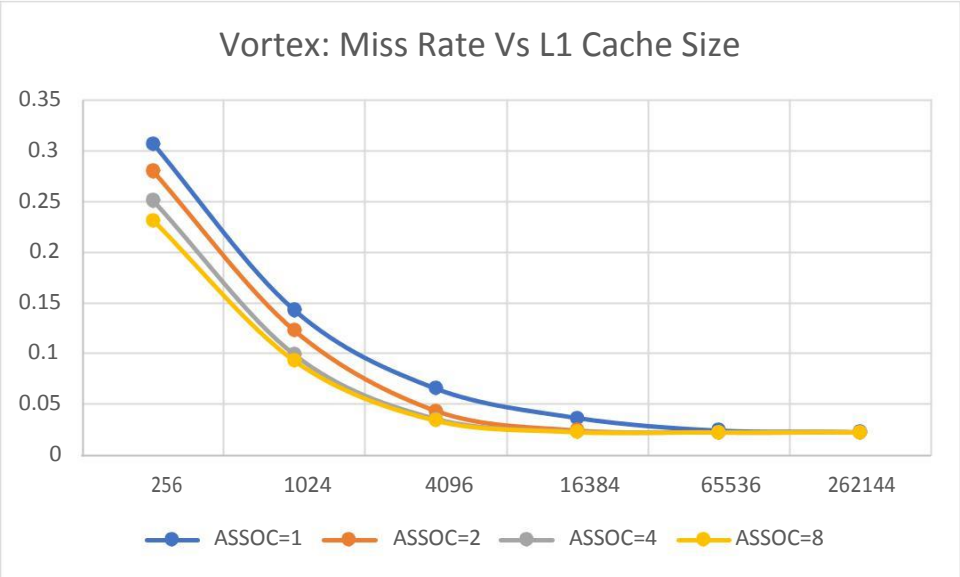
MISS RATE (go)						
L1: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.3385	0.2623	0.1128	0.1033	0.0880	0.0758
2	0.2424	0.1317	0.0994	0.0984	0.0810	0.0757
4	0.1780	0.1047	0.0986	0.0984	0.0819	0.0757
8	0.1597	0.1016	0.0985	0.0984	0.0803	0.0757



MISS RATE (perl)						
L1: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.4432	0.2599	0.1085	0.0536	0.0306	0.0261
2	0.3957	0.2236	0.0771	0.0297	0.0262	0.0261
4	0.3832	0.2236	0.0505	0.0273	0.0261	0.0261
8	0.3767	0.2004	0.0458	0.0270	0.0261	0.0261

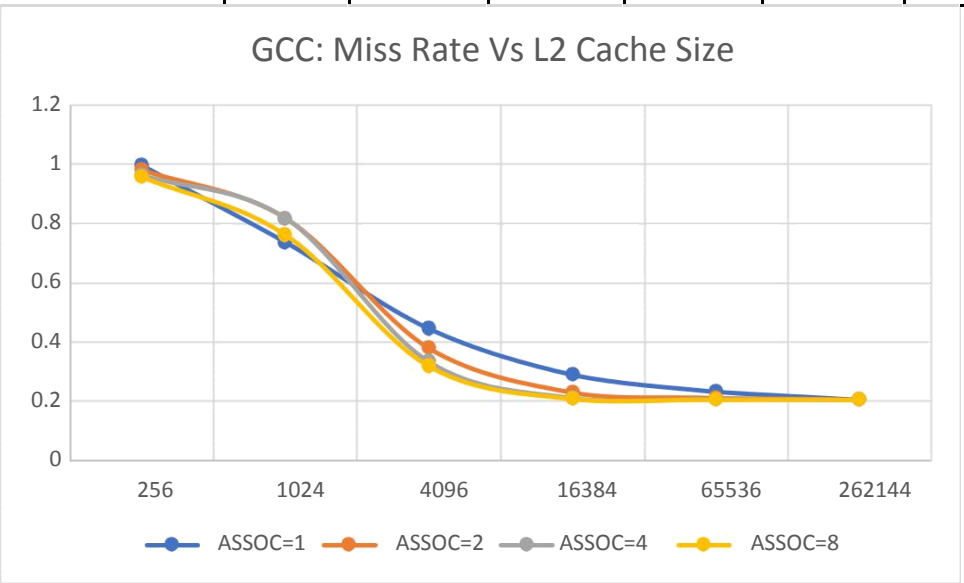


MISS RATE (vortex)						
L1: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.3067	0.1431	0.0654	0.0364	0.0240	0.0227
2	0.2799	0.1230	0.0436	0.0246	0.0222	0.0221
4	0.2510	0.0992	0.0357	0.0231	0.0221	0.0221
8	0.2313	0.0931	0.0339	0.0228	0.0221	0.0221

