Mark Mitri 5/3/2022
ECE201 Project 4

Formulas Used:

$$Miss\ Rate\ =\ 1\ -\ Hit\ Rate\ =\ \frac{\frac{Misses}{1000\ Instructions}/1000}{\frac{Memory\ Access}{Instructions}}\ =\ \frac{X}{Y}$$

$$Hit\ Rate\ =\ 1\ -\ Miss\ Rate\ =\ \frac{\#\ of\ cache\ hits}{\#\ of\ cache\ hits\ +\ \#\ of\ cache\ misses}$$

Problem #1: Cache Miss Rate

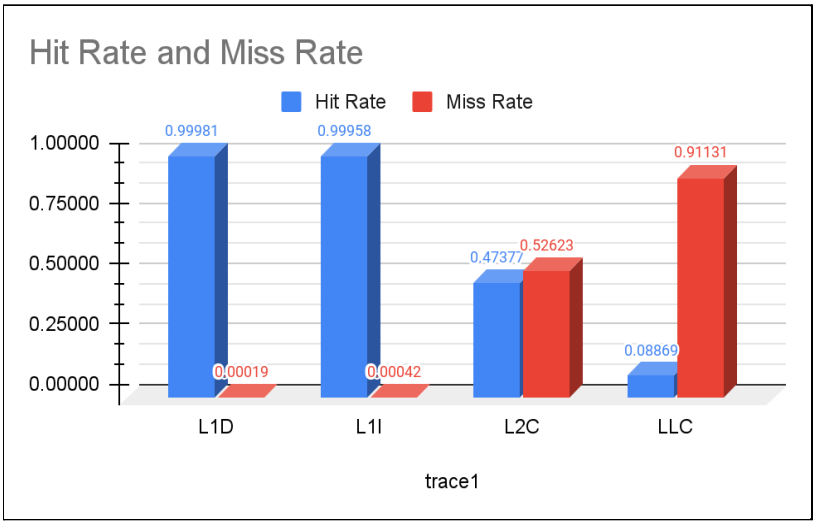
1. Using the baseline setting, collect and compare the cache miss rates of the L1-data, L1-instruction, L2, and last-level cache, for the 4 given traces.

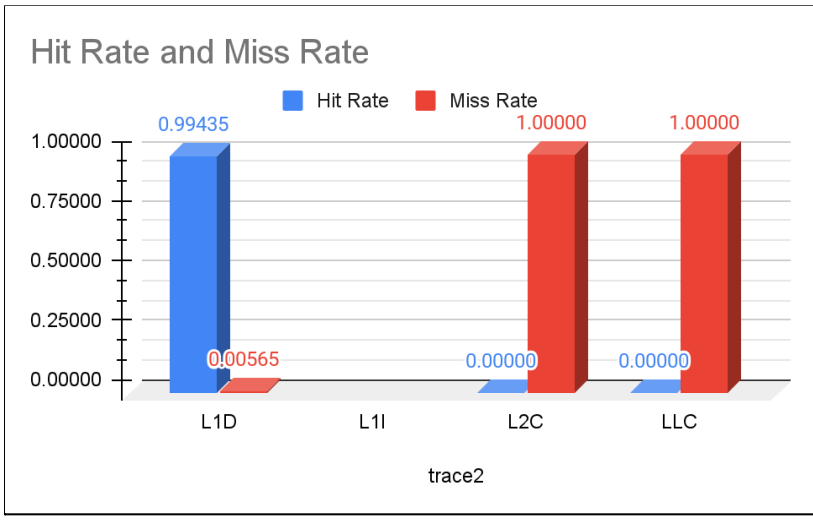
trace1	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1D	200000000	67587936	67575397	12539	0.99981	99.98145	0.00006	0.33794	0.00019	0.01855
L1I	200000000	9833672	9829570	4102	0.99958	99.95829	0.00002	0.04917	0.00042	0.04171
L2C	200000000	18452	8742	9710	0.47377	47.37698	0.00005	0.00009	0.52623	52.62302
LLC	200000000	10283	912	9371	0.08869	8.86901	0.00005	0.00005	0.91131	91.13099

