CSE 590
Computer Architecture

Project 2
Analysis of Cache Parameters and Trade-offs on X86 Processor Using Gem5

Report

Submitted By

Karan Manchandia
(karanman)
Person # 50290755

Introduction and Steps for running the commands:

- This project aims to analyse the cache parameters using Gem5 on a X86 processor. The CPU benchmarks that we will be using to simulate the architectural design are:
    - 401.bzip2
    - 456.hmmer
    - 458.sjeng

- We would be using Gem5 in System Call Emulation Mode. To execute a benchmark program using SE mode, we will run the following command in the metallics server through the command prompt:

  Time util/gem5/build/X86/gem5.opt -d <output_directory> -I 100000000 util/gem5/configs/example/se.py -c <program> -o <argument> <parameters>

  <program> refers to the benchmark file in src directory and <argument> is for input data and <parameters> refer to the different values of parameters we applied. For this project we varied following parameters.

Steps for creating a new project directory in /home/csgrad/karanman in the metallics server:

- Download and Install WinSCP software on your Windows System.
- After, installation is complete open the WinSCP software.
- Now, select the file protocol as SFTP, type hostname as metallica.cse.buffalo.edu, set port number as 22, type your UB username (karanman) and password.
- Now, click on login to log in to the metallica server.
- Click on the New button and select directory.
- Type the directory name as CAp2 and click on OK.
- A new project directory would be with the location /home/csgrad/karanman/CAp2 in the metallica server.
- This would be the directory in which the project files after running each command would be saved.
- Note that after running each command we need to copy the result file generated from the server to our local computer, before running the next command.

- The seven parameters on which our analysis will be based are shown below:

1. L1D Cache Size.
2. L1I Cache Size.
3. L2 Cache Size.
4. L1D Associativity
5. L1I Associativity
6. L2 Associativity
7. Block size

8. Benchmarks.

- We have varied the values of each of the 7 parameters 5 different times and this has been done for all the 3 benchmarks. So in all we have got 105 stats.txt files. For each of these cases we have noted/calculated the following: L1D hit rate, L1I hit rate, L2 hit rate, L1D miss rate, L1I miss rate, L2 miss rate and CPI.

Calculating the following Parameters:

- Hit rate of L1 D cache
  - We can find the miss rate of L1 D Cache from the stats.txt file by searching the following:
  - L1 D miss rate -> system.cpu.dcache.overall_miss_rate::total
  - For calculating the hit rate, we can use the formula hit rate = 1-miss rate.

- Hit rate of L1 I cache
  - We can find the miss rate of L1 I Cache from the stats.txt file by searching the following:
  - L1 I miss rate -> system.cpu.icache.overall_miss_rate::total
  - For calculating the hit rate, we can use the formula hit rate = 1 - miss rate.

- Hit rate of L2 cache
  - We can find the miss rate of L2 Cache from the stats.txt file by searching the following:
  - L2 miss rate -> system.l2.overall_miss_rate::total
  - For calculating the hit rate, we can use the formula hit rate = 1 - miss rate.

- CPI
  - CPI can be calculated by using the formula

$$CPI = 1 + \frac{(IL1.miss\_num + DL1.miss\_num) \times 6 + L2.miss\_num \times 50}{Total\_Inst\_num}$$

  - Here, IL1.miss_num, DL1.miss_num and L2.miss_num can be found in the stats.txt file by searching the following:
  - L1 D Miss number -> system.cpu.dcache.overall_misses::total
  - L1 I Miss number -> system.cpu.icache.overall_misses::total
  - L2 miss number -> system.l2.overall_misses::total
  - After getting these values the formula can be used to calculate the CPI.

Logging into the Metallica server from the command prompt and run the commands:

- Open command prompt on your system and type the command:
  ssh karanman@metallica.cse.buffalo.edu
- The command prompt will display a message: Are you sure you want to continue.
- Enter Yes.
- You will be asked to enter your UBIT password. Enter the password.

- Now change directory using the command cd /util/gem5
- Now inside metallica {/util/gem5}. Type the command: setenv PATH {$PATH}:"util/gcc/bin"
- Type the next command: setenv LD_LIBRARY_PATH {$LD_LIBRARY_PATH}:"/util/gcc/lib64"
- These commands will update the PATH and LD_LIBRARY_PATH environment variables to use the /util versions of gcc. That is, add /util/gcc/bin to your PATH and /util/gcc/lib64 to your LD_LIBRARY_PATH

Commands:
The commands that we have run for getting the stats.txt file is shown below:

Benchmark 401.bzip2:

Varying L1 D Cache Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=32kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=64kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=256kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

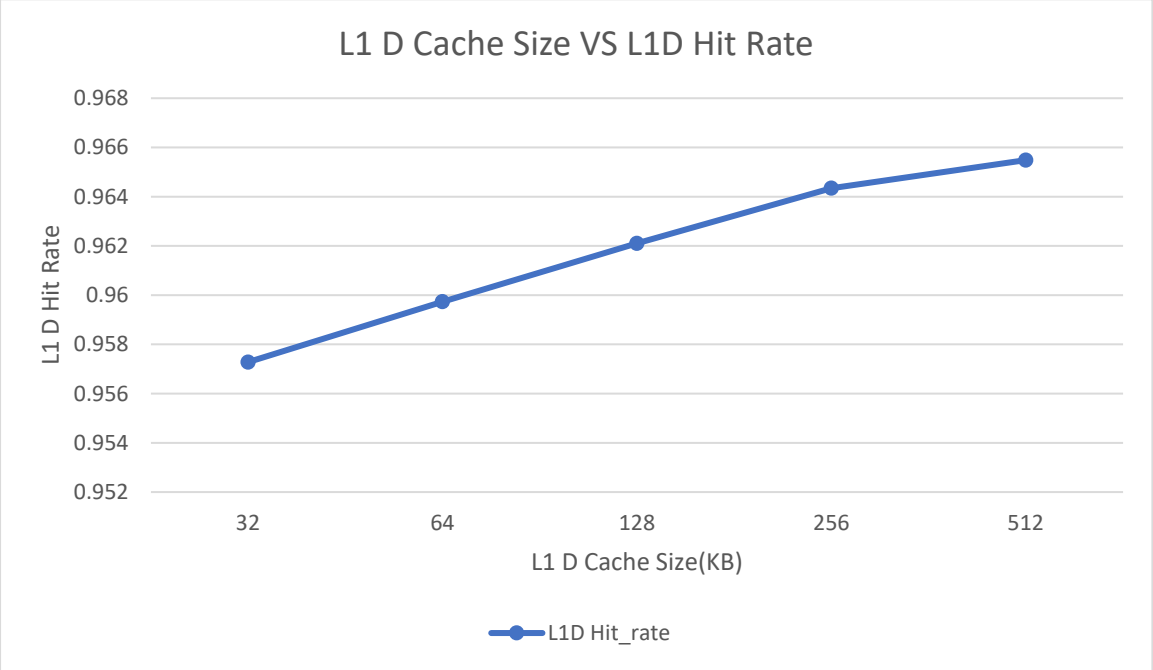
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I

100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=512kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

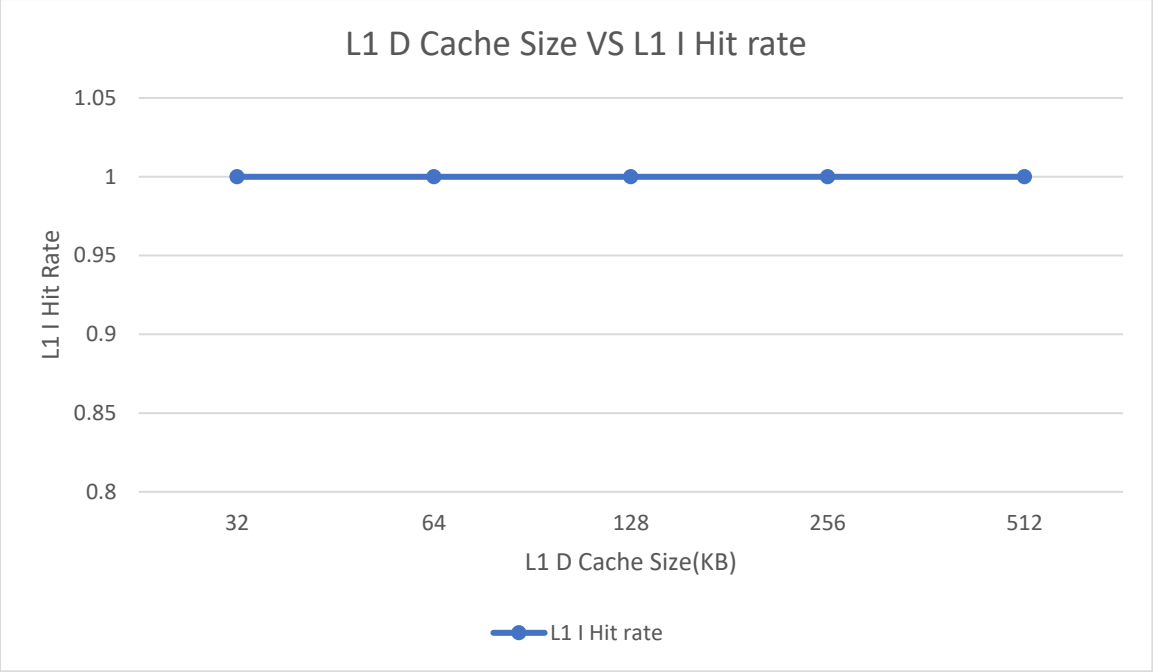
Analysis based on changes in L1D Cache size:

- Default values of Parameters: L1I Cache Size: 128KB, L2 Size: 1MB, L1D Associativity: 2, L1I Associativity: 2, L2 Associativity: 1, Block Size: 64B
- Results:

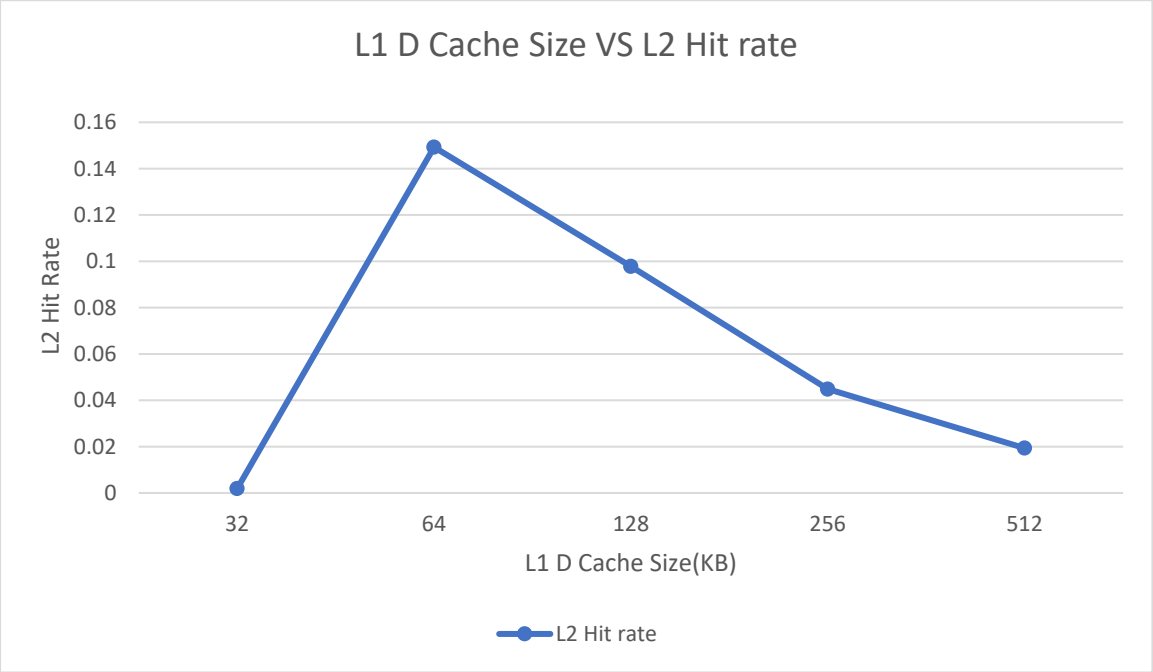
L1D Cache size (KB)	L1 D Hit rate	L1 I Hit rate	L2 Hit rate	CPI
32	0.957281	0.999996	0.001898	2.28182568
64	0.959733	0.999996	0.149268	2.27099548
128	0.962106	0.999996	0.097805	2.25951392
256	0.964344	0.999996	0.044844	2.24654222
512	0.965486	0.999996	0.019374	2.2352078

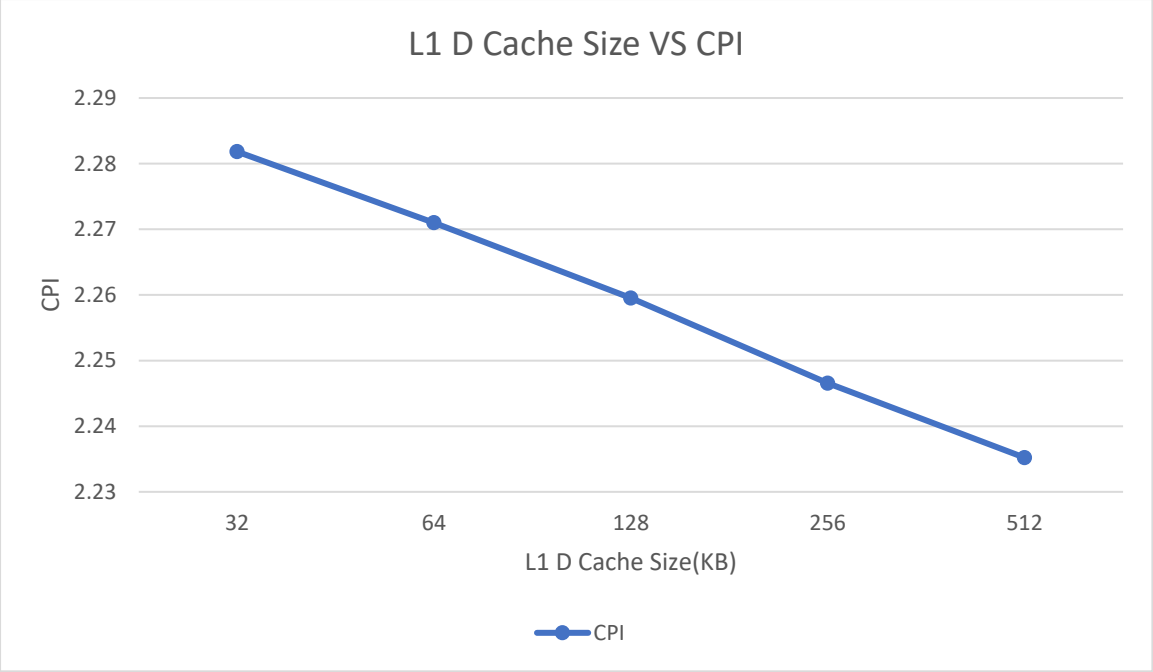


Here, as the size of L1D cache increase the Hit Rate also increases. This is because there is less chances of a conflict as the size of cache increases.



Here, L1I hit rate does not depend on change in L1D cache size.





As L1 D cache increases the hit rate of L1 D increases because there is less chance of conflict because size of the cache is larger. Now miss rate of L1 D cache is directly proportional to the CPI. So as seen in above graph CPI decreases with the increase in size of L1D.

Varying L1 I Cache Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=32kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=64kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --

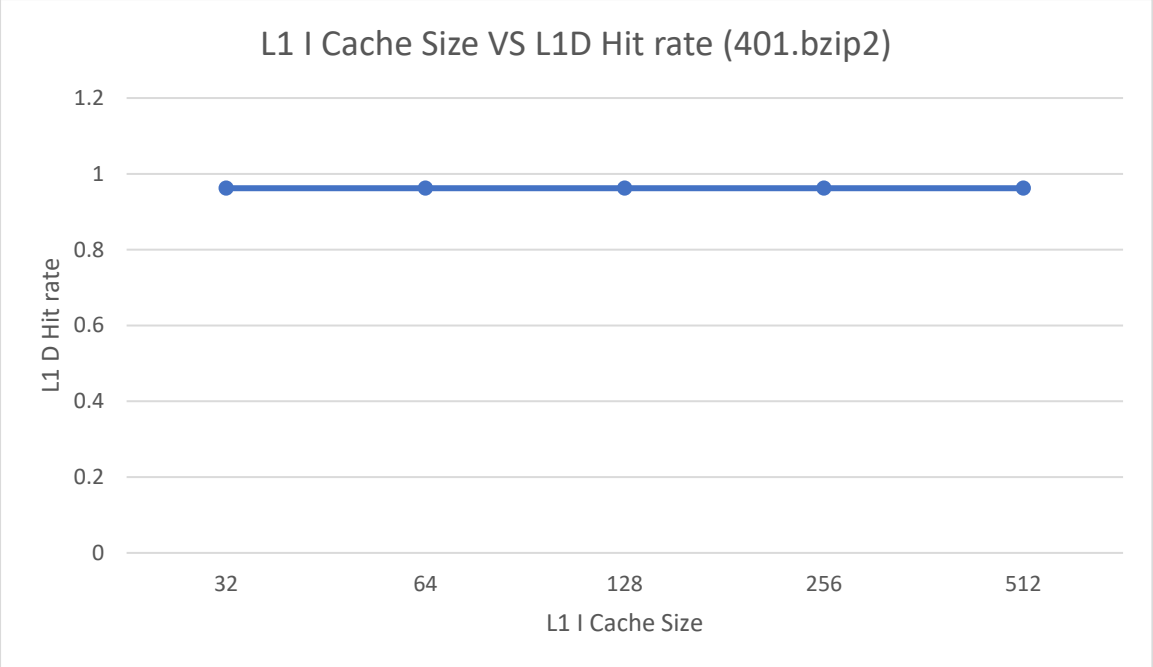
l1i_size=256kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