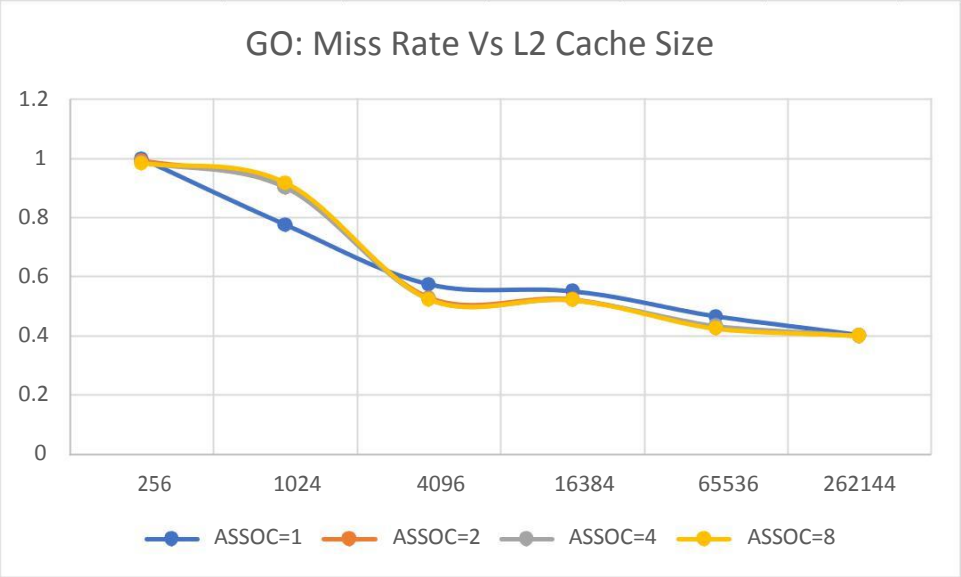


2. L2 Cache Configuration Vs Miss Rate (L1 size fixed at 4096B & 2-way assoc)

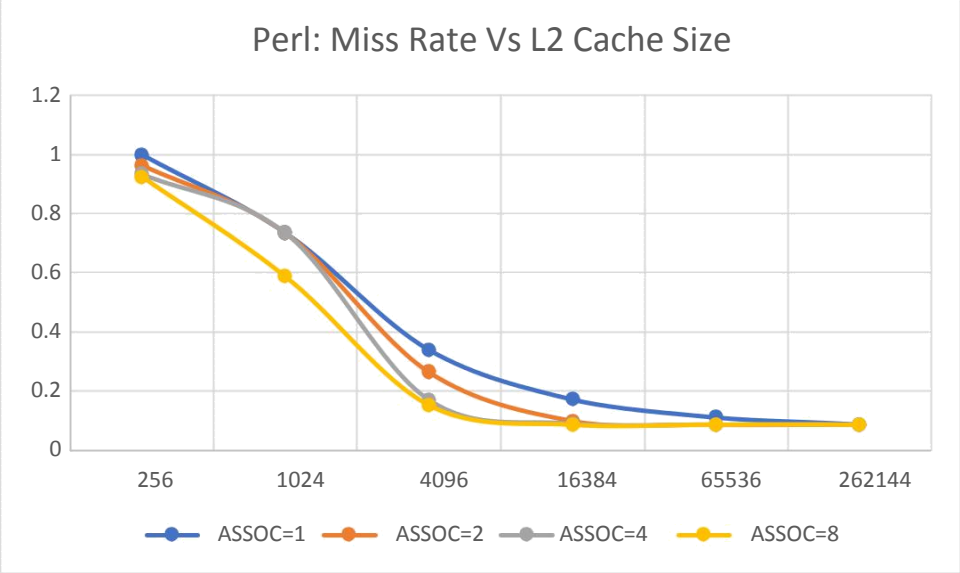
MISS RATE (gcc)						
L2: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.9960	0.7362	0.4443	0.2891	0.2321	0.2050
2	0.9814	0.8170	0.3787	0.2289	0.2108	0.2050
4	0.9630	0.8170	0.3326	0.2097	0.2053	0.2049
8	0.9560	0.7613	0.3168	0.2074	0.2050	0.2049



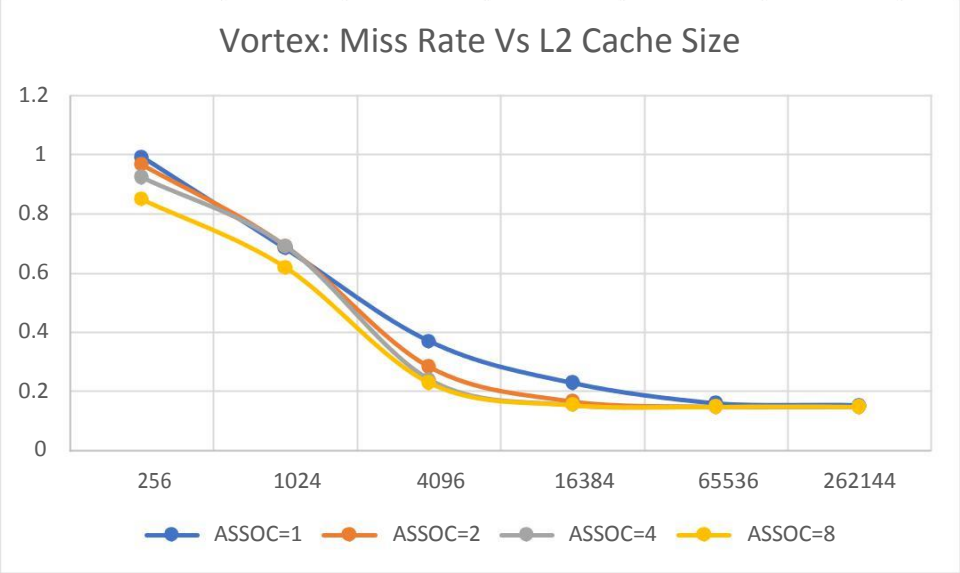
MISS RATE (go)						
L2: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.9976	0.7757	0.5732	0.5514	0.4660	0.4010
2	0.9927	0.9008	0.5296	0.5233	0.4280	0.4008
4	0.9857	0.9008	0.5240	0.5211	0.4334	0.4008
8	0.9837	0.9178	0.5226	0.5211	0.4243	0.4008