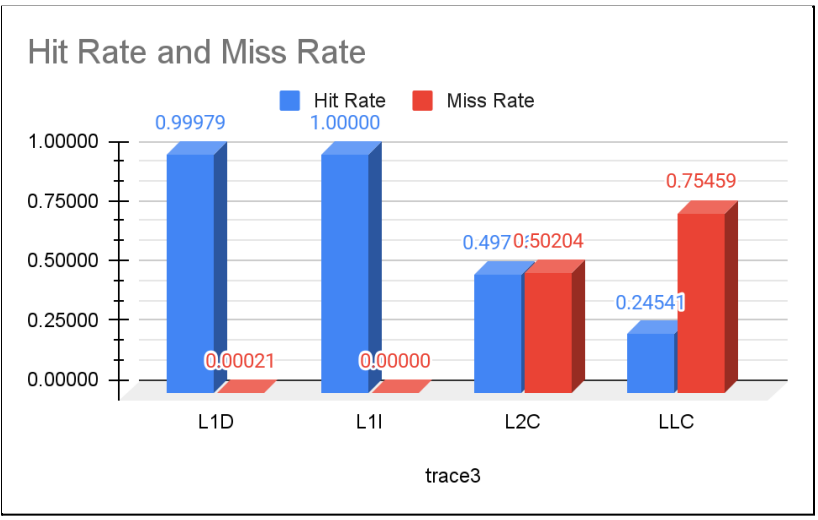
trace2	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate * 100
L1D	200000000	14818248	14734566	83682	0.99435	99.43528	0.00042	0.07409	0.00565	0.56472
L1I	200000000	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a
L2C	200000000	83682	0	83682	0.00000	0.00000	0.00042	0.00042	1.00000	100.00000
LLC	200000000	83682	0	83682	0.00000	0.00000	0.00042	0.00042	1.00000	100.00000

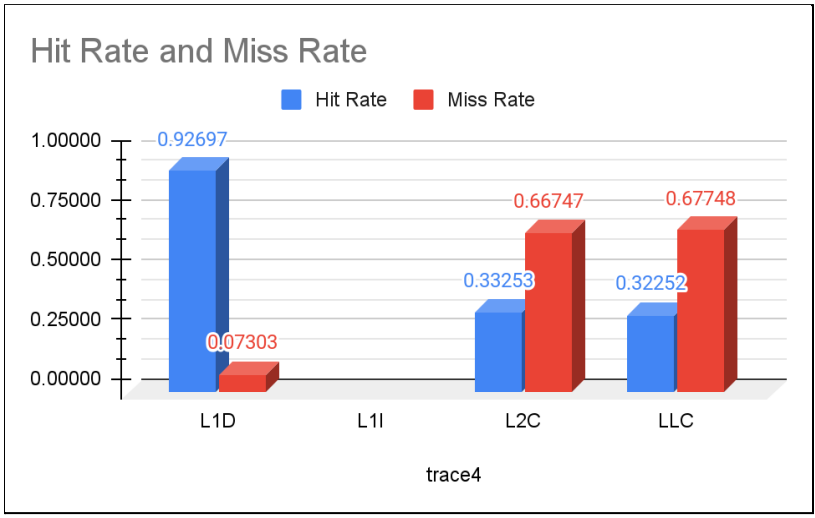
trace3	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate * 100
L1D	200000000	52440991	52430013	10978	0.99979	99.97907	0.00005	0.26220	0.00021	0.02093
L1I	200000000	11049058	11049054	4	1.00000	99.99996	0.00000	0.05525	0.00000	0.00004
L2C	200000000	21867	10889	10978	0.49796	49.79650	0.00005	0.00011	0.50204	50.20350
LLC	200000000	14543	3569	10974	0.24541	24.54102	0.00005	0.00007	0.75459	75.45898

trace4	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate * 100
L1D	200000000	76920374	71302839	5617535	0.92697	92.69695	0.02809	0.38460	0.07303	7.30305
L1I	200000000	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a
L2C	200000000	8228208	2736116	5492092	0.33253	33.25288	0.02746	0.04114	0.66747	66.74712
LLC	200000000	8009938	2583328	5426610	0.32252	32.25154	0.02713	0.04005	0.67748	67.74846









2. Qualitatively, does it make sense to split the L2 cache into L2-D and L2-I caches? Why or why not?

Splitting the L2 cache into the instruction and data portions decreases the latency by placing the instruction cache closer to the instruction fetch unit and the data cache close to the memory unit.[1] In addition, splitting the cache increases bandwidth by splitting the tasks that the cache executes.

[1] https://stackoverflow.com/questions/55752699/what-does-a-split-cache-means-and-how-is-it-usefulif-it-is

Problem #2: L1 Data Cache of 4 traces by varying the configuration

1. What is the performance impact of a slower L1 data cache? e.g. double the latency in champsim_config.json and check/compare the results.