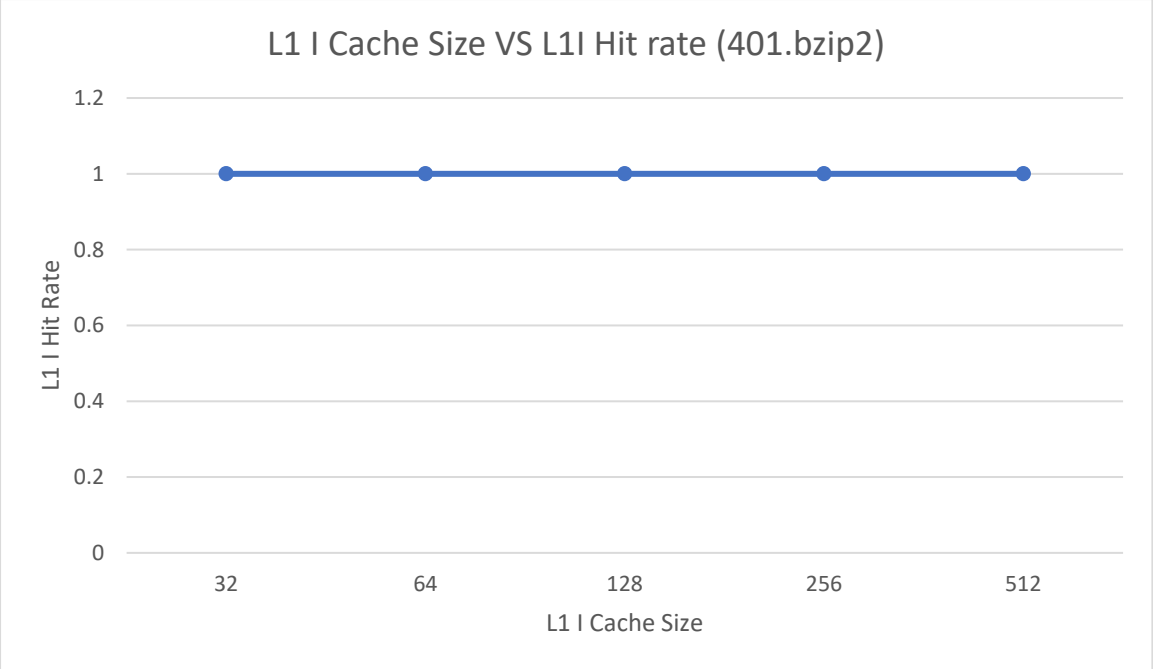
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=512kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

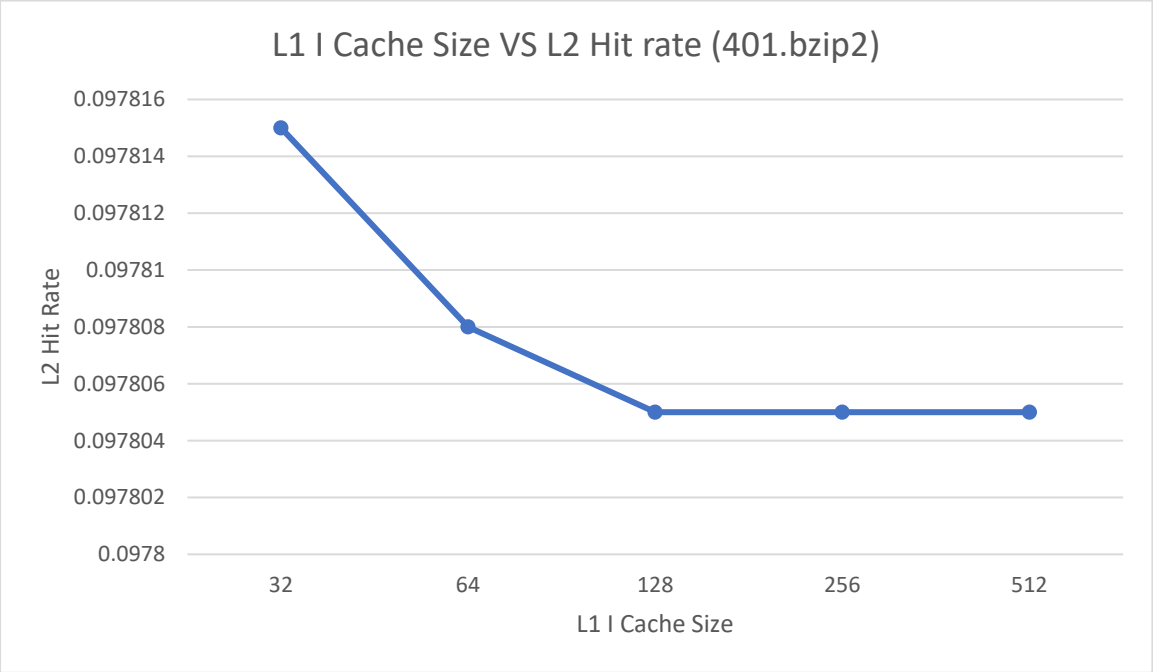
Analysis based on changes in L1I Cache size:

- Default values of Parameters: L1D Cache Size: 128KB, L2 Size: 1MB, L1D Associativity: 2, L1I Associativity: 2, L2 Associativity: 1, Block Size: 64B
- Results:

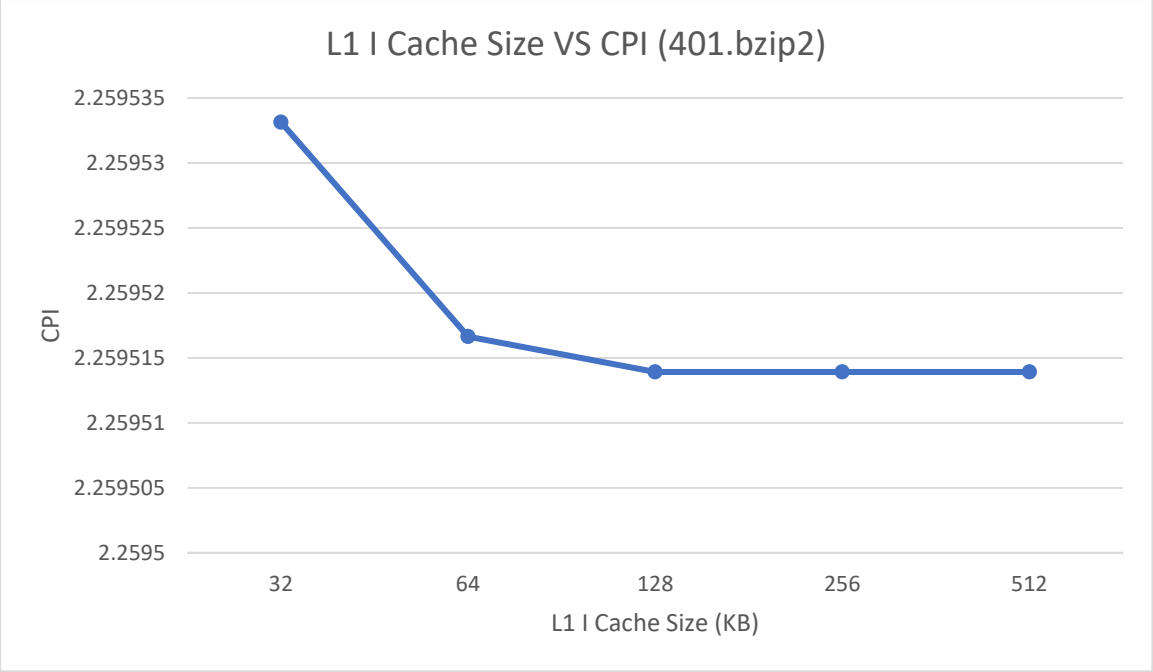
L1I size (KB)	L1D Hit rate	L1 I Hit rate	L2 Hit rate	CPI
32	0.962106	0.999996	0.097815	2.25953314
64	0.962106	0.999996	0.097808	2.25951664
128	0.962106	0.999996	0.097805	2.25951392
256	0.962106	0.999996	0.097805	2.25951392
512	0.962106	0.999996	0.097805	2.25951392







So, it can be clearly seen that as the L1I cache size increases, the L2 cache hit rate decreases.



As the L1I cache size increases the CPI decreases. This is because L1I cache size is directly proportional to hit rate which is inversely proportional to CPI.

Varying L2 Cache Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=256kB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=512kB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --

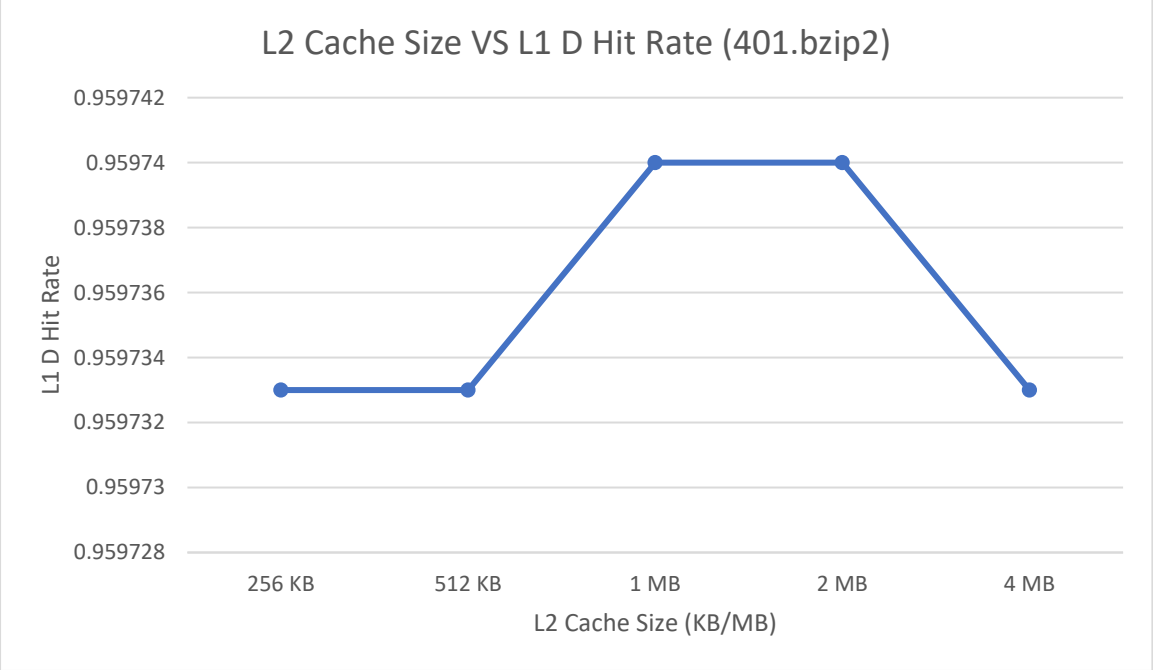
l1i_size=128kB --l2_size=2MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

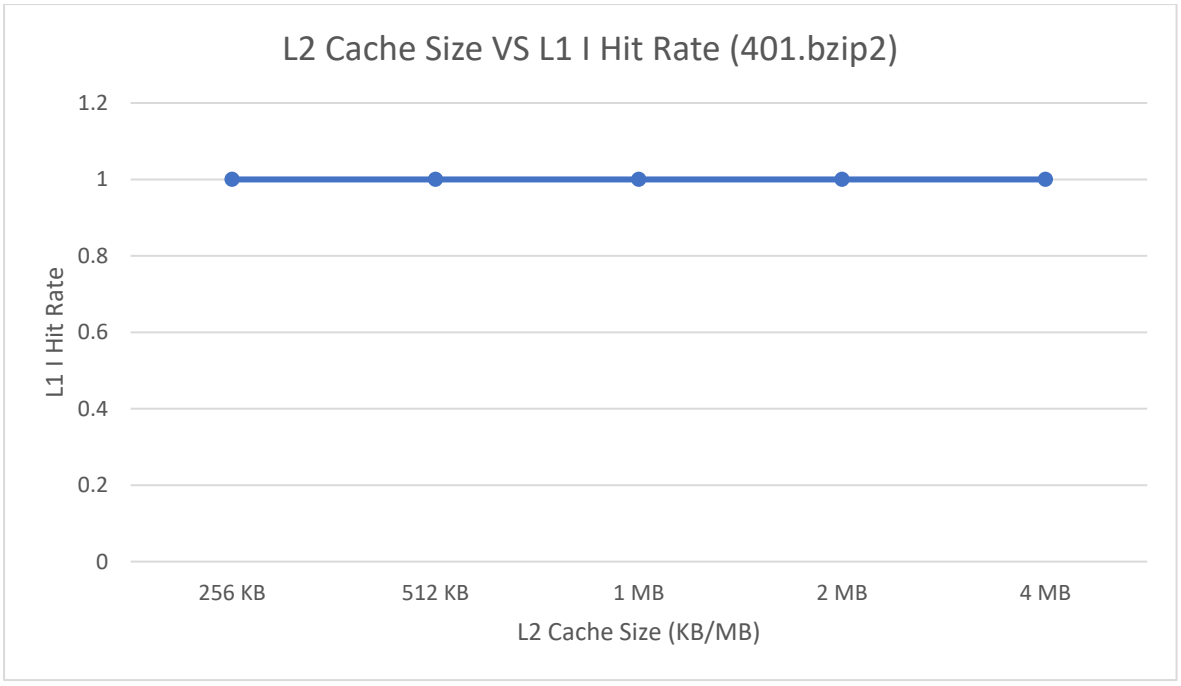
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=4MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

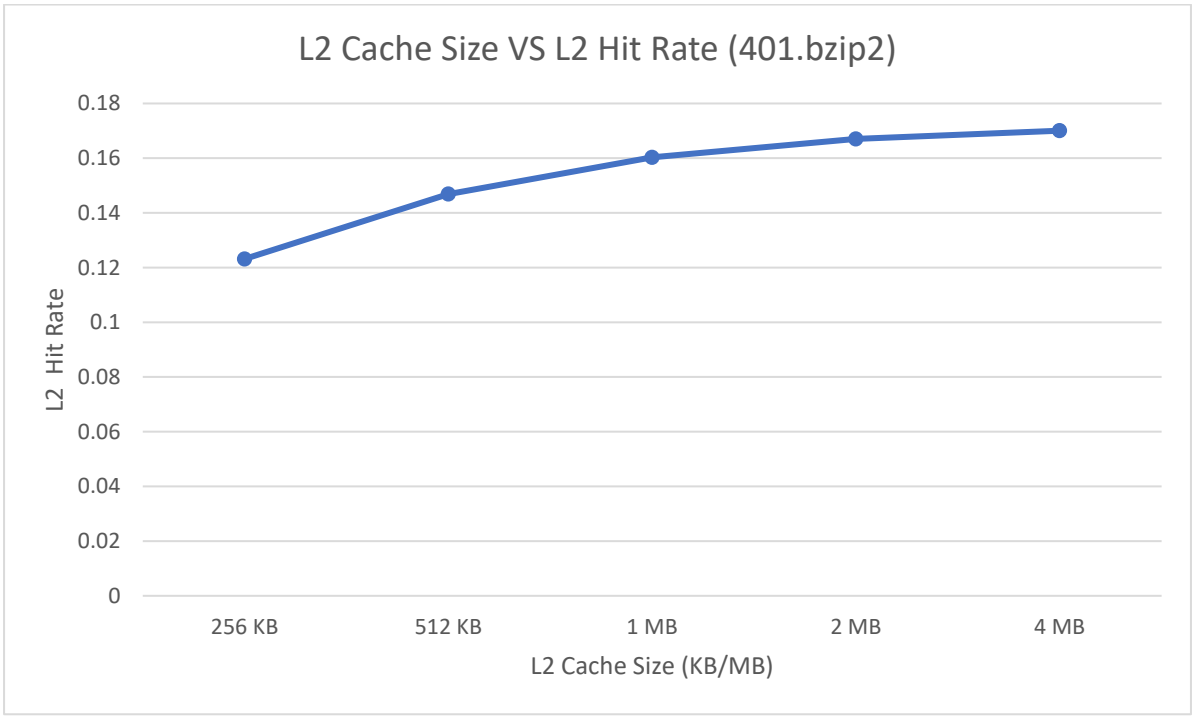
Analysis based on changes in L2 Cache size:

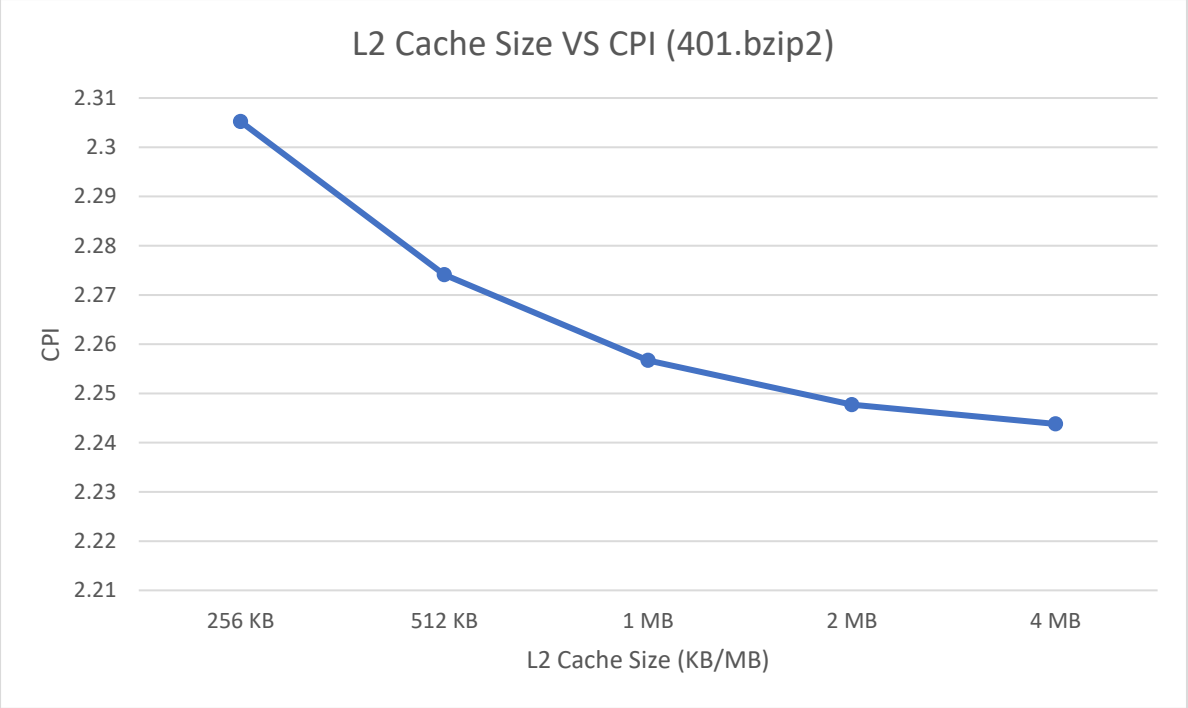
- Default values of Parameters: L1D Cache Size: 128KB, L1I Size: 128KB, L1D Associativity: 2, L1I Associativity: 2, L2 Associativity: 1, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L2 size (KB/MB)	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
256 KB	0.959733	0.999996	0.123156	2.3052
512 KB	0.959733	0.999996	0.146864	2.2741
1 MB	0.95974	0.999996	0.1603	2.2567
2 MB	0.95974	0.999996	0.167026	2.2477
4 MB	0.959733	0.999996	0.170045	2.2438




L2 Cache Size VS L1 I Hit Rate (401.bzip2)
1.2
1
0.8
0.6
0.4
0.2
0
L1 I Hit Rate
256 KB
512 KB
1 MB
2 MB
4 MB
L2 Cache Size (KB/MB)