MISS RATE (perl)						
L2: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.9975	0.7346	0.3393	0.1718	0.1112	0.0857
2	0.9639	0.7344	0.2640	0.0981	0.0860	0.0857
4	0.9326	0.7344	0.1695	0.0899	0.0857	0.0857
8	0.9231	0.5873	0.1522	0.0857	0.0857	0.0857

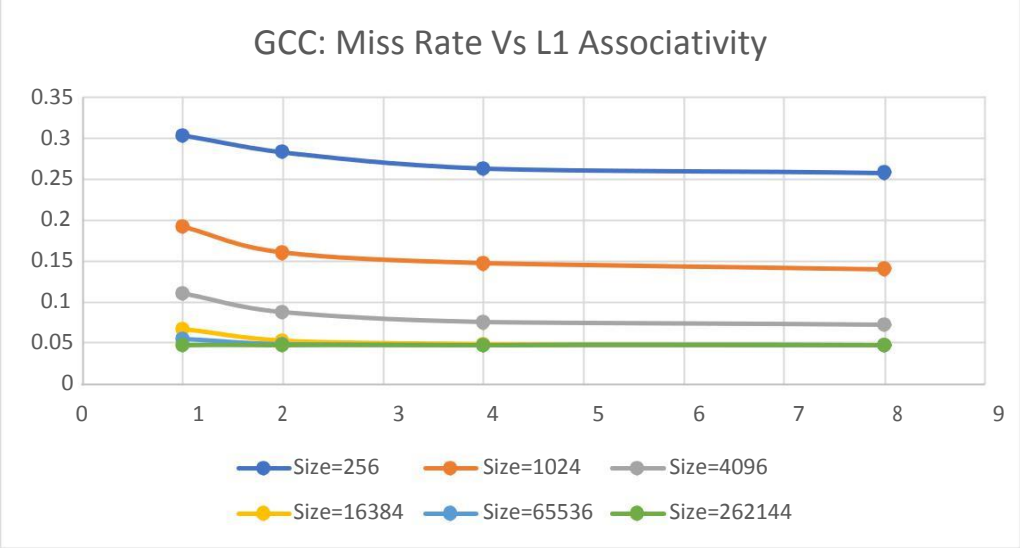


MISS RATE (vortex)						
L2: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.9913	0.6837	0.3685	0.2267	0.1601	0.1514
2	0.9669	0.6908	0.2826	0.1654	0.1479	0.1472
4	0.9241	0.6908	0.2398	0.1535	0.1472	0.1472
8	0.8482	0.6185	0.2272	0.1519	0.1472	0.1472