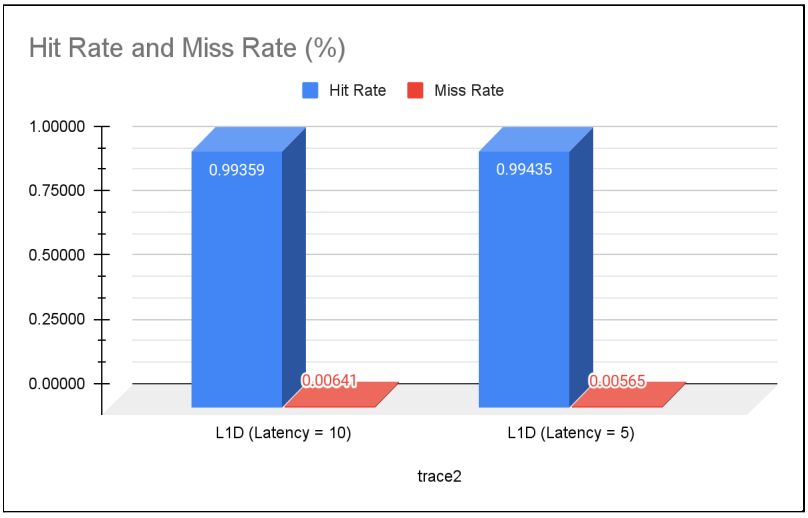
trace1	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1D (Latency = 10)	200000002	66011644	65999106	12538	0.99981	99.98101	0.00006	0.33006	0.00019	0.01899
L1D (Latency = 5)	200000000	67587936	67575397	12539	0.99981	99.98145	0.00006	0.33794	0.00019	0.01855

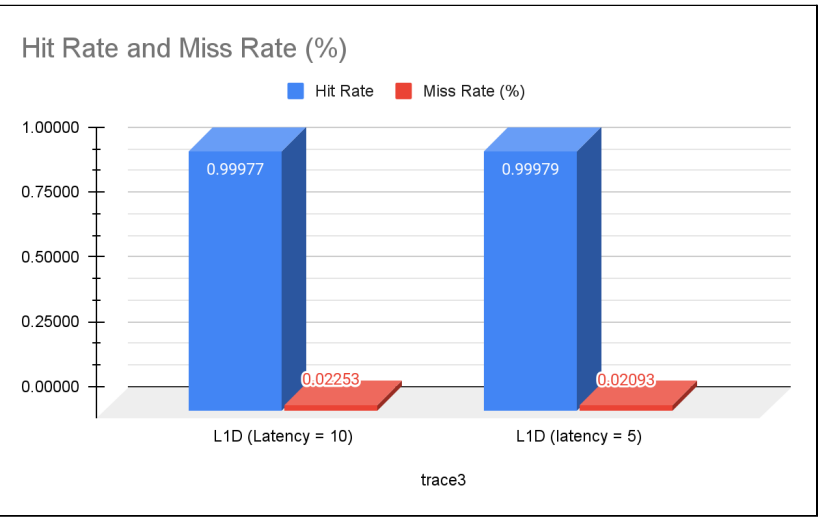
trace2	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1D (Latency = 10)	200000003	13050456	12966774	83682	0.99359	99.35878	0.00042	0.06525	0.00641	0.64122
L1D (Latency = 5)	200000000	14818248	14734566	83682	0.99435	99.43528	0.00042	0.07409	0.00565	0.56472

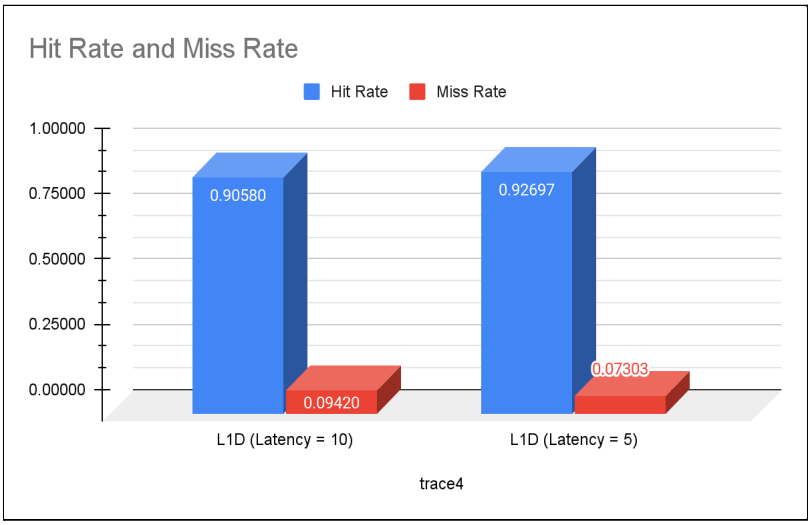
trace3	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1D (Latency = 10)	200000000	48715863	48704885	10978	0.99977	99.97747	0.00005	0.24358	0.00023	0.02253
L1D (latency = 5)	200000000	52440991	52430013	10978	0.99979	99.97907	0.00005	0.26220	0.00021	0.02093

trace4	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate * 100
L1D (Latency = 10)	200000004	59634625	54017091	5617534	0.90580	90.58008	0.02809	0.29817	0.09420	9.41992
L1D (Latency = 5)	200000000	76920374	71302839	5617535	0.92697	92.69695	0.02809	0.38460	0.07303	7.30305


Hit Rate and Miss Rate (%)
Hit Rate
Miss Rate (%)
1.00000
0.75000
0.50000
0.25000
0.00000
0.99981
0.01899
0.99981
0.01855
L1D (Latency = 10)
L1D (Latency = 5)
trace1