L2 Cache Size VS L2 Hit Rate (401.bzip2)
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
L2 Hit Rate
256 KB
512 KB
1 MB
2 MB
4 MB
L2 Cache Size (KB/MB)



Varying L1D Associativity:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=1 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=4 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

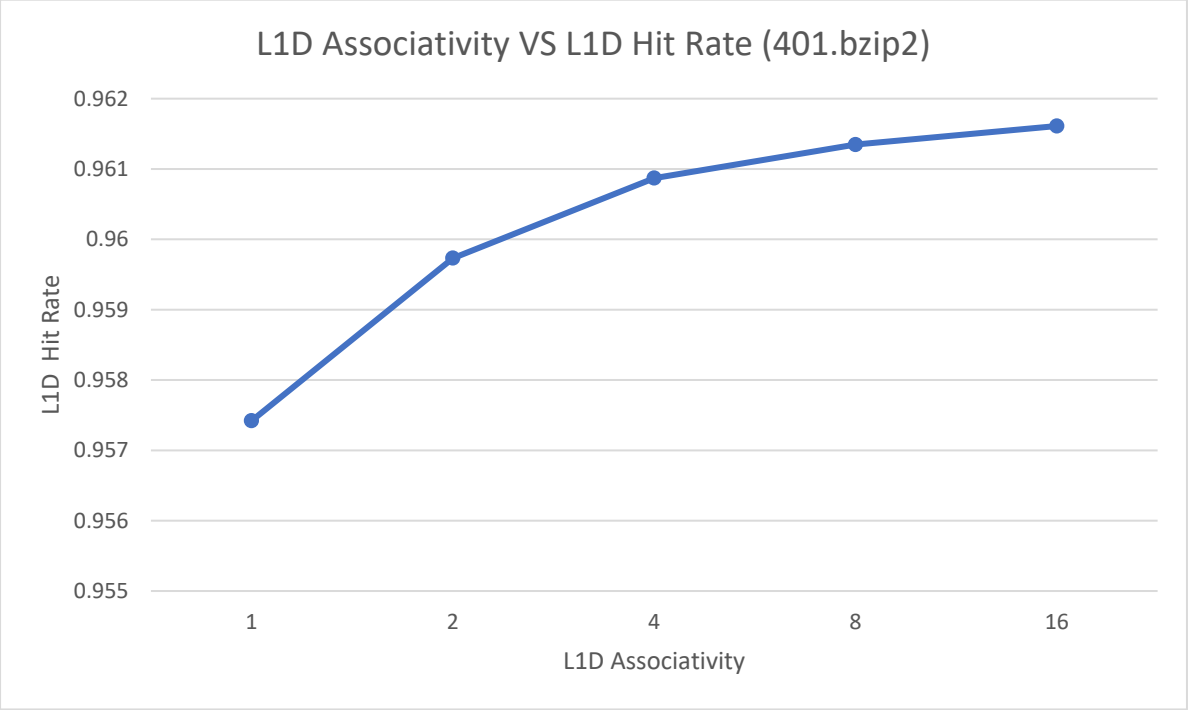
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=8 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

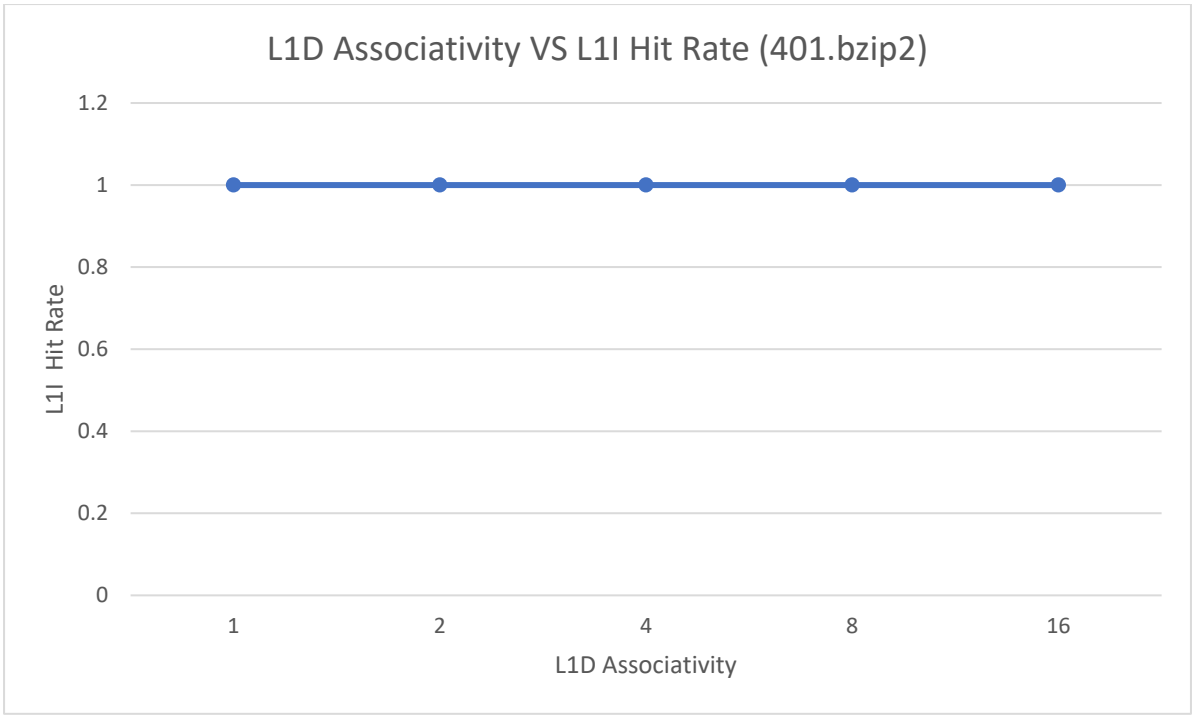
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=16 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

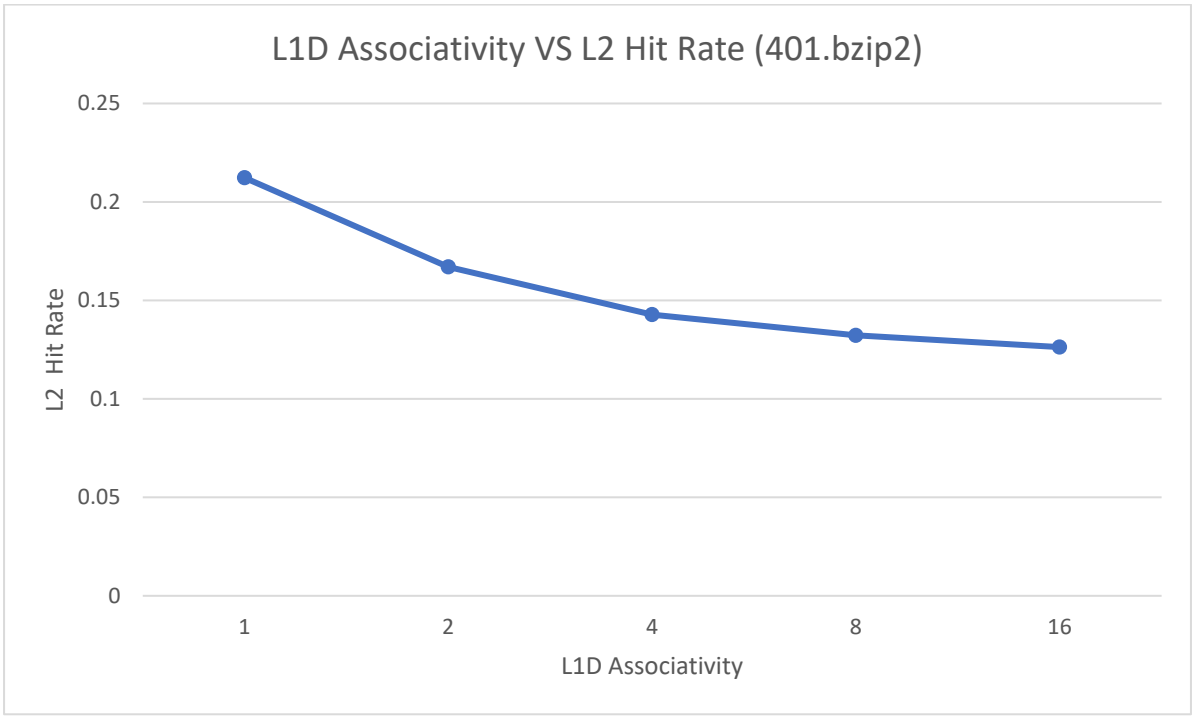
Analysis based on changes in L1D Associativity:

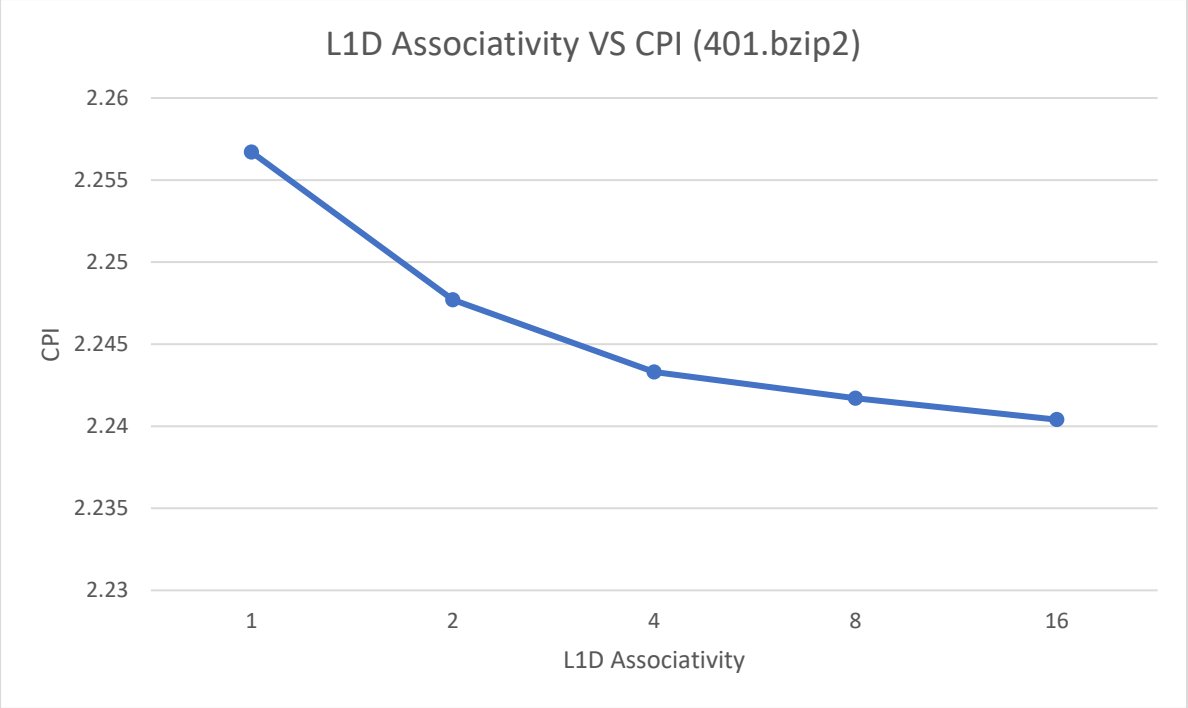
- Default values of Parameters: L1D Cache Size: 128KB, L1I Size: 128KB, L2 Cache Size:1MB, L1I Associativity: 2, L2 Associativity: 1, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1D Associativity	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
1	0.95742	0.999996	0.212263	2.2567
2	0.959733	0.999996	0.167026	2.2477
4	0.960871	0.999996	0.142798	2.2433
8	0.961347	0.999996	0.132259	2.2417
16	0.961611	0.999996	0.126296	2.2404




L1D Associativity VS L1I Hit Rate (401.bzip2)
1.2
1
0.8
0.6
0.4
0.2
0
L1I Hit Rate
1
2
4
8
16
L1D Associativity


L1D Associativity VS L2 Hit Rate (401.bzip2)
0.25
0.2
0.15
0.1
0.05
0
L2 Hit Rate
1
2
4
8
16
L1D Associativity



Varying L1I Associativity:

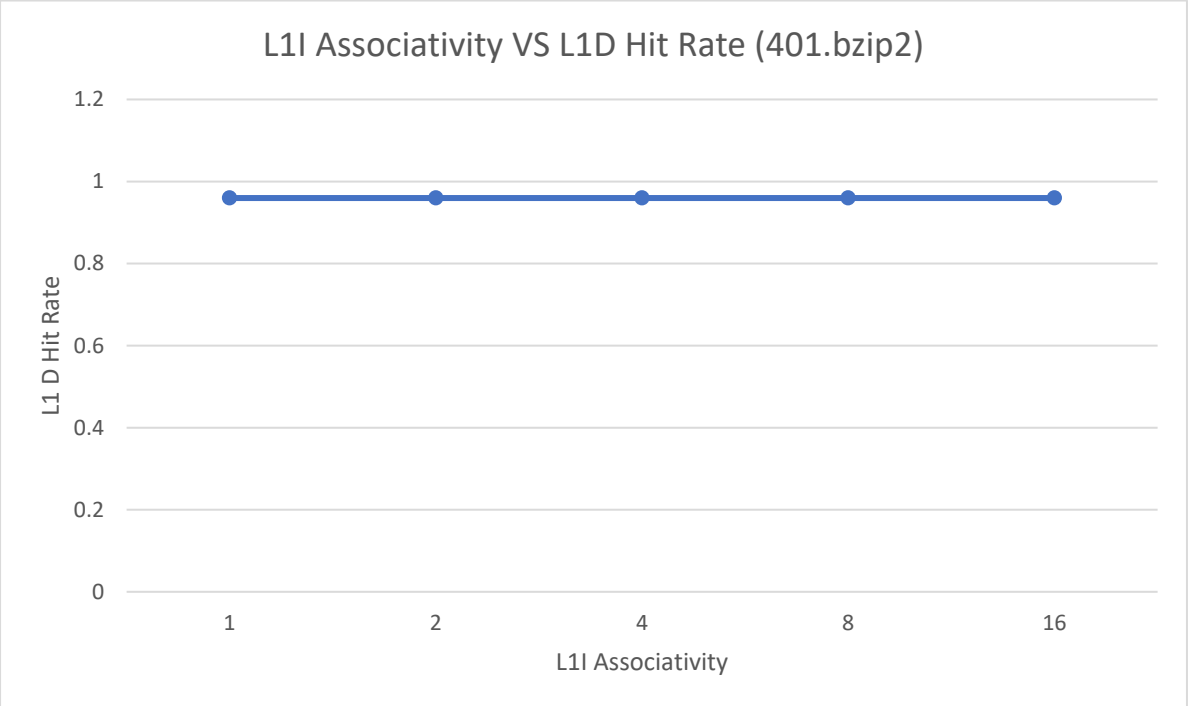
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=1 --l2_assoc=1 --cacheline_size=64
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=4 --l2_assoc=1 --cacheline_size=64
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=8 --l2_assoc=1 --cacheline_size=64
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --

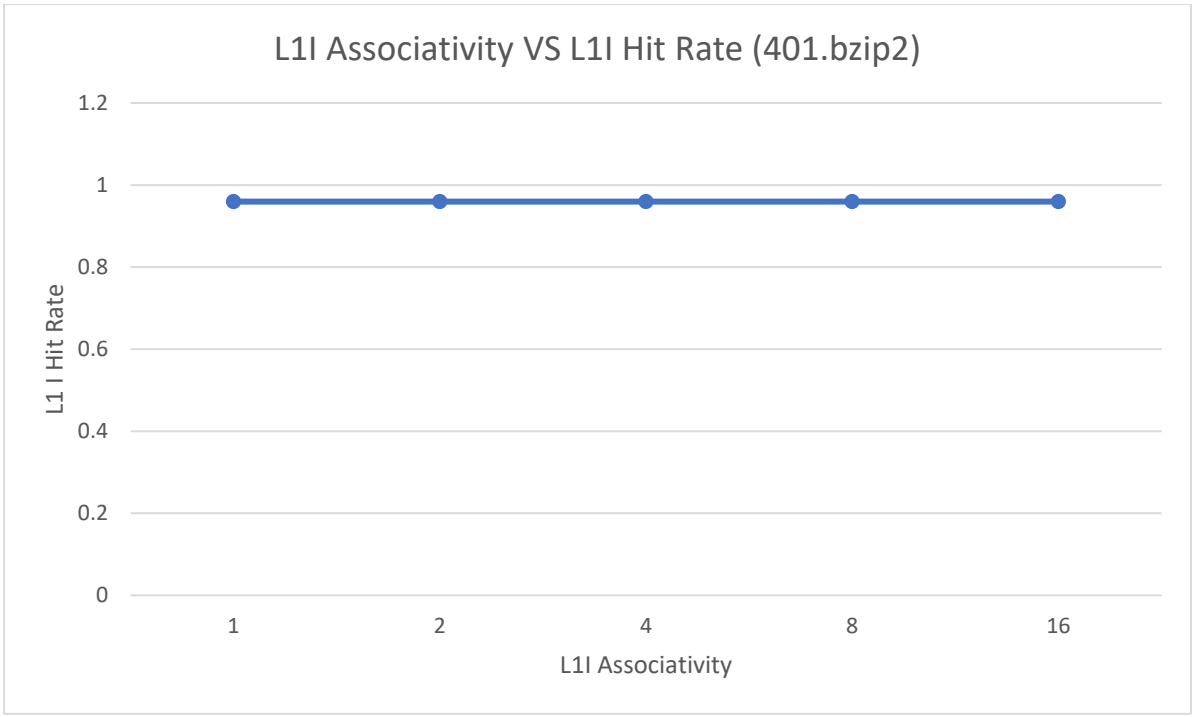
l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=16 --l2_assoc=1 --cacheline_size=64