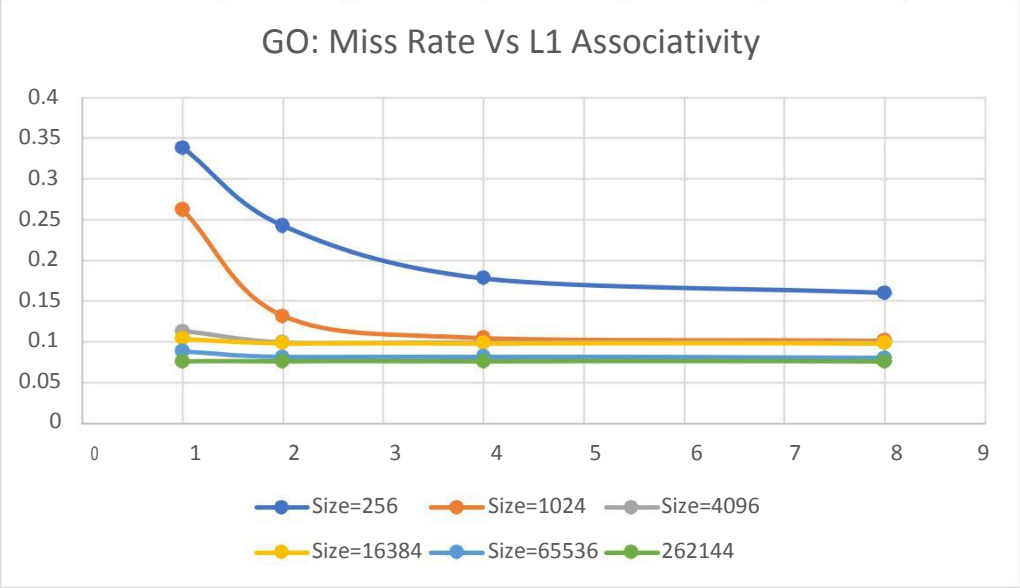


3. Associativity Vs Miss Rate

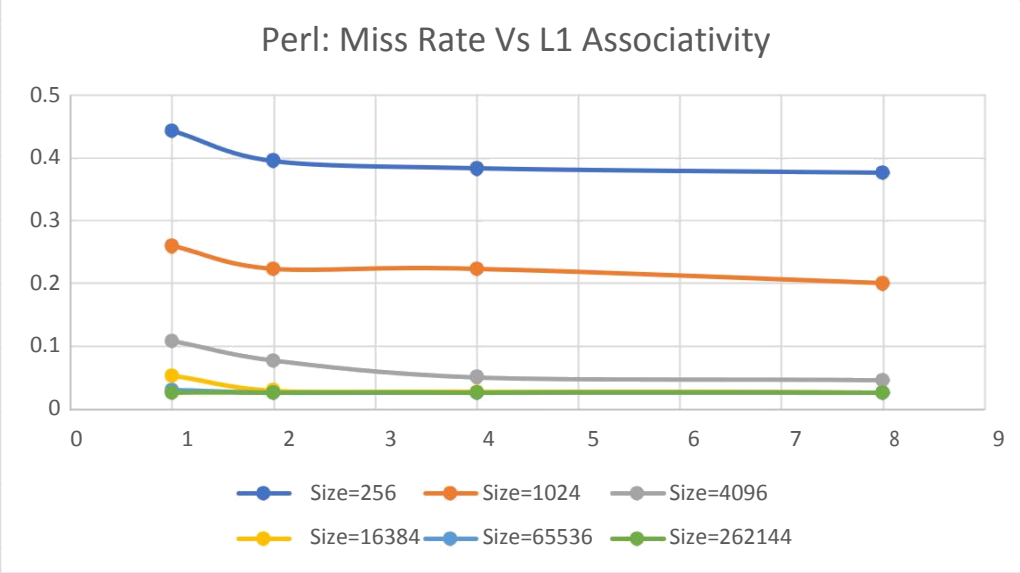
MISS RATE (gcc)						
L1: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.3026	0.1922	0.1102	0.0672	0.0546	0.0471
2	0.2829	0.1600	0.0875	0.0525	0.0485	0.0471
4	0.2627	0.1473	0.0755	0.0482	0.0472	0.0471
8	0.2577	0.1401	0.0727	0.0477	0.0472	0.0471



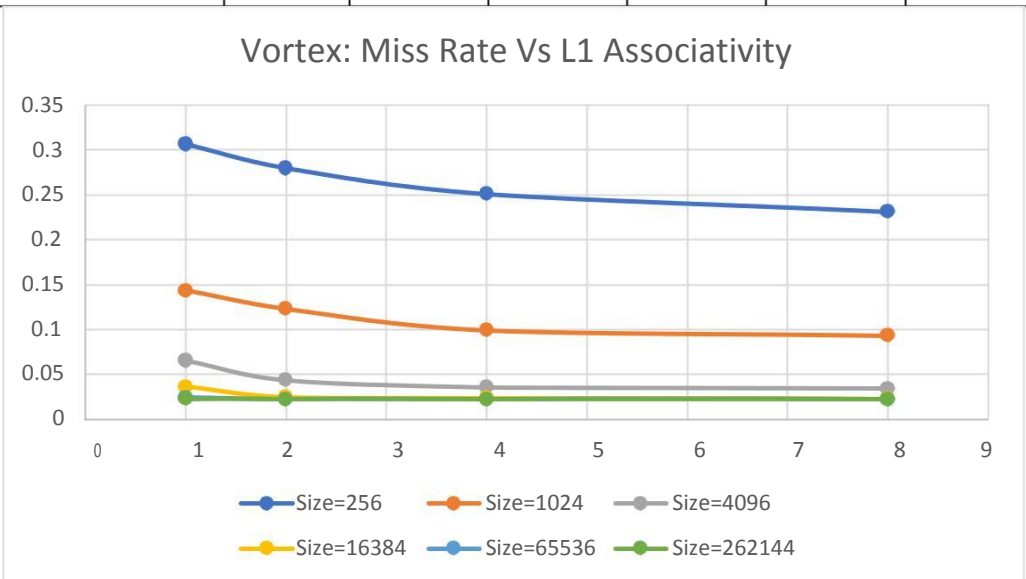
MISS RATE (go)						
L1: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.3385	0.2623	0.1128	0.1033	0.0880	0.0758
2	0.2424	0.1317	0.0994	0.0984	0.0810	0.0757
4	0.1780	0.1047	0.0986	0.0984	0.0819	0.0757
8	0.1597	0.1016	0.0985	0.0984	0.0803	0.0757



MISS RATE (perl)						
L1: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.4432	0.2599	0.1085	0.0536	0.0306	0.0261
2	0.3957	0.2236	0.0771	0.0297	0.0262	0.0261
4	0.3832	0.2236	0.0505	0.0273	0.0261	0.0261
8	0.3767	0.2004	0.0458	0.0270	0.0261	0.0261

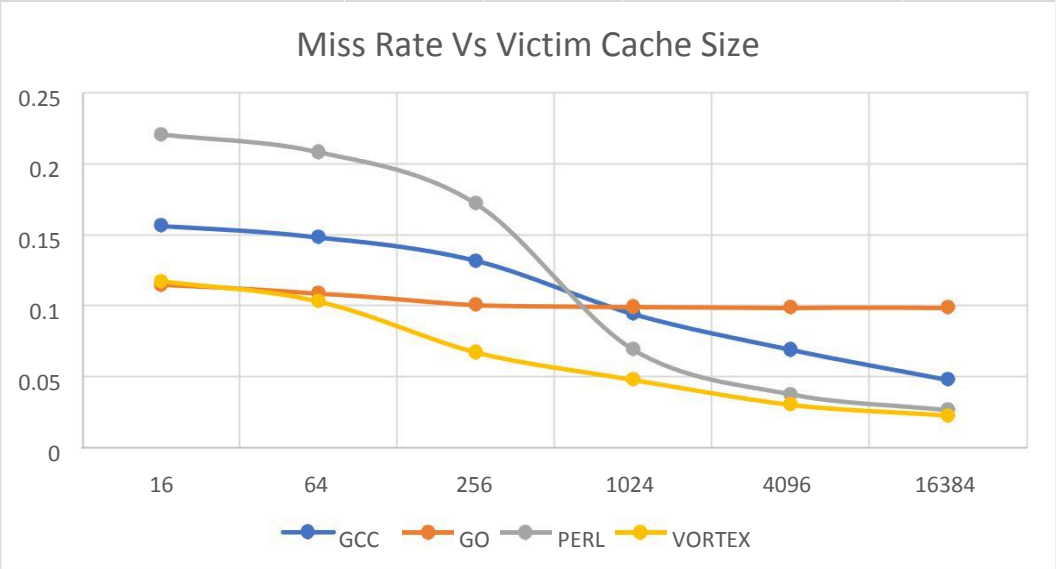


MISS RATE (vortex)						
L1: Assoc\Size	256	1024	4096	16384	65536	262144
1	0.3067	0.1431	0.0654	0.0364	0.0240	0.0227
2	0.2799	0.1230	0.0436	0.0246	0.0222	0.0221
4	0.2510	0.0992	0.0357	0.0231	0.0221	0.0221
8	0.2313	0.0931	0.0339	0.0228	0.0221	0.0221