Hit Rate and Miss Rate (%)
Hit Rate
Miss Rate
1.00000
0.75000
0.50000
0.25000
0.00000
0.99359
0.00641
0.99435
0.00565
L1D (Latency = 10)
L1D (Latency = 5)
trace2


Hit Rate and Miss Rate (%)
Hit Rate
Miss Rate (%)
1.00000
0.75000
0.50000
0.25000
0.00000
0.99977
0.02253
0.99979
0.02093
L1D (Latency = 10)
L1D (latency = 5)
trace3


Hit Rate and Miss Rate
Hit Rate
Miss Rate
1.00000
0.75000
0.50000
0.25000
0.00000
0.90580
0.09420
0.92697
0.07303
L1D (Latency = 10)
L1D (Latency = 5)
trace4

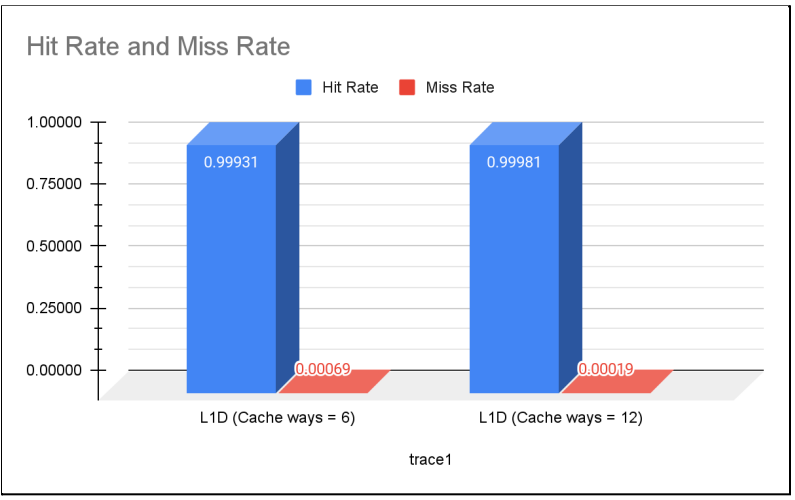
2. What is the performance impact of a smaller L1 data cache? e.g. decrease cache size by half (associativity) in champsim_config.json and check/compare the results.

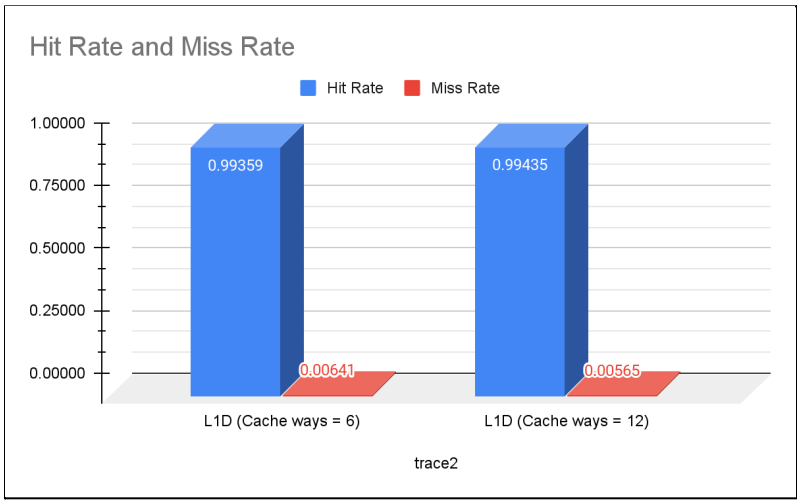
trace1	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1D (Cache ways = 6)	200000002	66005834	65960085	45749	0.99931	99.93069	0.00023	0.33003	0.00069	0.06931
L1D (Cache ways = 12)	200000000	67587936	67575397	12539	0.99981	99.98145	0.00006	0.33794	0.00019	0.01855

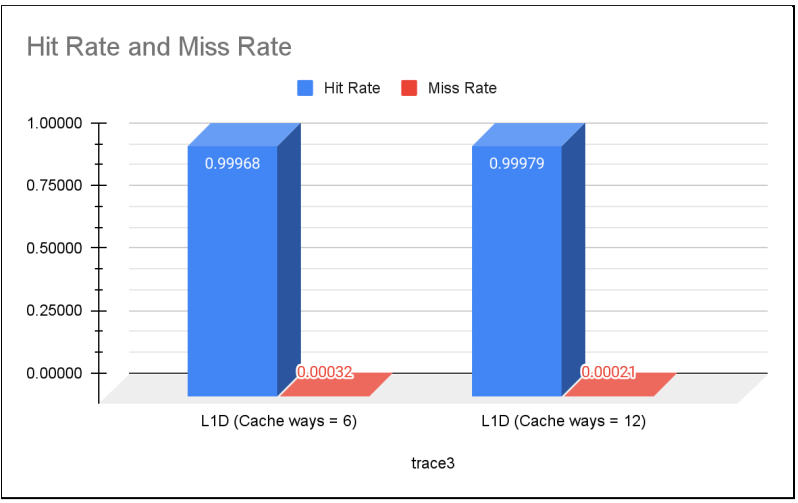
trace2	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1D (Cache ways = 6)	200000003	13050456	12966774	83682	0.99359	99.35878	0.00042	0.06525	0.00641	0.64122
L1D (Cache ways = 12)	200000000	14818248	14734566	83682	0.99435	99.43528	0.00042	0.07409	0.00565	0.56472