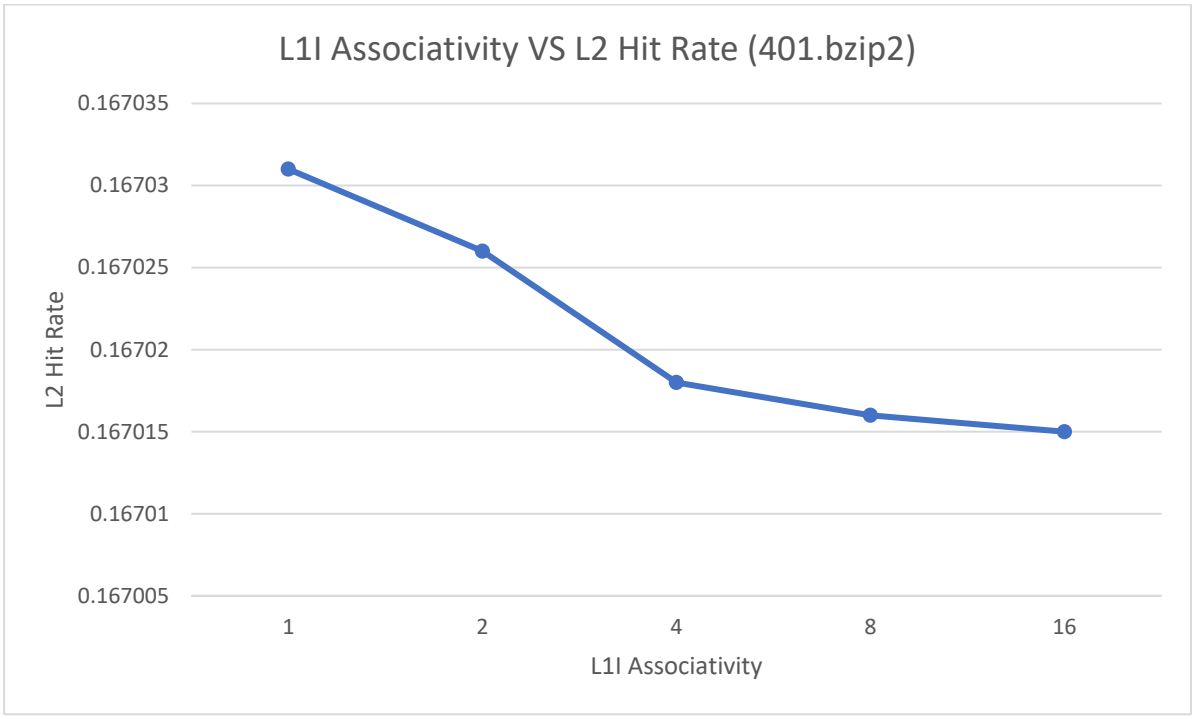
Analysis based on changes in L1I Associativity:

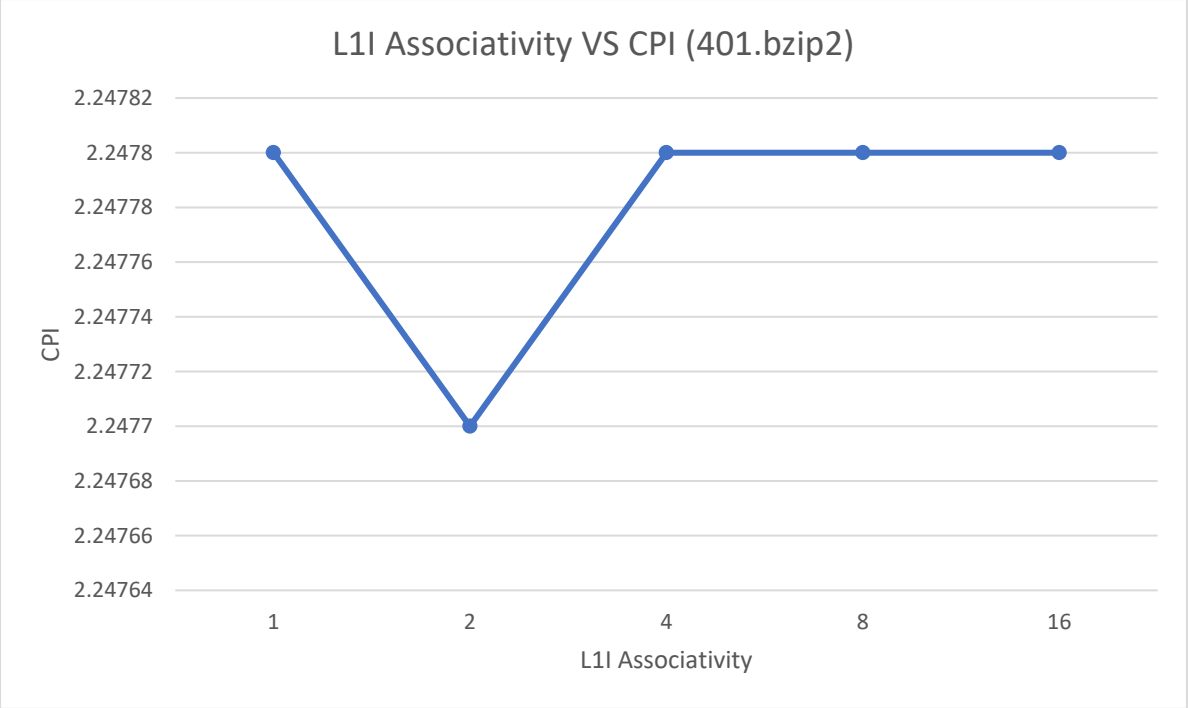
- Default values of Parameters: L1D Cache Size: 128KB, L1I Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L2 Associativity: 1, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1I Associativity	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
1	0.959733	0.999995	0.167031	2.2478
2	0.959733	0.999996	0.167026	2.2477
4	0.959733	0.999996	0.167018	2.2478
8	0.959733	0.999996	0.167016	2.2478
16	0.959733	0.999996	0.167015	2.2478




L1I Associativity VS L1I Hit Rate (401.bzip2)
1.2
1
0.8
0.6
0.4
0.2
0
L1 I Hit Rate
1
2
4
8
16
L1I Associativity


L1I Associativity VS L2 Hit Rate (401.bzip2)
0.167035
0.16703
0.167025
0.16702
0.167015
0.16701
0.167005
L2 Hit Rate
1
2
4
8
16
L1I Associativity



Varying L2 Associativity:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=2 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=4 --cacheline_size=64

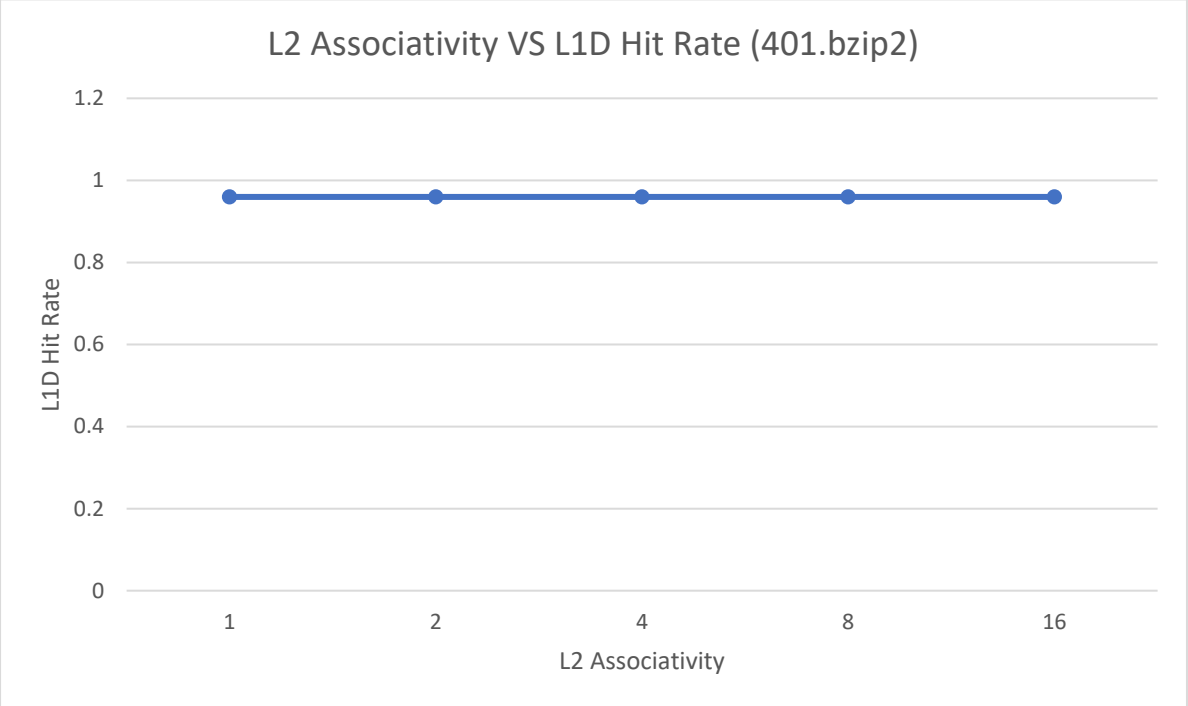
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=8 --cacheline_size=64

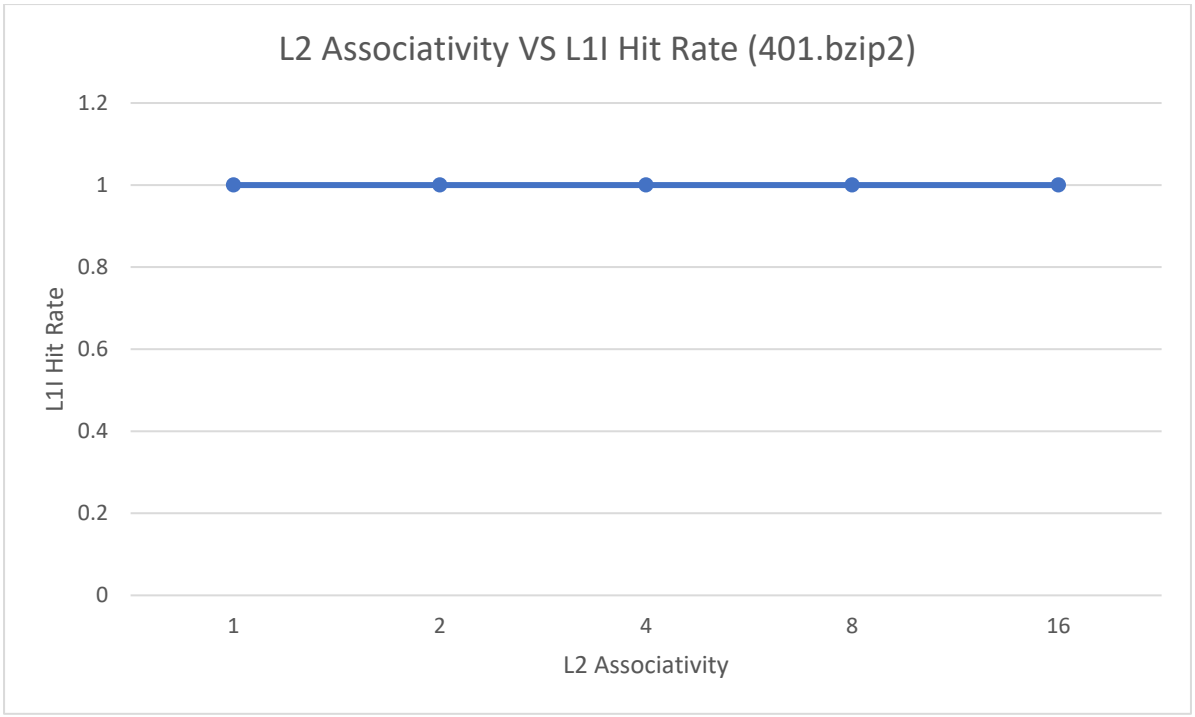
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=16 --cacheline_size=64

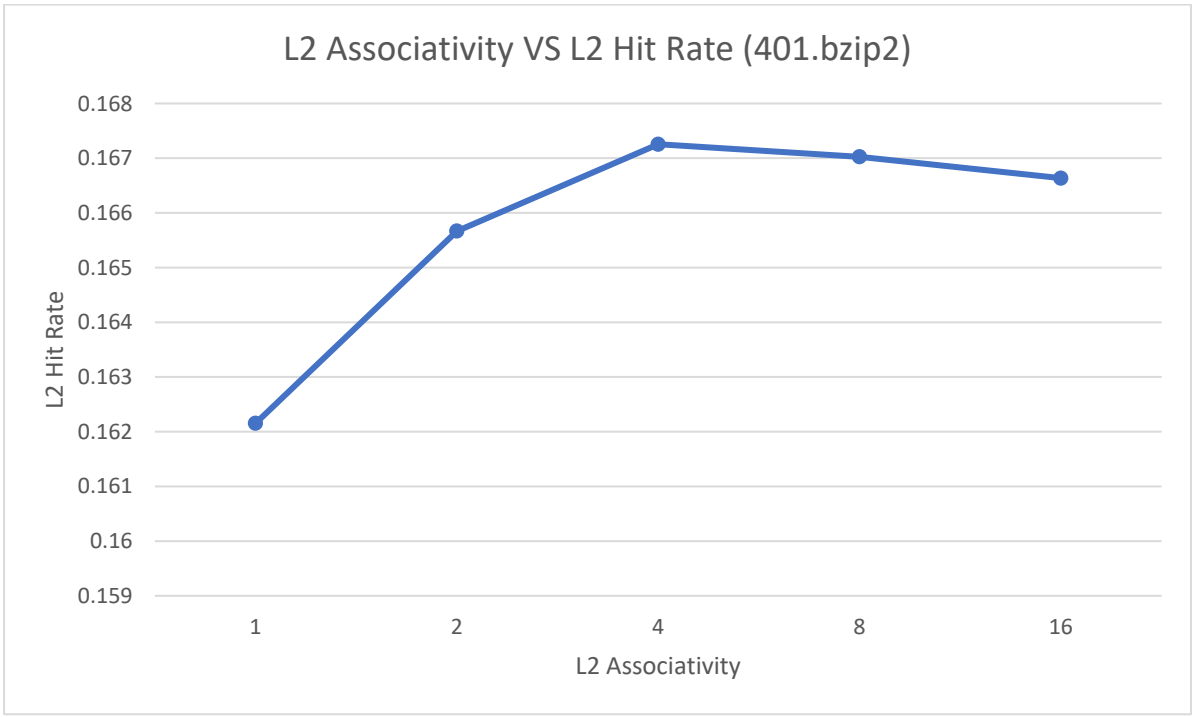
Analysis based on changes in L2 Associativity:

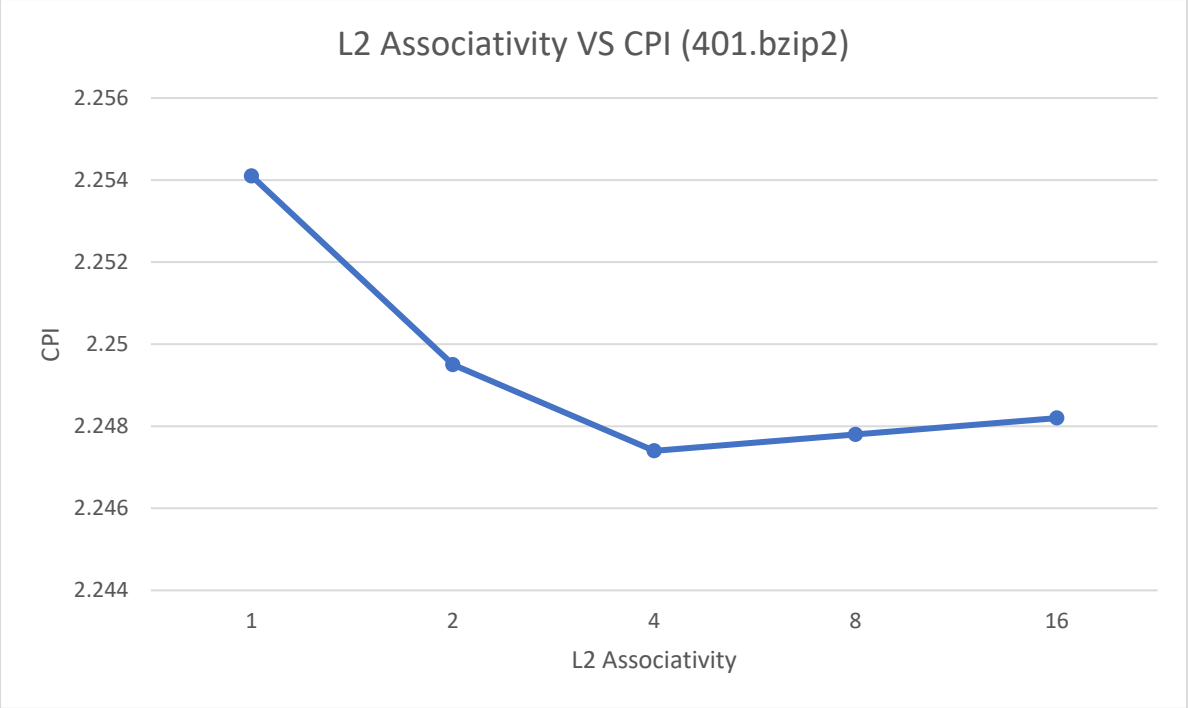
- Default values of Parameters: L1D Cache Size: 128KB, L1I Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L1I Associativity: 2, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L2 Associativity	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
1	0.959733	0.999996	0.162154	2.2541
2	0.959733	0.999996	0.165668	2.2495
4	0.959733	0.999996	0.167255	2.2474
8	0.959733	0.999996	0.167026	2.2478
16	0.959733	0.999996	0.166635	2.2482




L2 Associativity VS L1I Hit Rate (401.bzip2)
1.2
1
0.8
0.6
0.4
0.2
0
L1I Hit Rate
1
2
4
8
16
L2 Associativity


L2 Associativity VS L2 Hit Rate (401.bzip2)
0.168
0.167
0.166
0.165
0.164
0.163
0.162
0.161
0.16
0.159
L2 Hit Rate
1
2
4
8
16
L2 Associativity



Varying Block Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=8

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=16

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=32

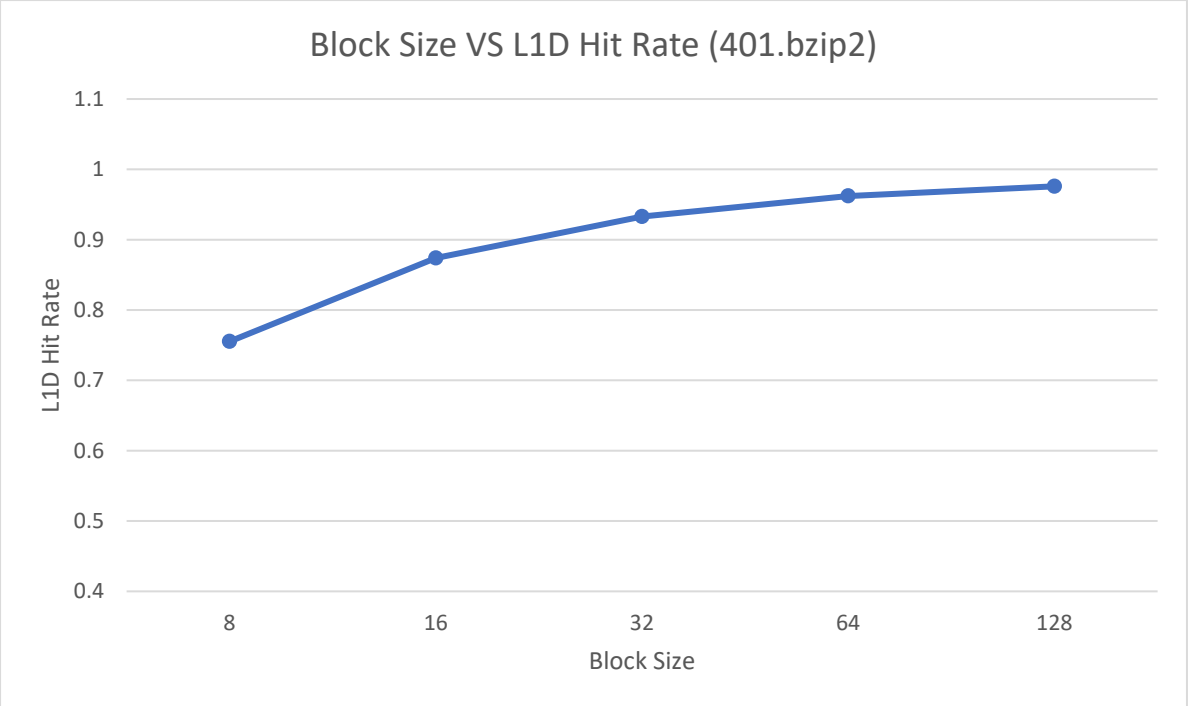
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