4. Victim Cache Vs Miss Rate
   L1: Size = 1024, Assoc = 2
   L2: Size = 2048, Assoc = 4

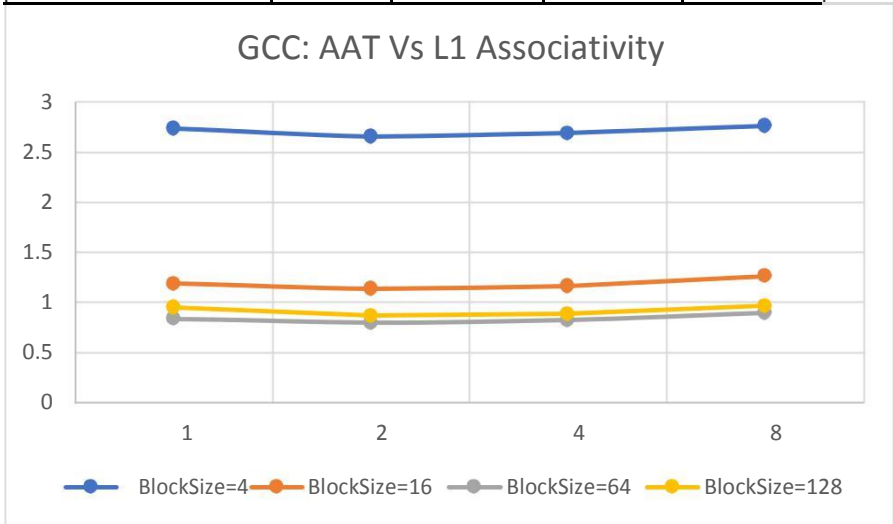
MISS RATE						
File\VC Size	16	64	256	1024	4096	16384
gcc	0.1560	0.1480	0.1314	0.0939	0.0685	0.0475
go	0.1143	0.1086	0.1003	0.0986	0.0984	0.0984
perl	0.2203	0.2079	0.1716	0.0690	0.0372	0.0262
vortex	0.1168	0.1029	0.0668	0.0476	0.0299	0.0223



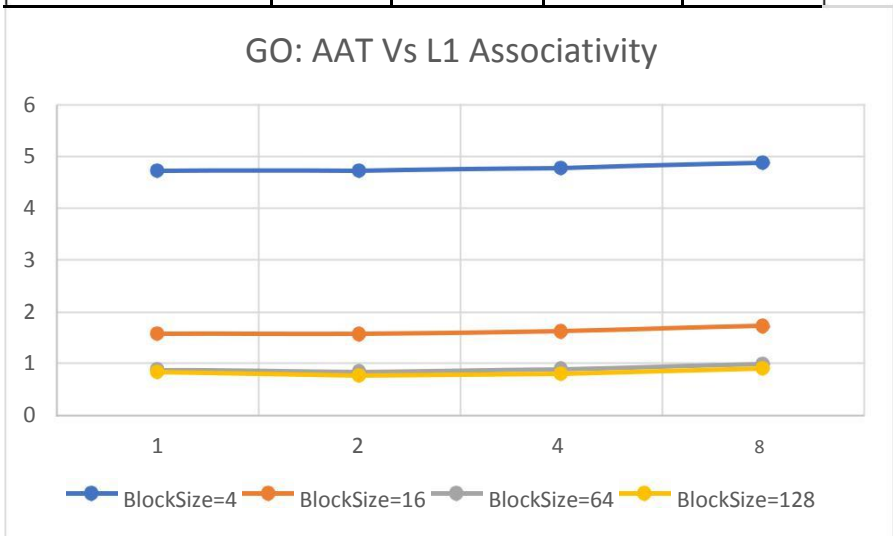
9.1.2 Thoroughly explore the design space and discuss noteworthy trends