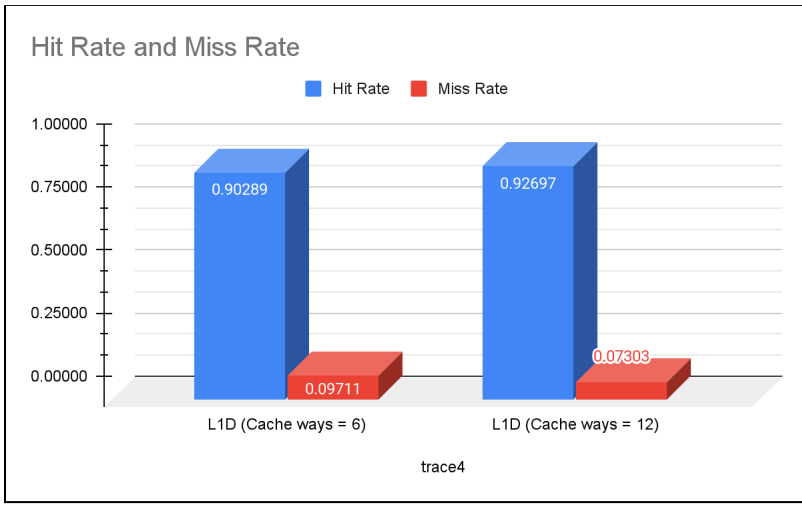
trace3	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1D (Cache ways = 6)	200000000	48716164	48700540	15624	0.99968	99.96793	0.00008	0.24358	0.00032	0.03207
L1D (Cache ways = 12)	200000000	52440991	52430013	10978	0.99979	99.97907	0.00005	0.26220	0.00021	0.02093

trace4	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1D (Cache ways = 6)	200000000	59620476	53830982	5789494	0.90289	90.28942	0.02895	0.29810	0.09711	9.71058
L1D (Cache ways = 12)	200000000	76920374	71302839	5617535	0.92697	92.69695	0.02809	0.38460	0.07303	7.30305









3. How important is the associativity of the L1 data cache? Conventional wisdom is that associativity is only important when the cache is small, is this true based on your results?

Based on my results, associativity (# of cache ways) is very important, if you want to minimize the number of misses. Because the L1 data cache is very small, if you can increase associativity, then you can maximize the number of accesses, and therefore, decrease the Miss Rate (shown in trace4's memory accesses and miss rate).

Problem #3: L1 Instruction Cache of 4 traces by varying the configuration

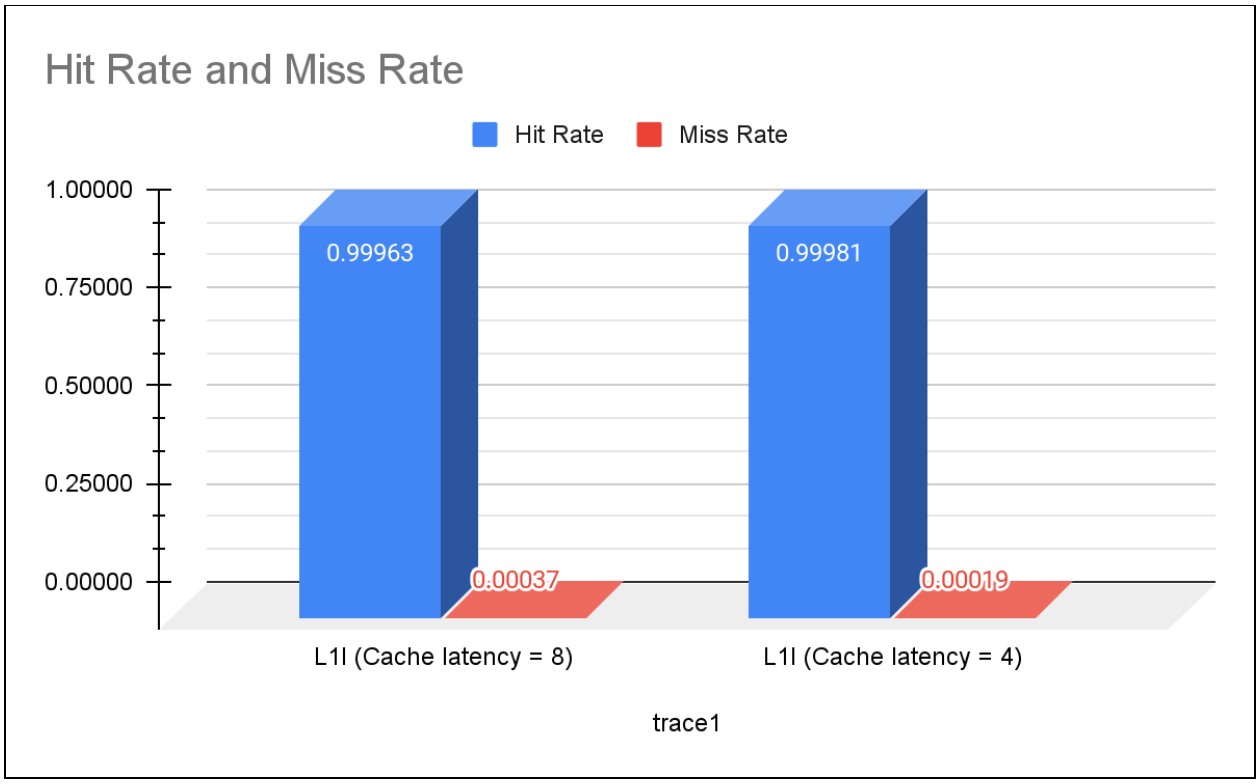
1. What is the performance impact of a slower L1 instruction cache? e.g. double the latency in champsim_config.json and check/compare the results.