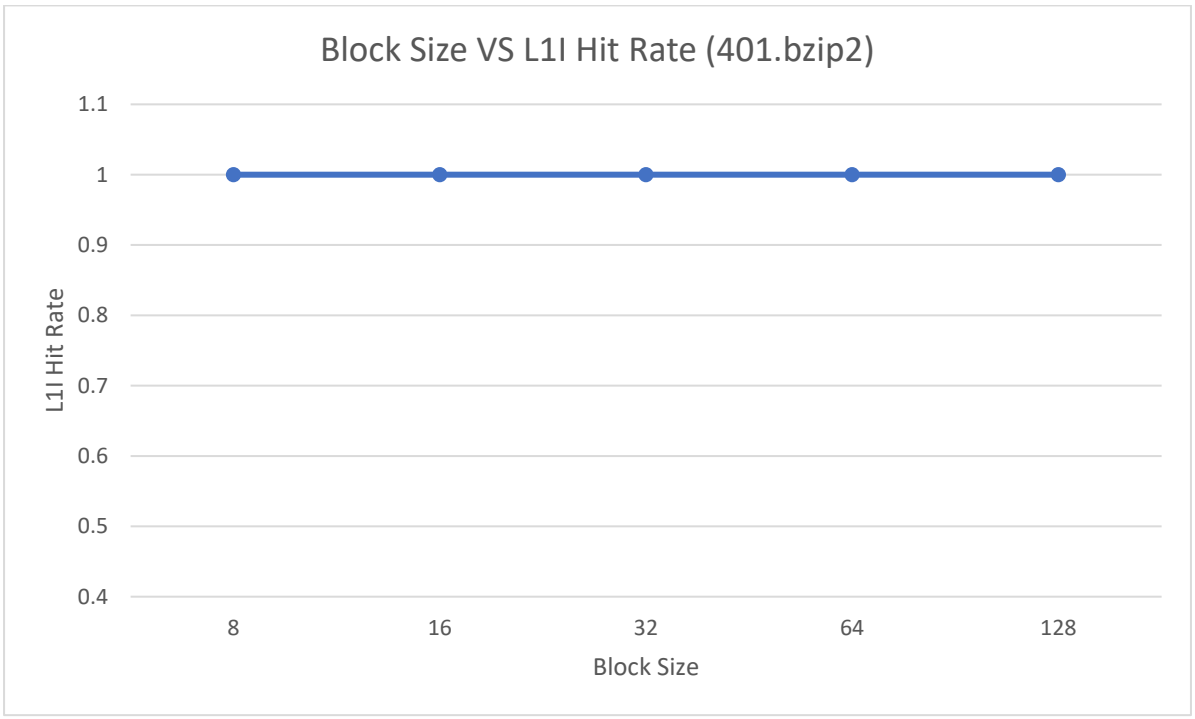
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=128

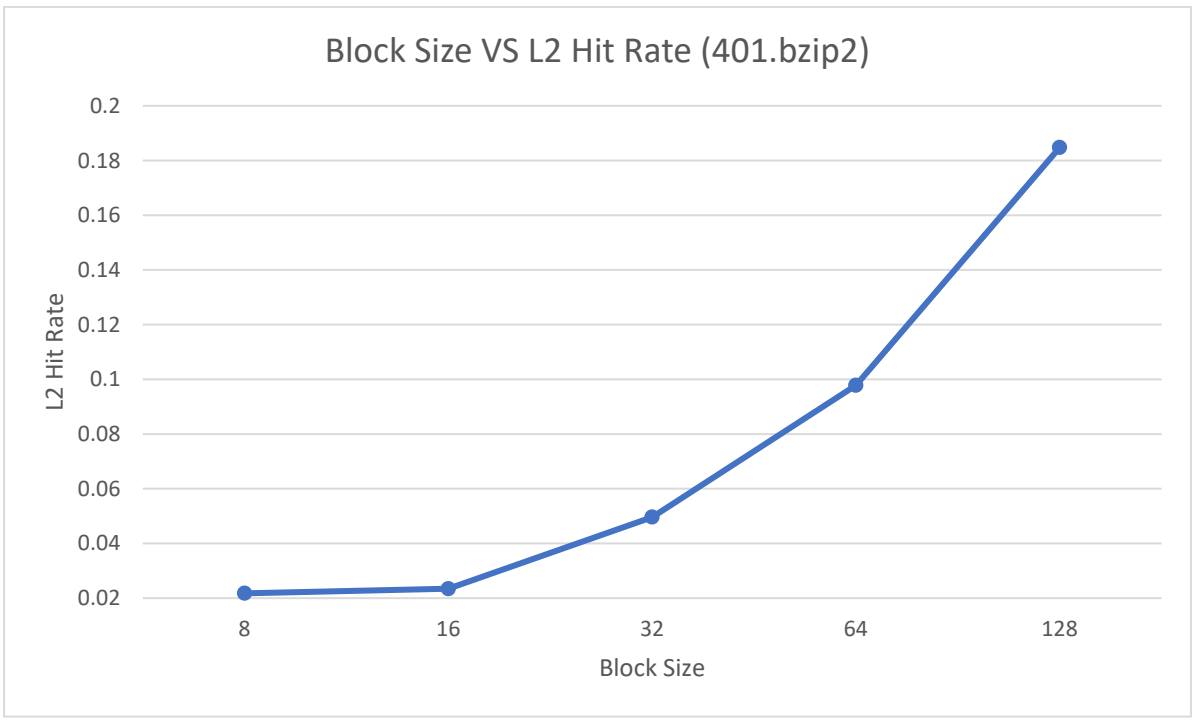
Analysis based on changes in Block Size:

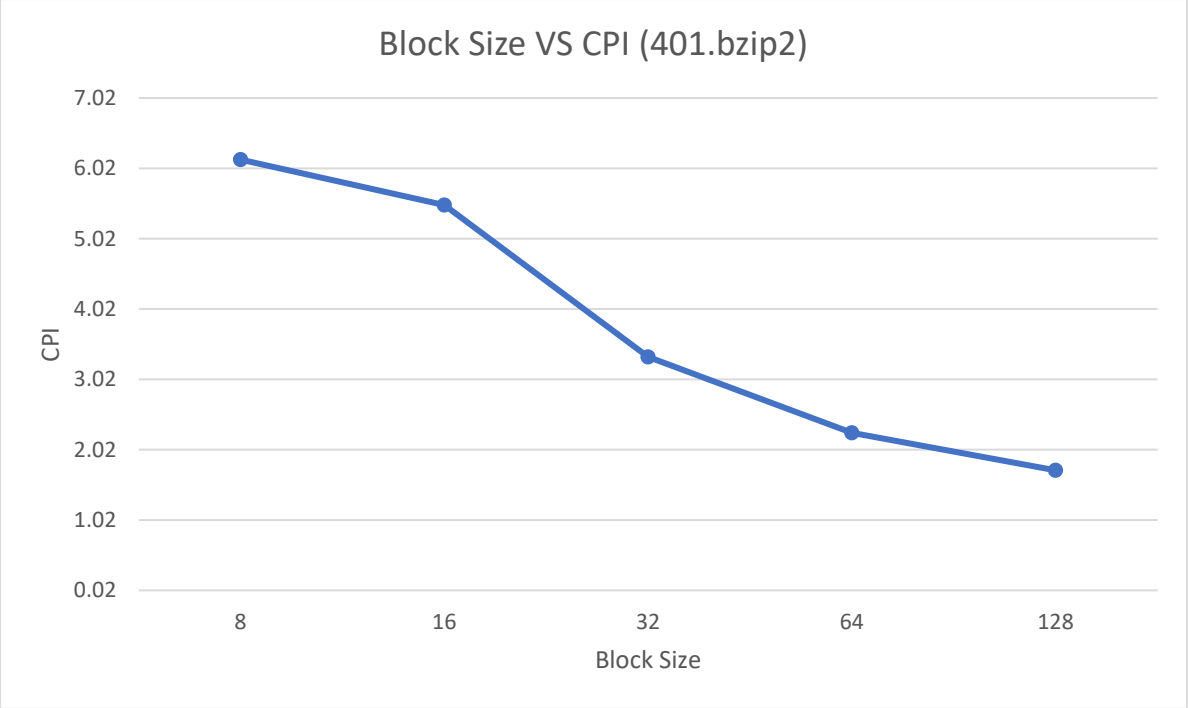
- Default values of Parameters: L1D Cache Size: 128KB, L1I Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L1I Associativity: 2, L2 Associativity:1.
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

Block size	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
8	0.7553	0.999979	0.021759	6.14379778
16	0.873866	0.999989	0.023449	5.49729918
32	0.93283	0.999993	0.049628	3.33778874
64	0.962106	0.999996	0.097805	2.25951392
128	0.975847	0.999998	0.184772	1.72650308




Block Size VS L1I Hit Rate (401.bzip2)
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
L1I Hit Rate
8
16
32
64
128
Block Size


Block Size VS L2 Hit Rate (401.bzip2)
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
L2 Hit Rate
8
16
32
64
128
Block Size



Benchmark 456.hmmer:

Varying L1D Cache Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAPp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=32kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=64kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=256kB --

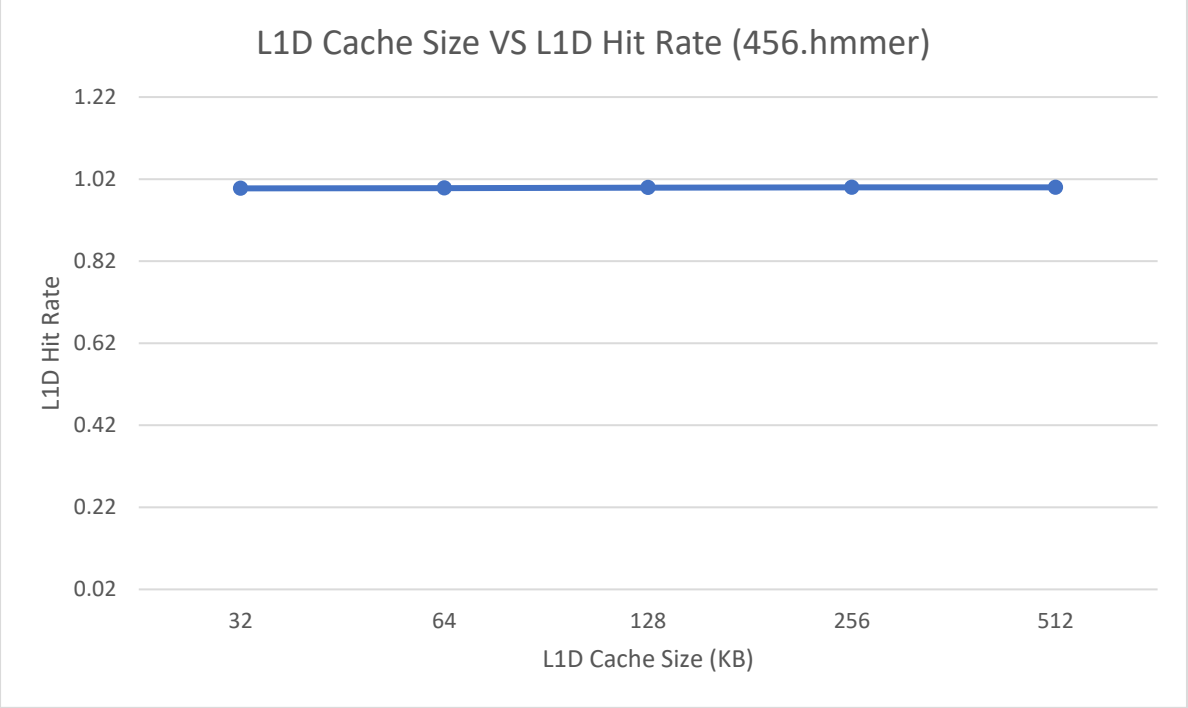
l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

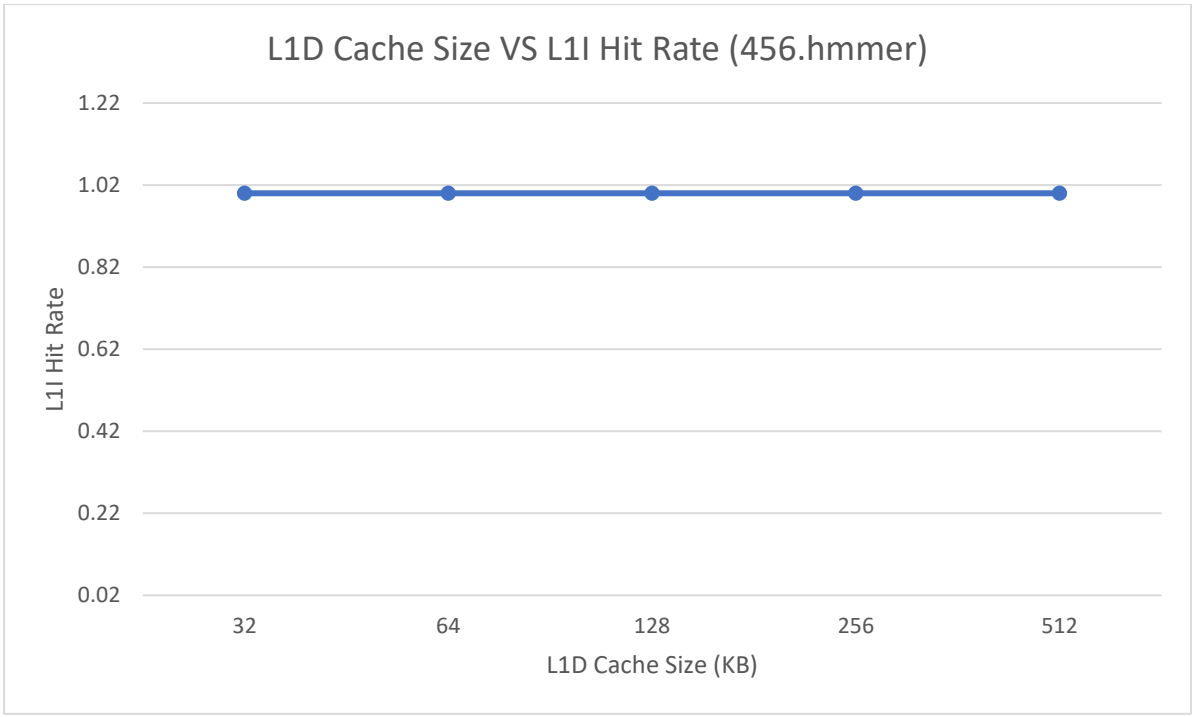
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=512kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

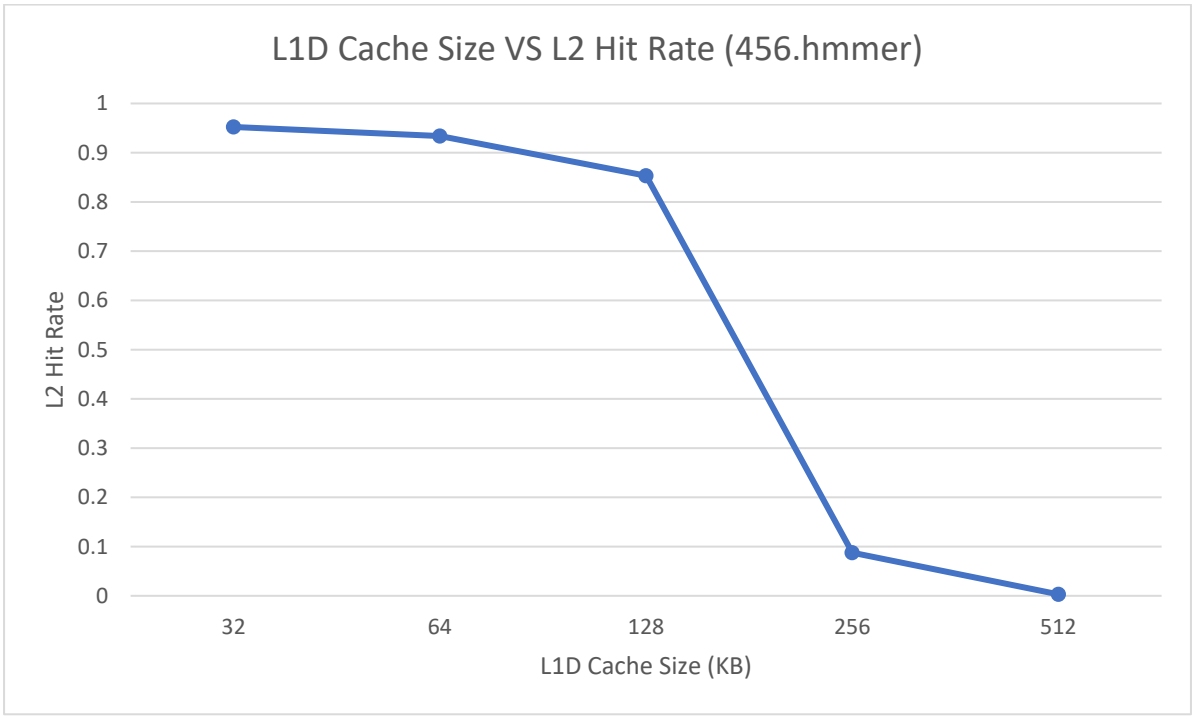
Analysis based on changes in L1D Cache Size:

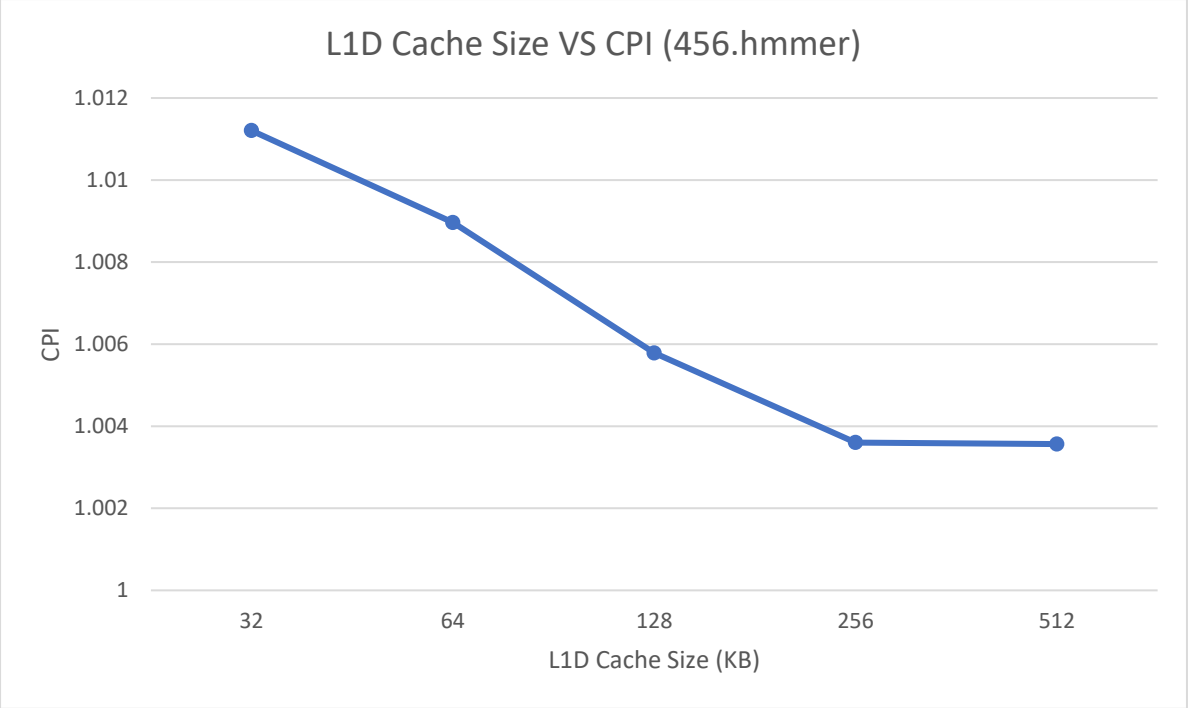
- Default values of Parameters: L1I Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L1I Associativity: 2, L2 Associativity:1, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1D size (KB)	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
32	0.997408	0.99999	0.95227	1.01120492
64	0.998138	0.99999	0.933848	1.00896538
128	0.999177	0.99999	0.853143	1.0057848
256	0.99989	0.99999	0.087573	1.00360162
512	0.999901	0.99999	0.002976	1.00356554




L1D Cache Size VS L1I Hit Rate (456.hmmer)
1.22
1.02
0.82
0.62
0.42
0.22
0.02
L1I Hit Rate
32
64
128
256
512
L1D Cache Size (KB)


L1D Cache Size VS L2 Hit Rate (456.hmmer)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
L2 Hit Rate
32
64
128
256
512
L1D Cache Size (KB)



Varying L1I Cache Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=32kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=64kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=256kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

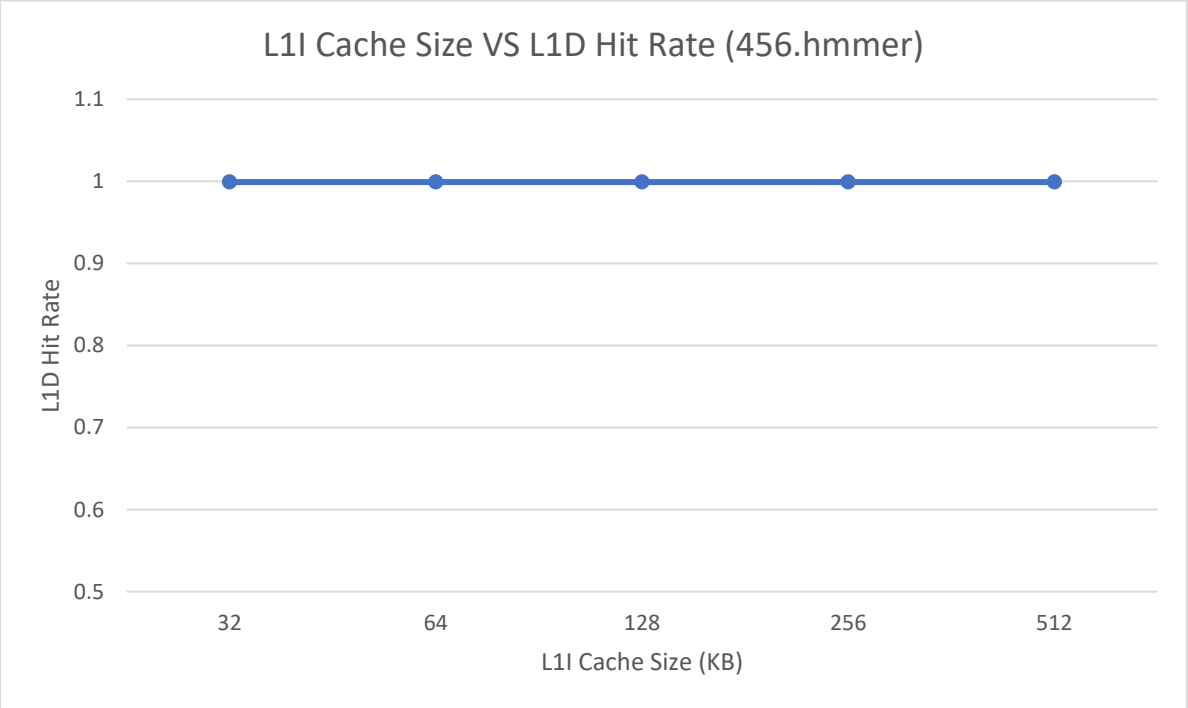
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new

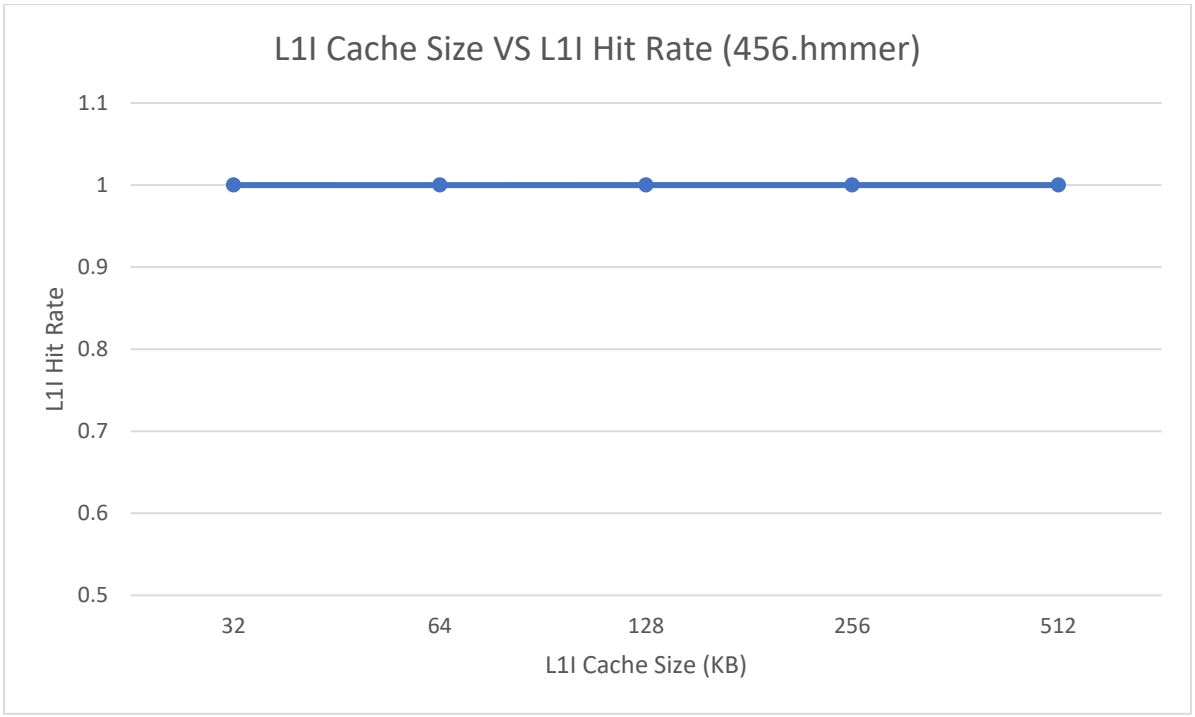
-I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=512kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

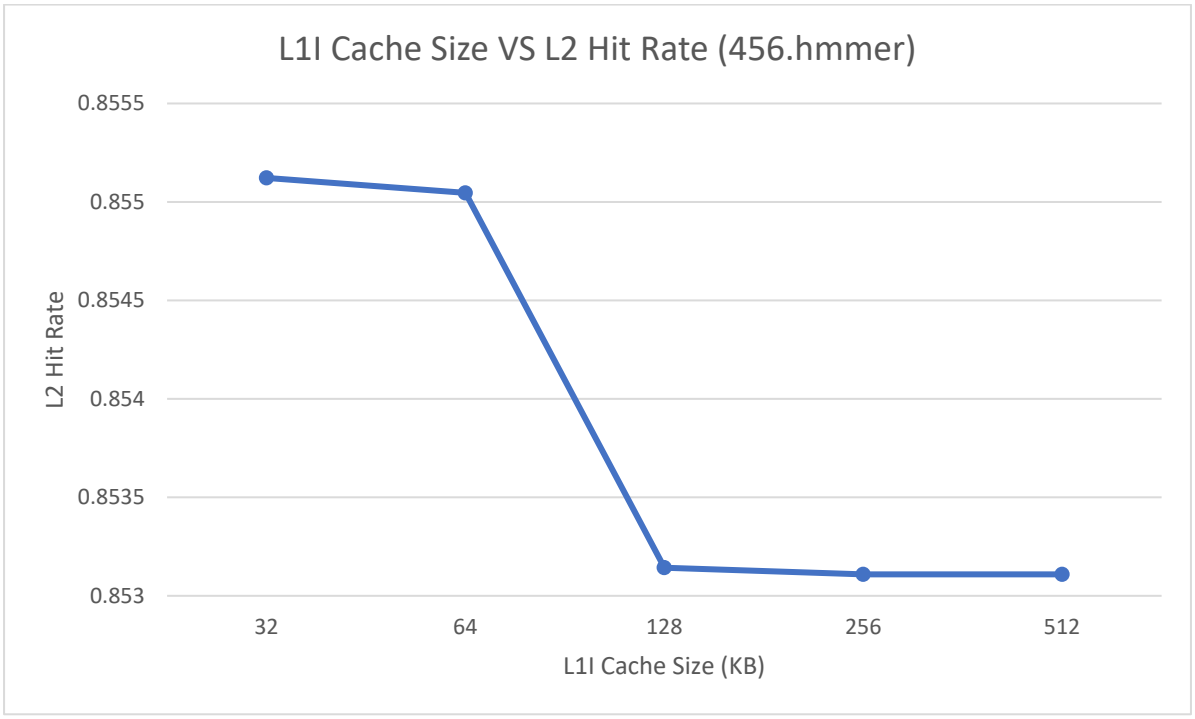
Analysis based on changes in L1I Cache Size:

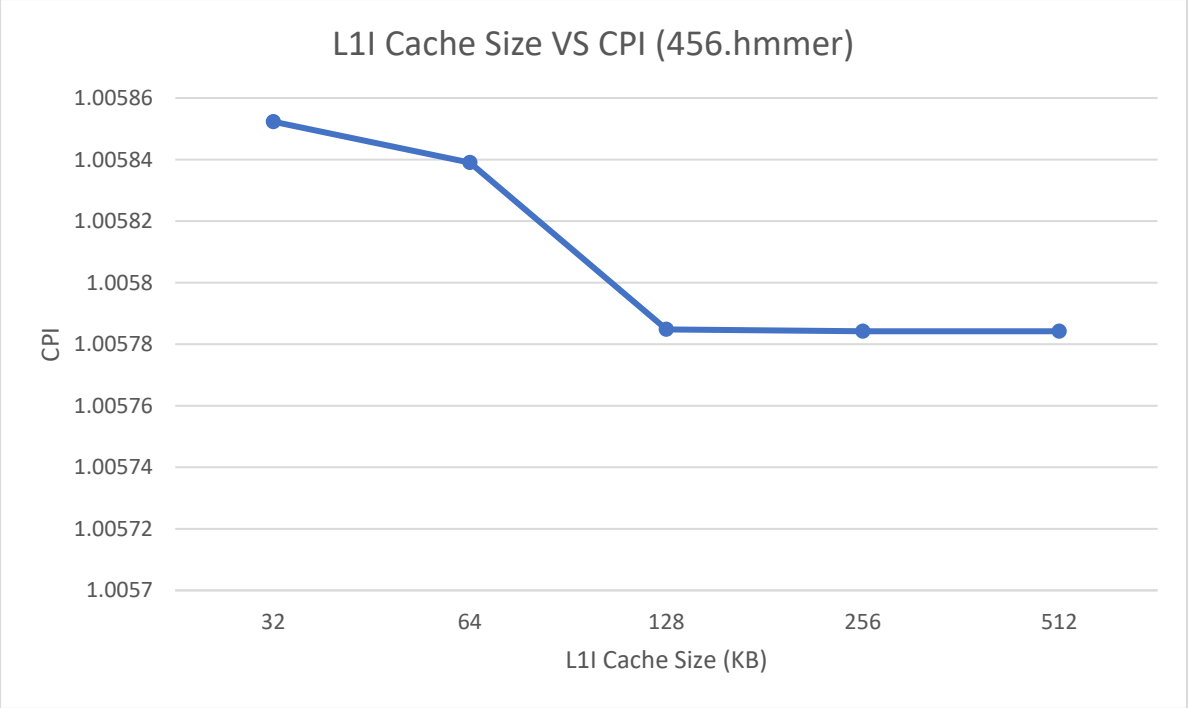
- Default values of Parameters: L1D Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L1I Associativity: 2, L2 Associativity:1, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1I size (KB)	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
32	0.999177	0.999984	0.855122	1.0058523
64	0.999177	0.999985	0.855046	1.0058390
128	0.999177	0.99999	0.853143	1.0057848
256	0.999177	0.99999	0.853109	1.0057842
512	0.999177	0.99999	0.853109	1.0057842




L1I Cache Size VS L1I Hit Rate (456.hmmer)
1.1
1
0.9
0.8
0.7
0.6
0.5
L1I Hit Rate
32
64
128
256
512
L1I Cache Size (KB)


L1I Cache Size VS L2 Hit Rate (456.hmmer)
0.8555
0.855
0.8545
0.854
0.8535
0.853
L2 Hit Rate
32
64
128
256
512
L1I Cache Size (KB)



Varying L2 Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=256kB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=512kB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

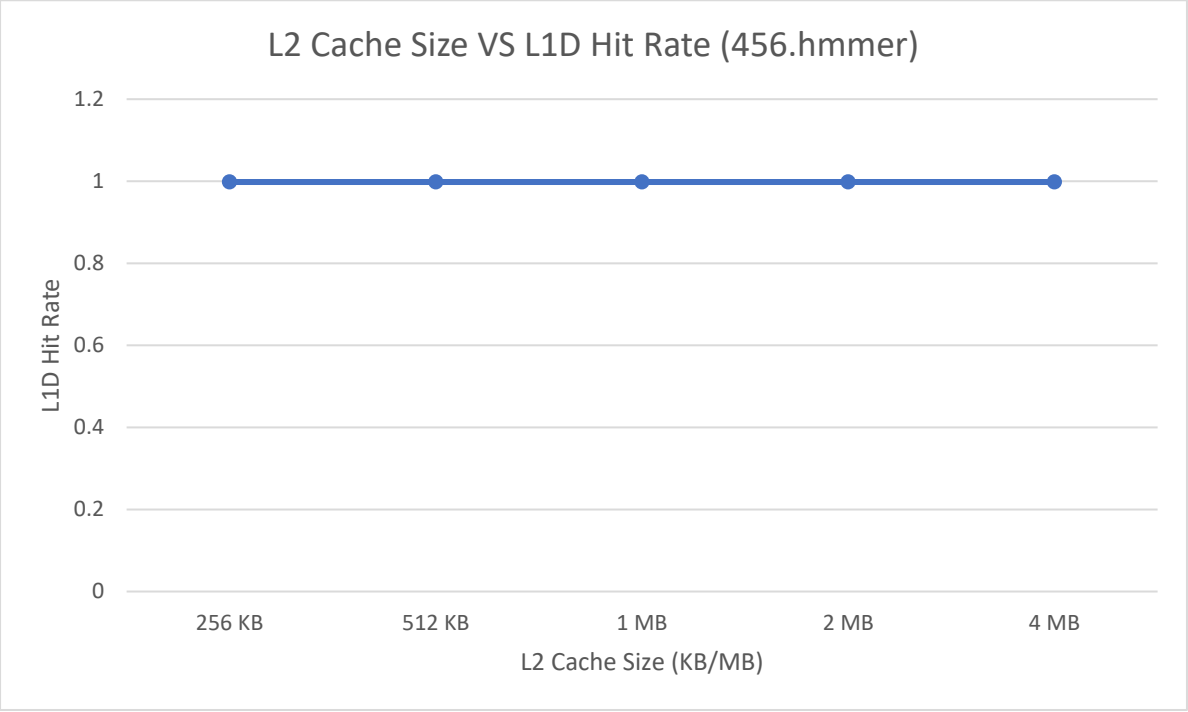
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=2MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