1. L1 Configuration Vs AAT (without VC)
   L1: Size = 8kB; VC: Size = 8B; L2: Size = 64kB, Assoc = 4

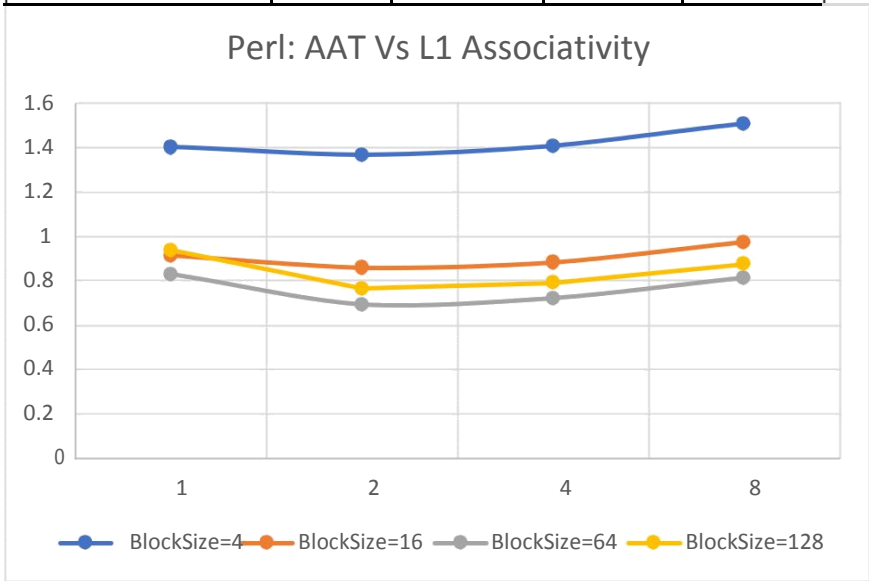
AAT gcc				
Blocksize/Assoc	1	2	4	8
4	2.7376	2.6612	2.6896	2.7623
16	1.1848	1.1349	1.1651	1.2611
64	0.8349	***0.7936***	0.8205	0.8944
128	0.9483	0.8665	0.8865	0.9631



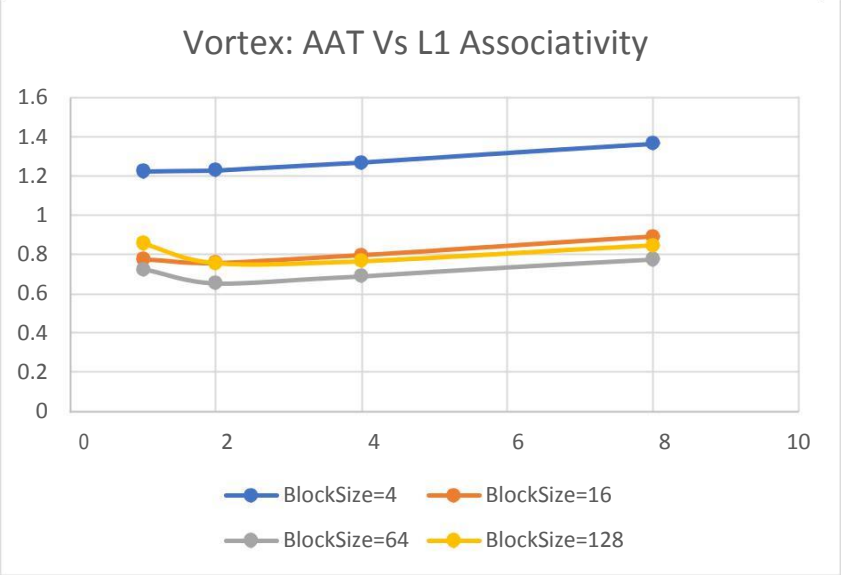
AAT go				
Blocksize/Assoc	1	2	4	8
4	4.7274	4.7252	4.7808	4.8816
16	1.5760	1.5696	1.6220	1.7225
64	0.8782	0.8413	0.8881	0.9883
128	0.8355	***0.7640***	0.8073	0.9076



AAT perl				
Blocksize/Assoc	1	2	4	8
4	1.4023	1.3669	1.4081	1.5078
16	0.9138	0.8595	0.8824	0.9743
64	0.8296	***0.6951***	0.7211	0.8137
128	0.9389	0.7657	0.7925	0.8762

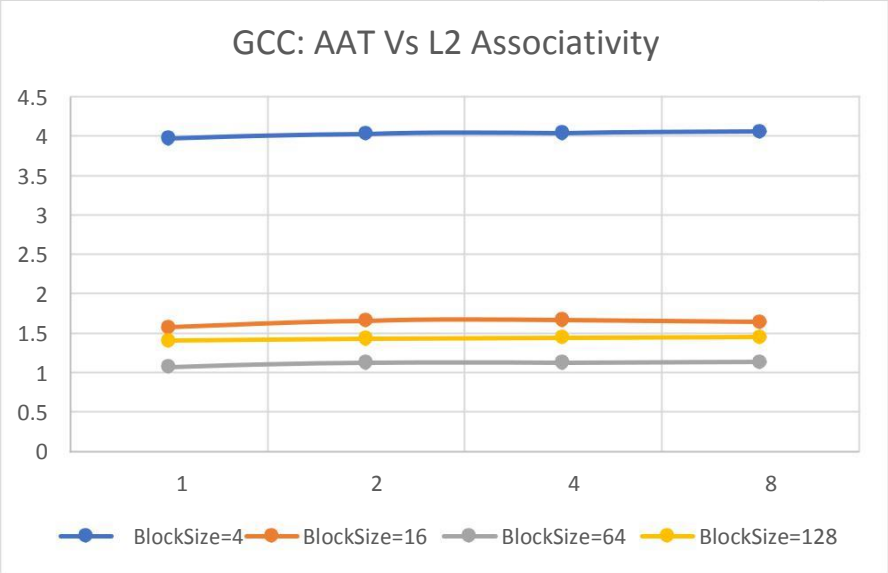


AAT vortex				
Blocksize/Assoc	1	2	4	8
4	1.2241	1.2294	1.2693	1.3654
16	0.7746	0.7566	0.7949	0.8910
64	0.7239	***0.6537***	0.6884	0.7754
128	0.8556	0.7543	0.7656	0.8452