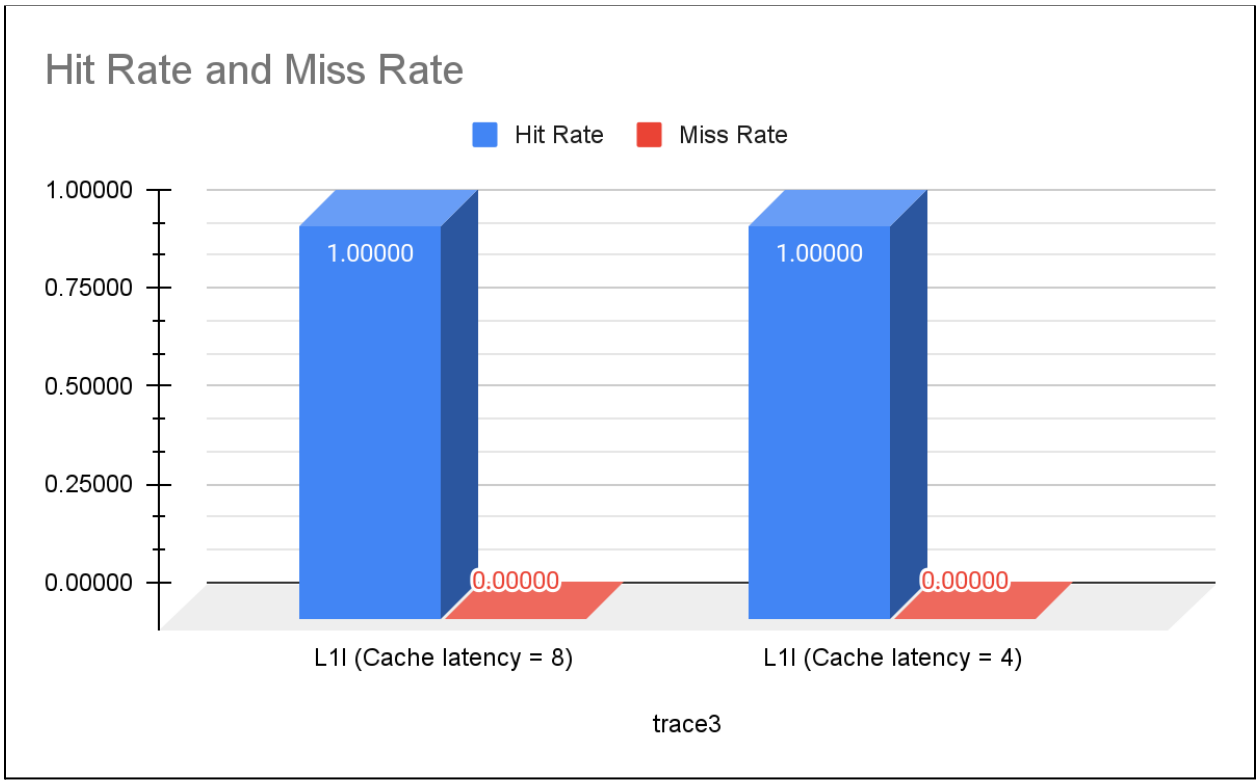
trace1	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1I (Cache latency = 8)	200000002	11124815	11120689	4126	0.99963	99.96291	0.00002	0.05562	0.00037	0.03709
L1I (Cache latency = 4)	200000000	9833672	9829570	4102	0.99981	99.98145	0.00006	0.33794	0.00019	0.01855

trace2	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1I (Cache latency = 8)	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a
L1I (Cache latency = 4)	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a

trace3	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1I (Cache latency = 8)	200000000	11764791	11764787	4	1.00000	99.99997	0.00000	0.05882	0.00000	0.00003
L1I (Cache latency = 4)	200000000	11049058	11049054	4	1.00000	99.99996	0.00000	0.05525	0.00000	0.00004

trace4	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1I (Cache latency = 8)	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a
L1I (Cache latency = 4)	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a