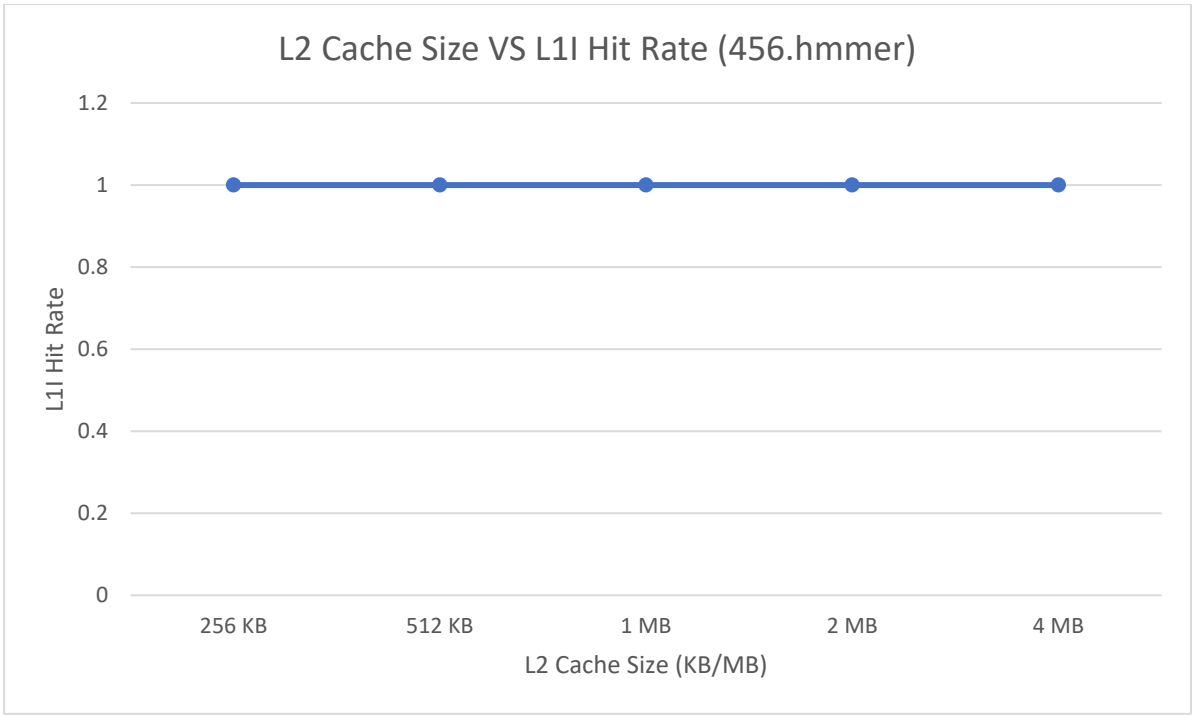
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=4MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

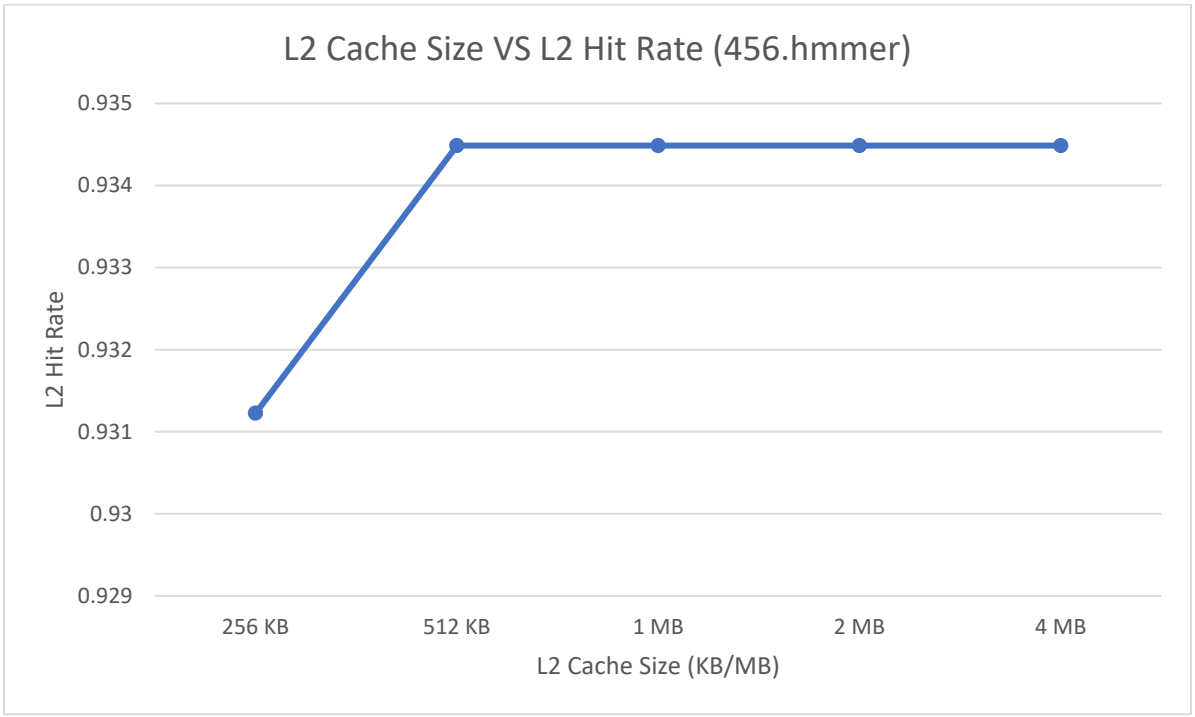
Analysis based on changes in L2 Cache Size:

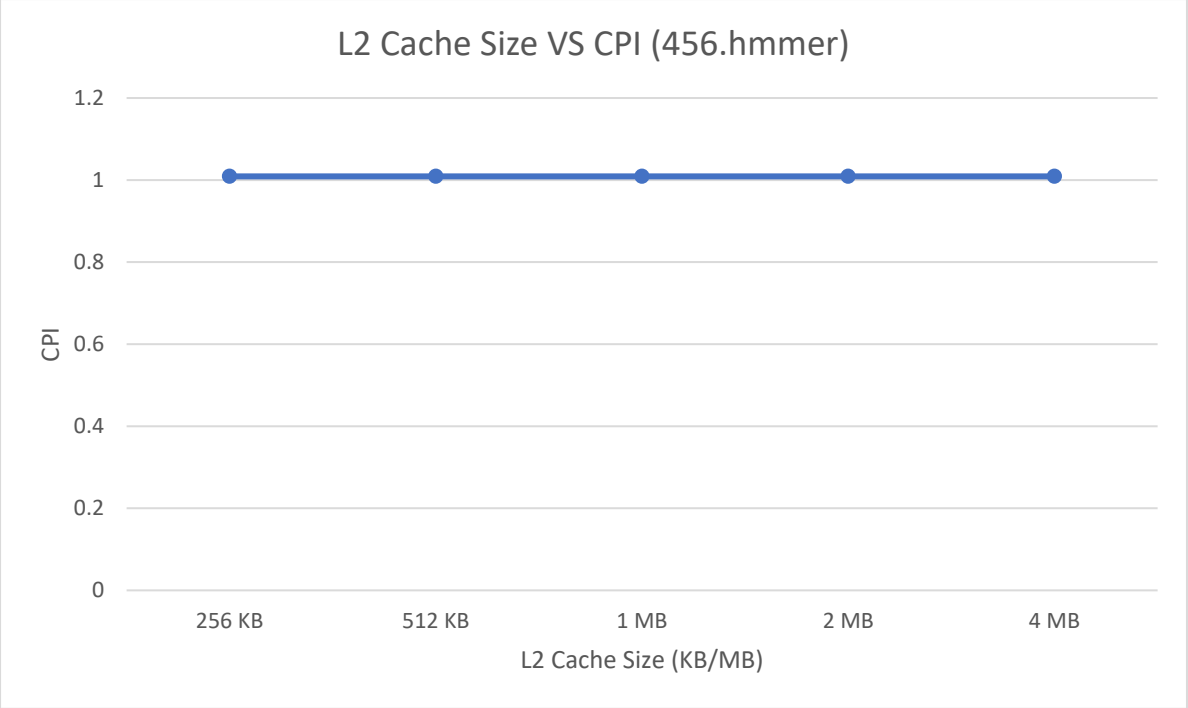
- Default values of Parameters: L1I Size: 128KB, L1D Cache Size:128KB, L1D Associativity: 2, L1I Associativity: 2, L2 Associativity:1, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L2 size (KB/MB)	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
256 KB	0.998138	0.999984	0.931225	1.009
512 KB	0.998138	0.999984	0.934487	1.009
1 MB	0.998138	0.999984	0.934487	1.009
2 MB	0.998138	0.999984	0.934487	1.009
4 MB	0.998138	0.999984	0.934487	1.009




L2 Cache Size VS L1I Hit Rate (456.hmmer)
1.2
1
0.8
0.6
0.4
0.2
0
L1I Hit Rate
256 KB
512 KB
1 MB
2 MB
4 MB
L2 Cache Size (KB/MB)


L2 Cache Size VS L2 Hit Rate (456.hmmer)
0.935
0.934
0.933
0.932
0.931
0.93
0.929
L2 Hit Rate
256 KB
512 KB
1 MB
2 MB
4 MB
L2 Cache Size (KB/MB)



Varying L1D Associativity:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=1 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=4 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

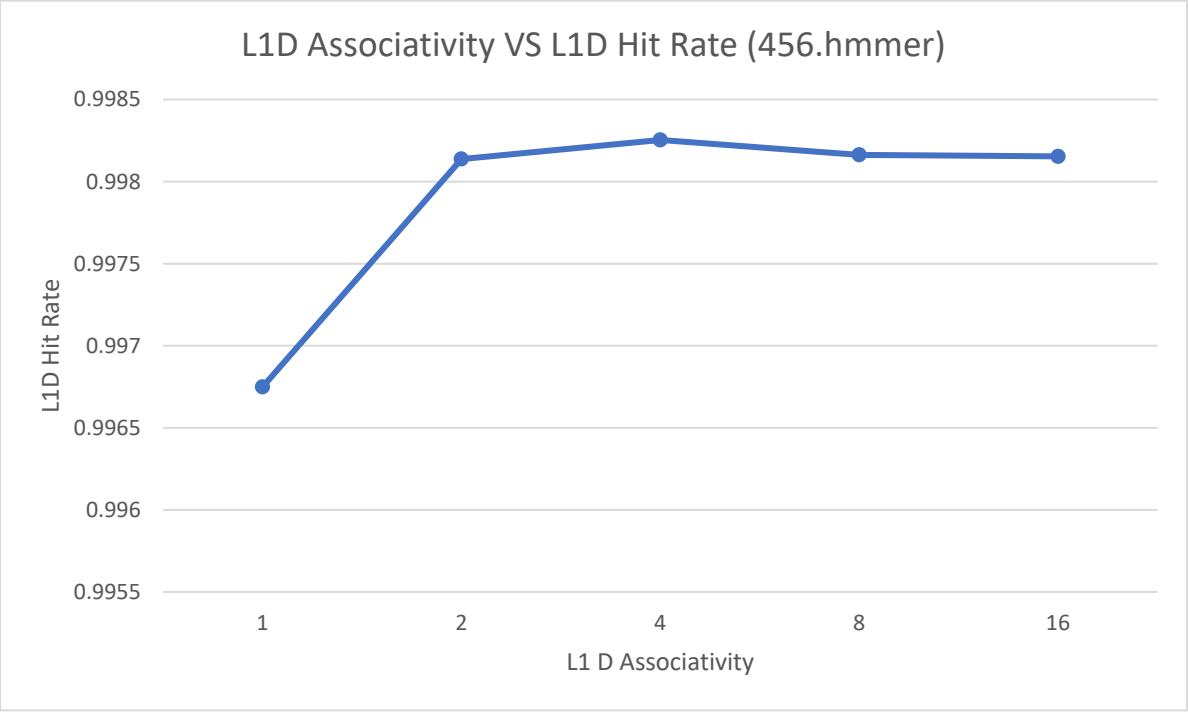
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=8 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

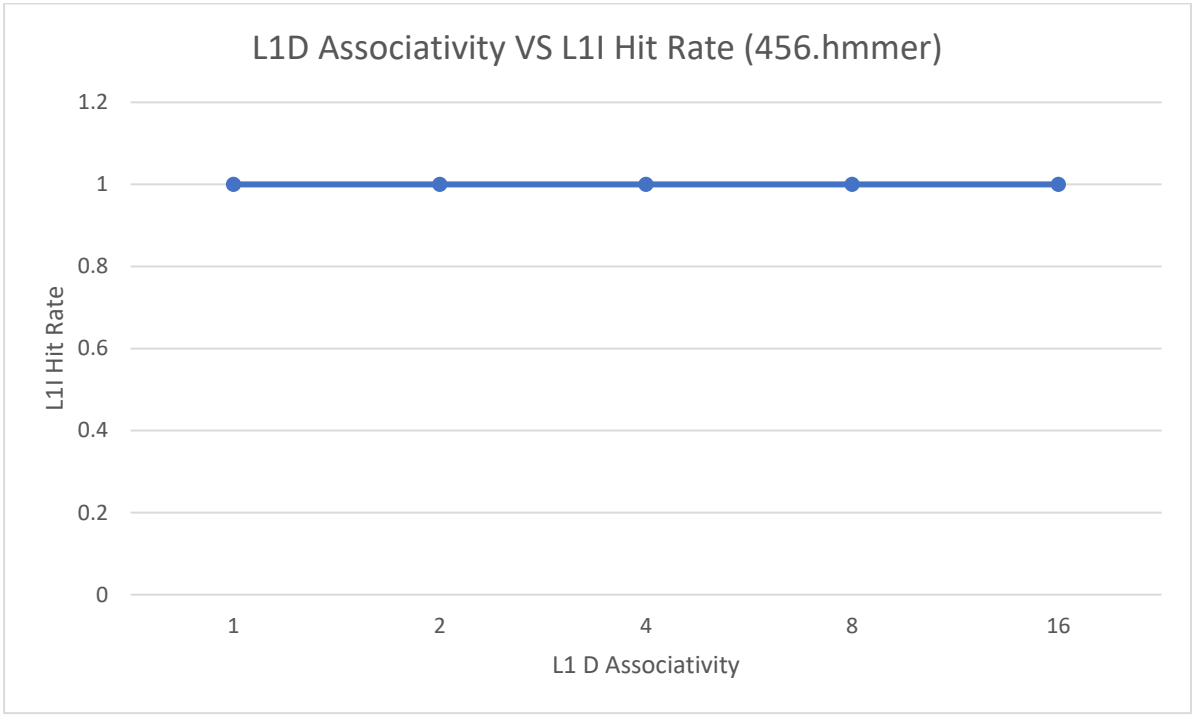
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=16 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

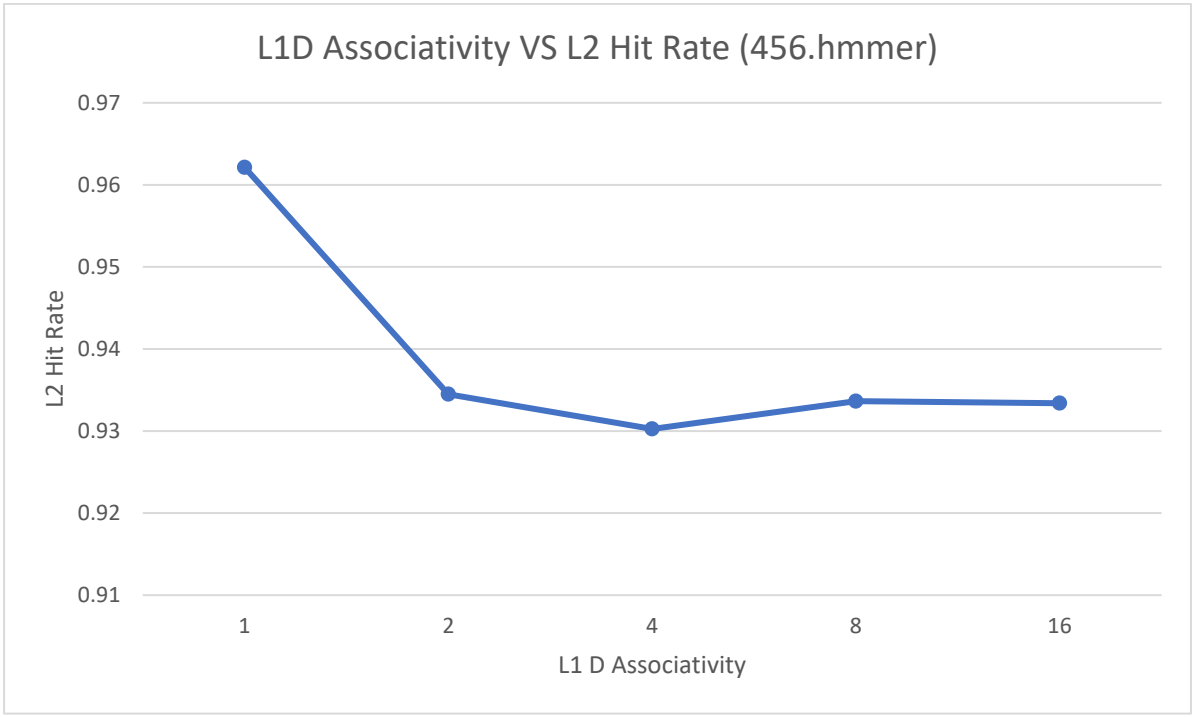
Analysis based on changes in L1D Associativity:

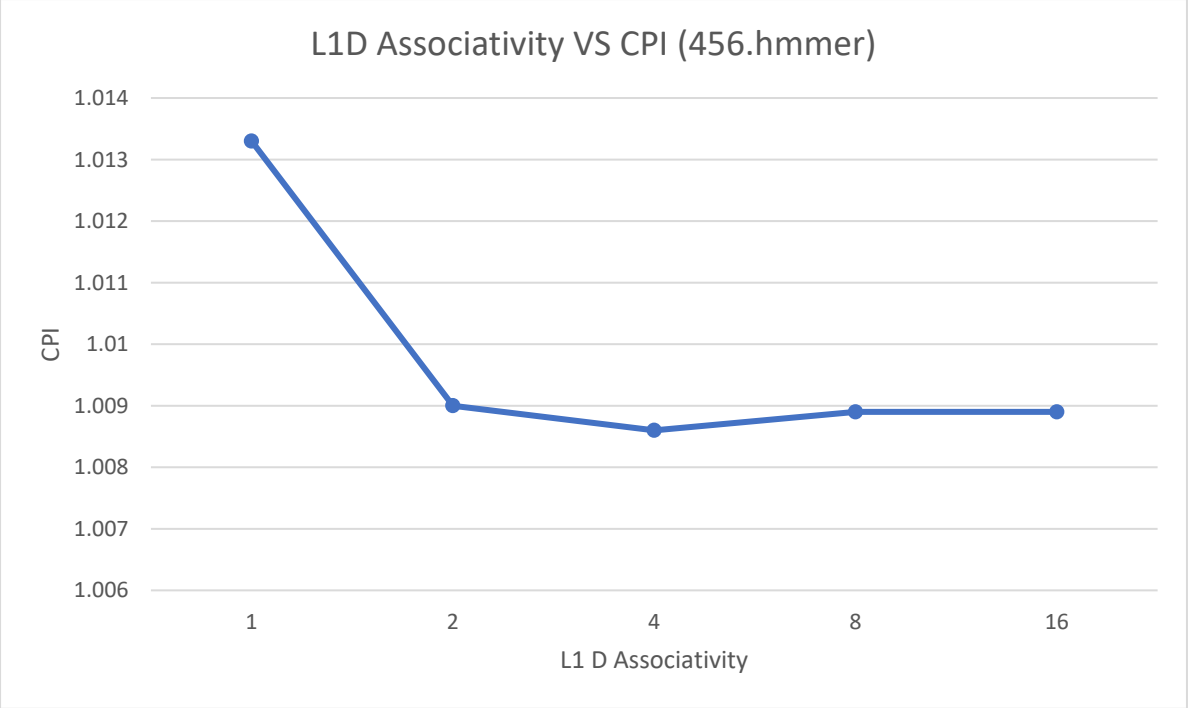
- Default values of Parameters: L1D Size:128KB, L1I Size: 128KB, L2 Cache Size:1MB, L1I Associativity: 2, L2 Associativity:1, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1D Associativity	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
1	0.996749	0.999984	0.962126	1.0133
2	0.998138	0.999984	0.934487	1.009
4	0.998254	0.999984	0.930254	1.0086
8	0.998163	0.999984	0.933641	1.0089
16	0.998154	0.999984	0.9333959	1.0089