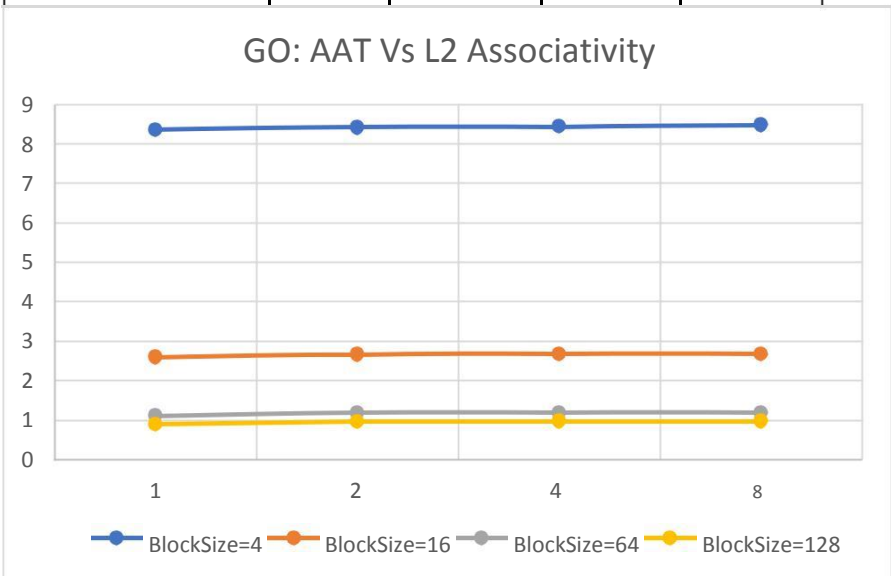


2. L2 Configuration Vs AAT (without VC)
   L1: Size = 8kB, Assoc = 4; VC: Size = 8B; L2: Size = 64kB

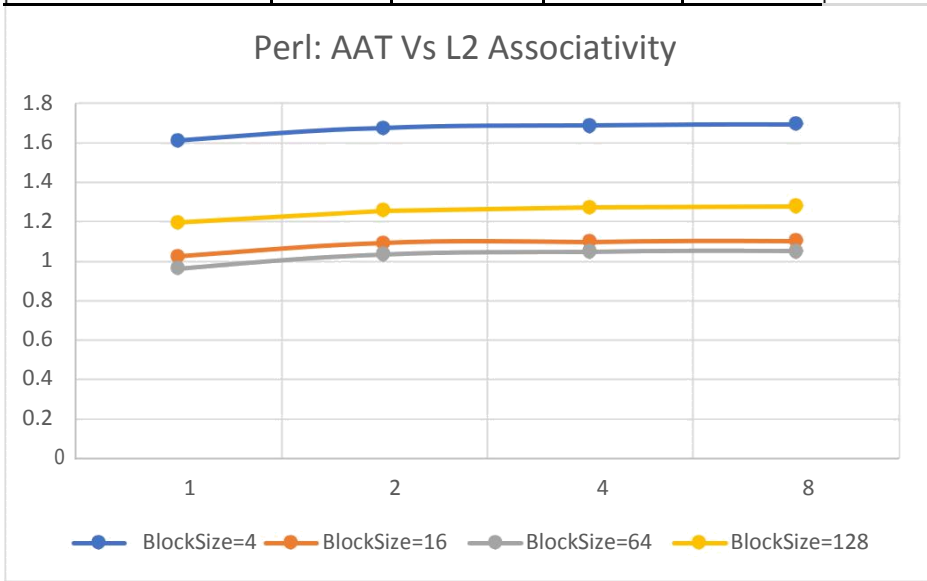
AAT gcc				
Blocksize/Assoc	1	2	4	8
4	3.9688	4.0330	4.0419	4.0588
16	1.5772	1.6607	1.6638	1.6436
64	***1.0708***	1.1208	1.1249	1.1324
128	1.4052	1.4273	1.4461	1.4495



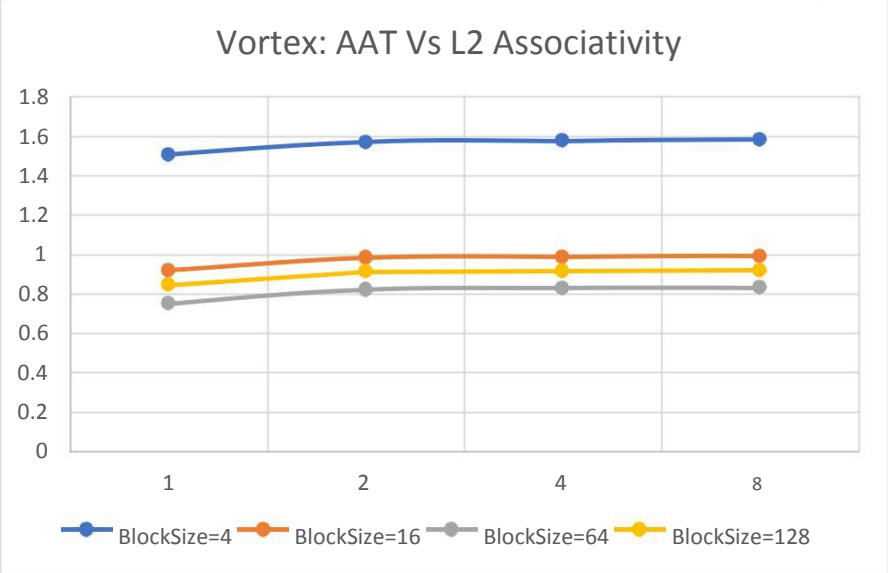
AAT go				
Blocksize/Assoc	1	2	4	8
4	8.3712	8.4285	8.4464	8.4819
16	2.5974	2.6677	2.6726	2.6825
64	1.1050	1.1788	1.1802	1.1831
128	***0.8848***	0.9590	0.9598	0.9612