Hit Rate and Miss Rate
Hit Rate
Miss Rate
1.00000
0.75000
0.50000
0.25000
0.00000
0.99963
0.00037
0.99981
0.00019
L1I (Cache latency = 8)
L1I (Cache latency = 4)
trace1


Hit Rate and Miss Rate
Hit Rate
Miss Rate
1.00000
0.75000
0.50000
0.25000
0.00000
1.00000
0.00000
1.00000
0.00000
L1I (Cache latency = 8)
L1I (Cache latency = 4)
trace3

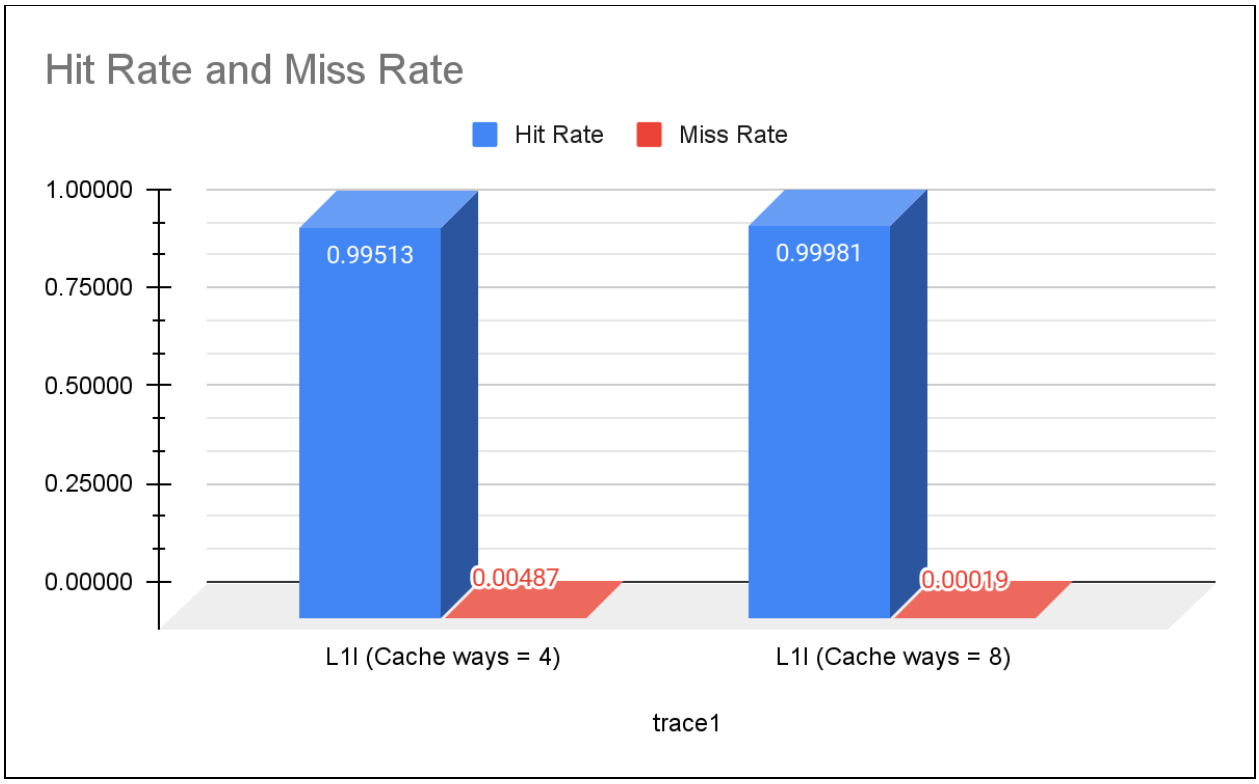
2. What is the performance impact of a smaller L1 instruction cache? decrease cache size by half (associativity) in champsim_config.json and check/compare the results.