AAT perl				
Blocksize/Assoc	1	2	4	8
4	1.6115	1.6759	1.6855	1.6935
16	1.0248	1.0924	1.0979	1.1026
64	***0.9625***	1.0345	1.0464	1.0497
128	1.1944	1.2576	1.2724	1.2770

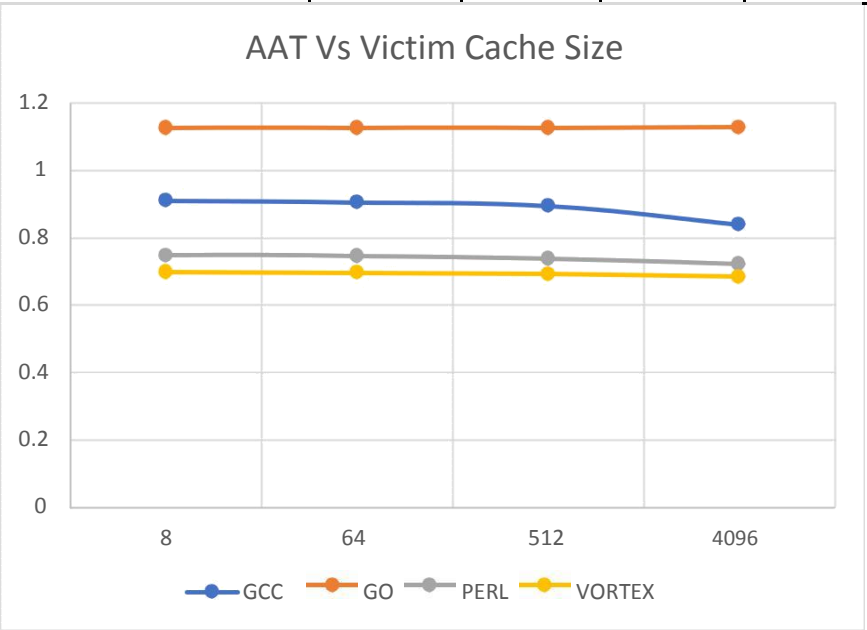


AAT vortex				
Blocksize/Assoc	1	2	4	8
4	1.5069	1.5703	1.5767	1.5823
16	0.9178	0.9825	0.9875	0.9920
64	***0.7490***	0.8218	0.8277	0.8297
128	0.8436	0.9111	0.9168	0.9196



3. Victim Cache Vs AAT L1:
   Size = 8kB, Assoc = 4
   L2: Size = 64kB, Assoc =
   4 Block Size = 32B

AAT Vs Victim Cache Size				
Access Time\VC Size	8	64	512	4096
gcc	0.9091	0.9054	0.8935	0.8379
go	1.1250	1.1251	1.1255	1.1277
perl	0.7482	0.7458	0.7375	0.7214
vortex	0.6980	0.6951	0.6925	0.6844



Conclusion summaries:

- For L1/L2, increasing "block size" will not decrease AAT all the time. Due to increased miss penalty, due to the larger block size.
- For L1/L2, increasing "associativity" will not reduce AAT at all times. Due to increased cache hit time (which is due to increased time to search a set/line)
- For victim cache, increasing the size reduces AAT all the time. Because, larger the victim cache size lesser the miss rate of L1

9.1.3 Find best memory hierarchy configuration

From the above conclusions, we can see that the best memory hierarchy configurations don't have huge block size or associativity. So, please find the best configurations for each trace file as per the tables above in 9.1.2 section.

1. gcc_trace.txt
   L1: Block Size = 64B &Associativity = 2-way
   L2: Block Size = 64B & Associativity = 1-way
   VC: Block Size = 4096B
2. go_trace.txt
   L1: Block Size = 128B &Associativity = 2-way
   L2: Block Size = 128B & Associativity = 1-way
   VC: Block Size = 4096B
3. perl_trace.txt
   L1: Block Size = 64B &Associativity = 2-way
   L2: Block Size = 64B & Associativity = 1-way
   VC: Block Size = 4096B
4. vortex_trace.txt
   L1: Block Size = 64B &Associativity = 2-way
   L2: Block Size = 64B & Associativity = 1-way
   VC: Block Size = 4096B

9.1.4 Compare and contrast different benchmarks

As per the above best memory hierarchy configuration, we can conclude that the benchmarks "gcc", "perl", "vortex" are of similar type, while "go" is of different type.

On further observation of the input .txt files, we can see all the address have the first 3 hex digits as same (400, i.e., first 12 bits of address = 0100 0000 0000) in the "go" benchmark. Meaning it uses lesser range of addresses more frequently=> needs more block size than other benchmarks for less access-time. This is reflected in having the best memory configuration at block size 128B than 64B, unlike other benchmarks. So, based on this we can also conclude that the additional dominant type of misses in "go" are compulsory miss and capacity miss. Whereas, the dominant type of misses in others are compulsory misses and capacity misses too =>
"go" uses spatial locality advantage where as other don't.

Block Size: 64B
L1: 8kB & Assoc = 2
L2: 64kB & Assoc = 1
VC: 8B

	Trace File		Dominant Miss Types
1.	gcc_trace.txt	=>	Compulsory miss, capacity miss
2.	go_trace.txt	=>	Compulsory miss, capacity miss(spatial)
3.	perl_trace.txt	=>	Compulsory miss, capacity miss
4.	vortex_trace.txt	=>	Compulsory miss, capacity miss