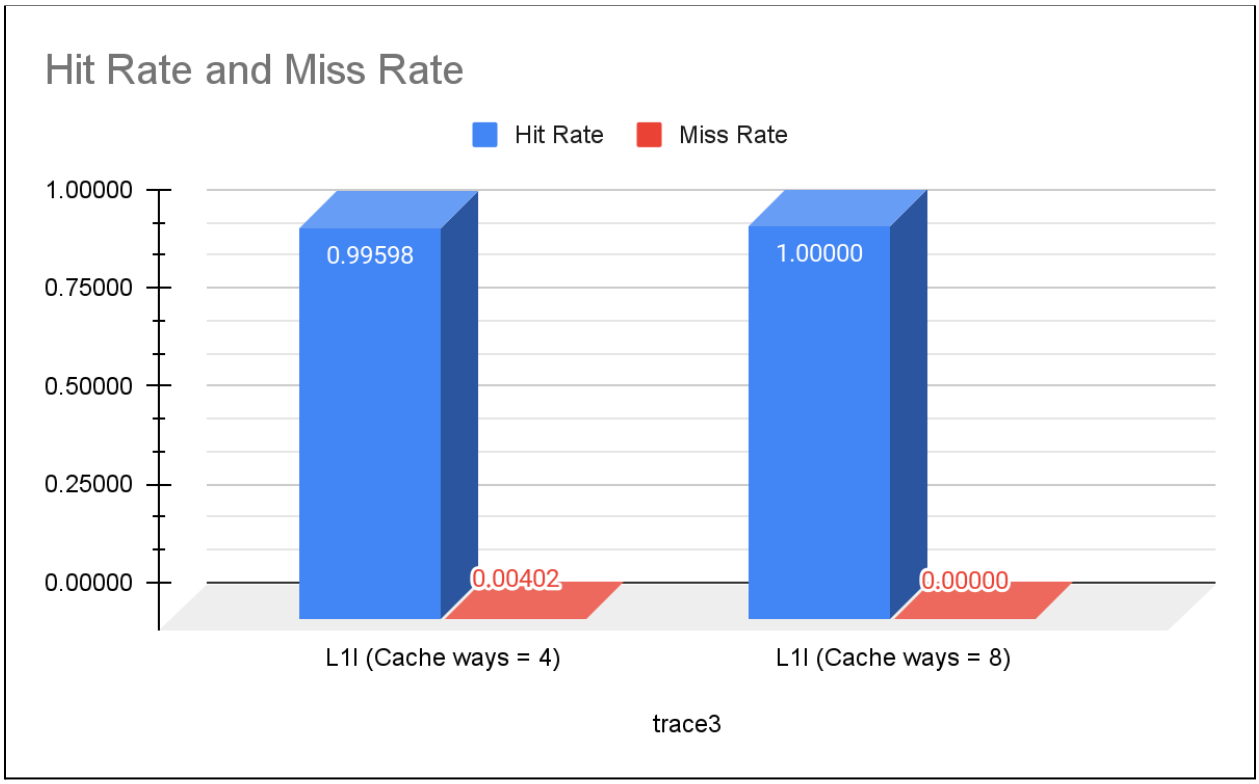
trace1	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1I (Cache ways = 4)	200000000	10444924	10394082	50842	0.99513	99.51324	0.00025	0.05222	0.00487	0.48676
L1I (Cache ways = 8)	200000000	9833672	9829570	4102	0.99981	99.98145	0.00006	0.33794	0.00019	0.01855

trace2	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1I (Cache ways = 4)	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a
L1I (Cache ways = 8)	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a

trace3	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1I (Cache ways = 4)	200000000	11053374	11008903	44471	0.99598	99.59767	0.00022	0.05527	0.00402	0.40233
L1I (Cache ways = 8)	200000000	11049058	11049054	4	1.00000	99.99996	0.00000	0.05525	0.00000	0.00004

trace4	# of Instructions	Memory Accesses	Hit	Miss	Hit Rate	Hit Rate (%)	X	Y	Miss Rate	Miss Rate (%)
L1I (Cache ways = 4)	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a
L1I (Cache ways = 8)	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a	N/a


Hit Rate and Miss Rate
Hit Rate
Miss Rate
1.00000
0.75000
0.50000
0.25000
0.00000
0.99513
0.00487
0.99981
0.00019
L1I (Cache ways = 4)
L1I (Cache ways = 8)
trace1


Hit Rate and Miss Rate
Hit Rate
Miss Rate
1.00000
0.75000
0.50000
0.25000
0.00000
0.99598
0.00402
1.00000
0.00000
L1I (Cache ways = 4)
L1I (Cache ways = 8)
trace3

3. How important is the associativity of the L1 instruction cache?

Higher associativity allows for more efficient utilization of cache but also increases the cost.[2] Because L1 IC holds the instruction of the process and not the data, if associativity decreases, this means there are fewer sets to hold instructions, thus decreasing the efficiency of execution i.e. fewer CPU executions over time. In general, as associativity increases, the miss rate decreases.[3]

[2] https://en.algorithmica.org/hpc/cpu-cache/associativity/
[3] https://www.ecs.umass.edu/ece/koren/architecture/Simplescalar/lab1caches.htm