L1D Associativity VS L1I Hit Rate (456.hmmer)
1.2
1
0.8
0.6
0.4
0.2
0
L1I Hit Rate
1
2
4
8
16
L1 D Associativity


L1D Associativity VS L2 Hit Rate (456.hmmer)
0.97
0.96
0.95
0.94
0.93
0.92
0.91
L2 Hit Rate
1
2
4
8
16
L1 D Associativity



Varying L1I Associativity:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=1 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=4 --l2_assoc=1 --cacheline_size=64

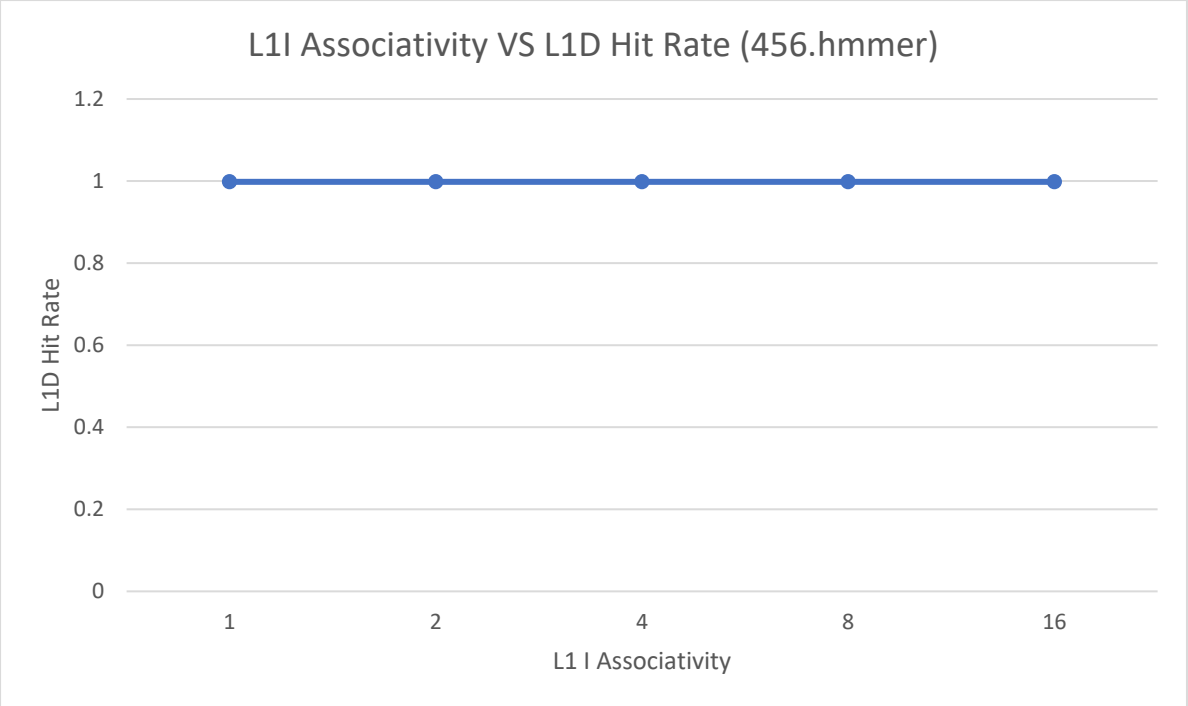
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=8 --l2_assoc=1 --cacheline_size=64

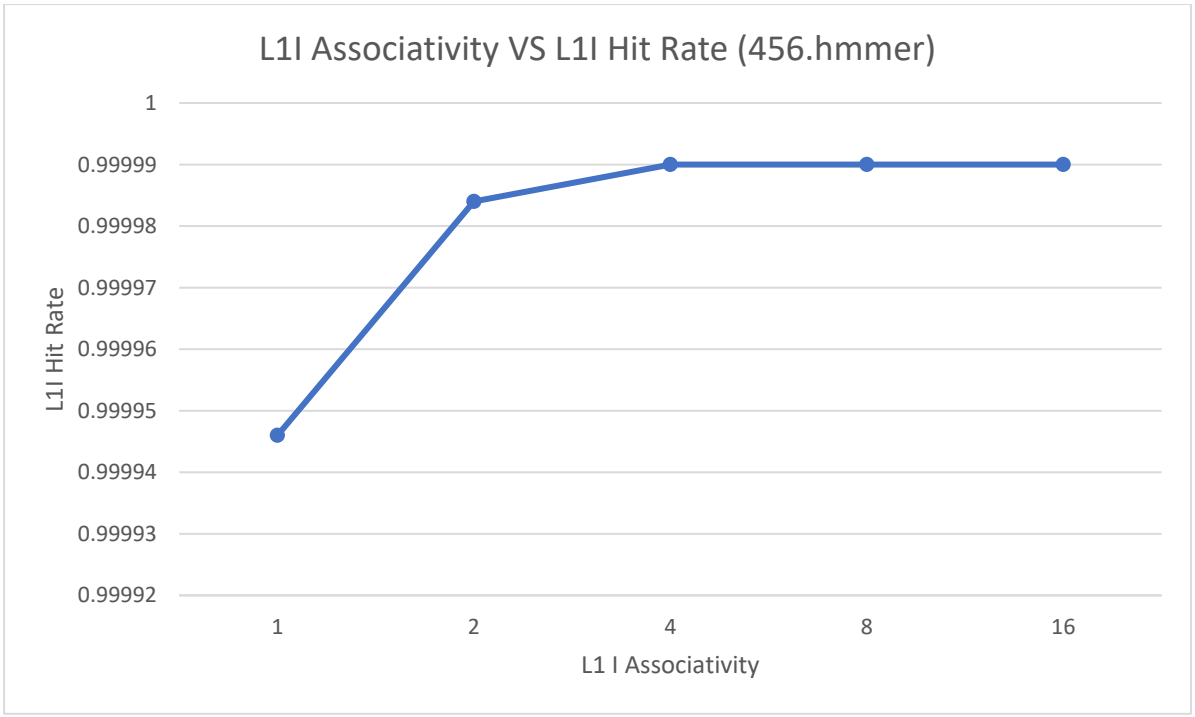
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=16 --l2_assoc=1 --cacheline_size=64

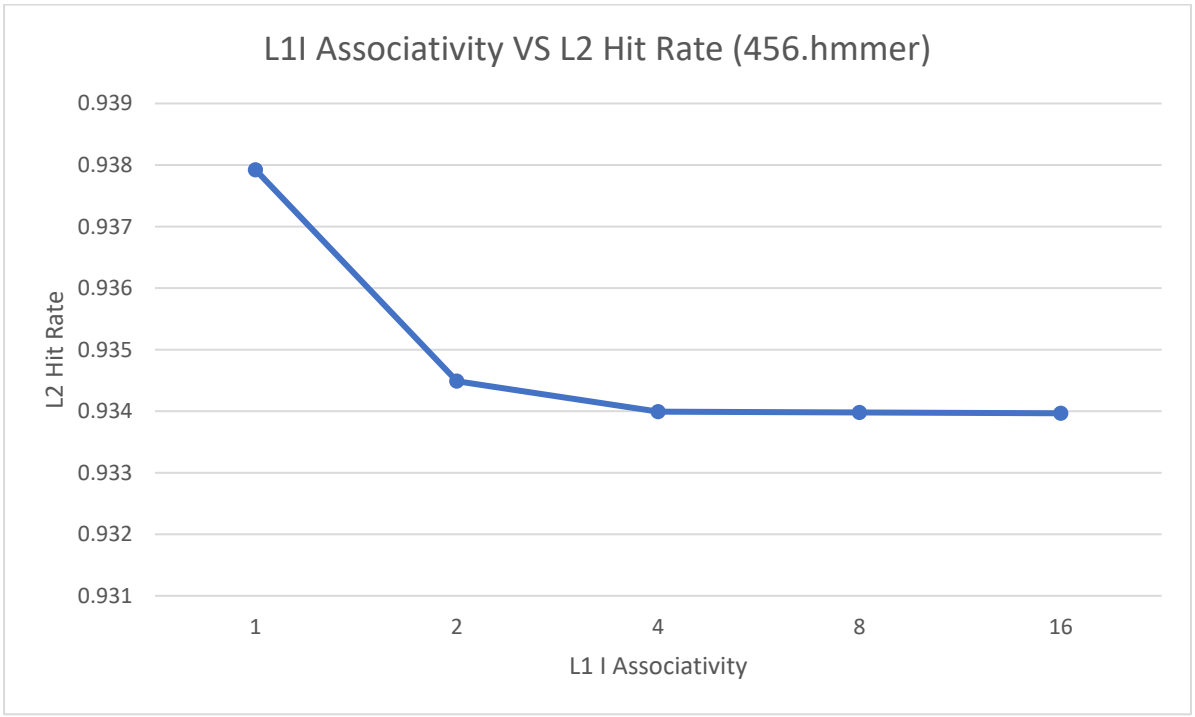
Analysis based on changes in L1I Associativity:

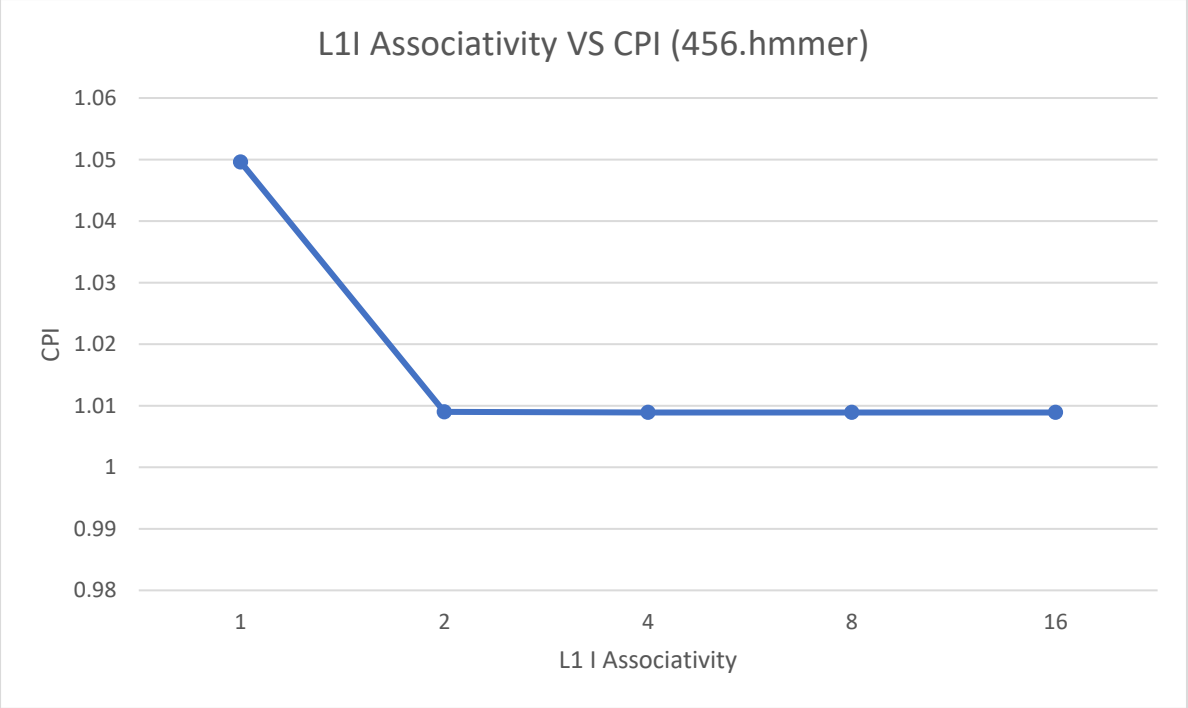
- Default values of Parameters: L1D Size:128KB, L1I Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L2 Associativity:1, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1I Associativity	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
1	0.998138	0.999946	0.937921	1.0496
2	0.998138	0.999984	0.934487	1.009
4	0.998138	0.99999	0.933992	1.0089
8	0.998138	0.99999	0.933979	1.0089
16	0.998138	0.99999	0.933966	1.0089




L1I Associativity VS L1I Hit Rate (456.hmmer)
1
0.99999
0.99998
0.99997
0.99996
0.99995
0.99994
0.99993
0.99992
L1I Hit Rate
1
2
4
8
16
L1 I Associativity


L1I Associativity VS L2 Hit Rate (456.hmmer)
0.939
0.938
0.937
0.936
0.935
0.934
0.933
0.932
0.931
L2 Hit Rate
1
2
4
8
16
L1 I Associativity



Varying L2 Associativity:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=2 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=4 --cacheline_size=64

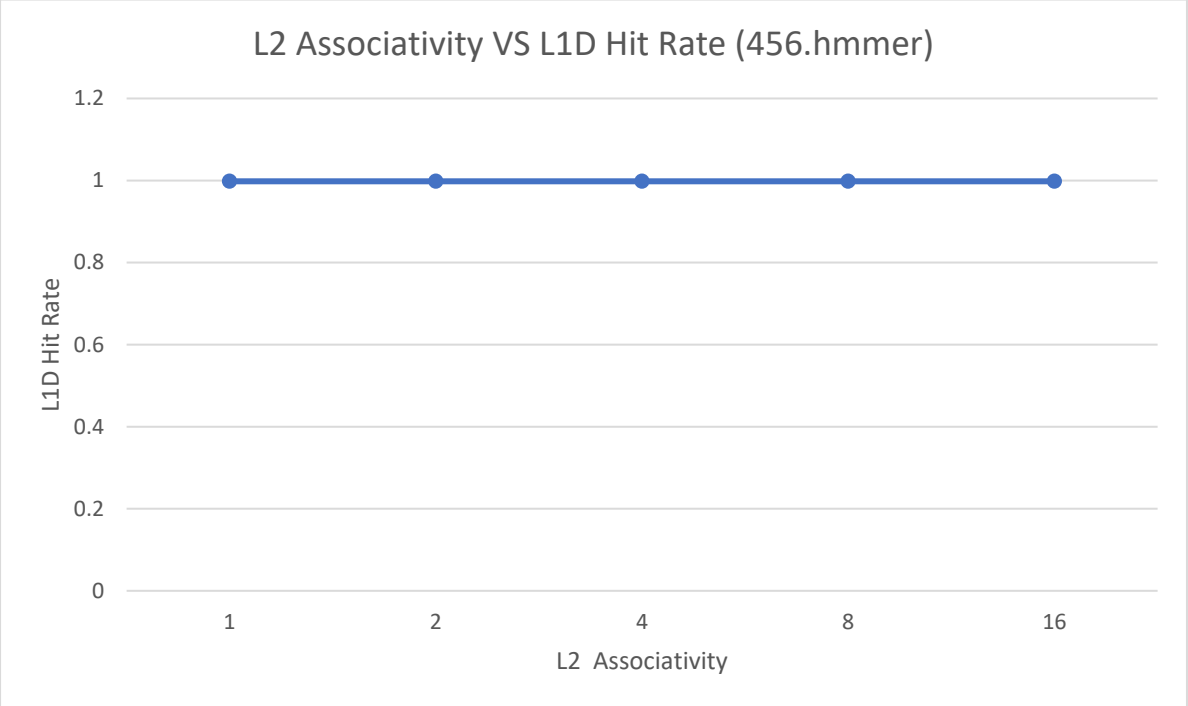
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=8 --cacheline_size=64

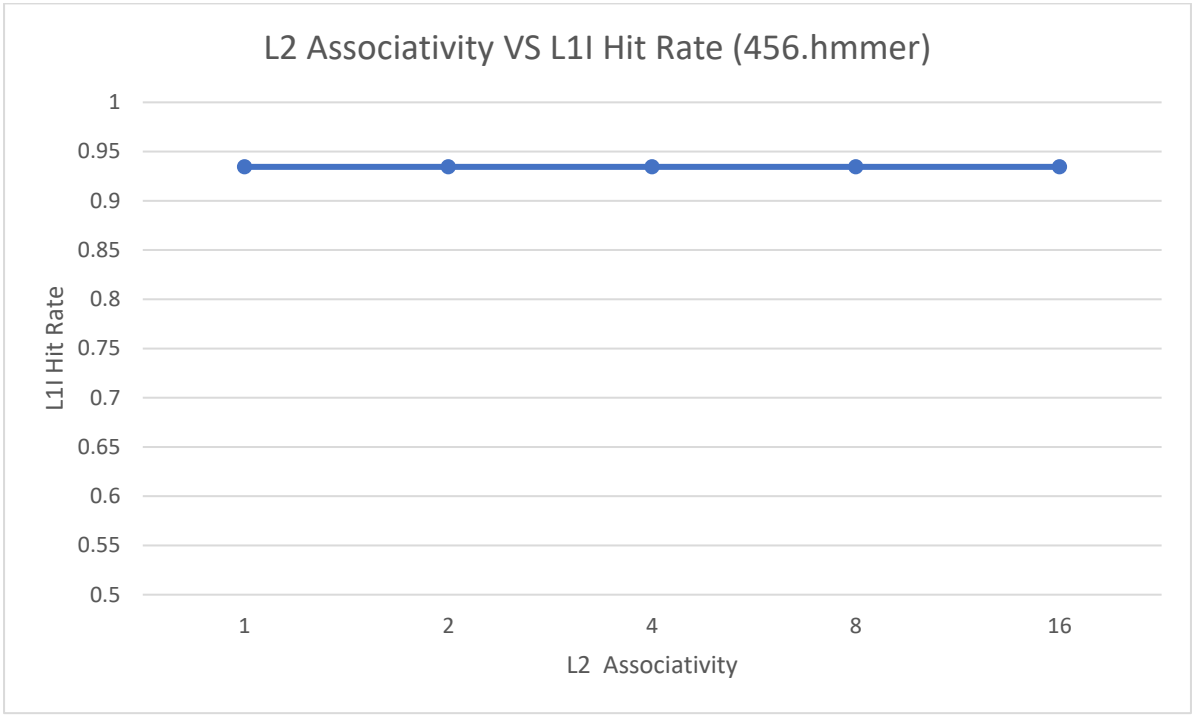
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=16 --cacheline_size=64

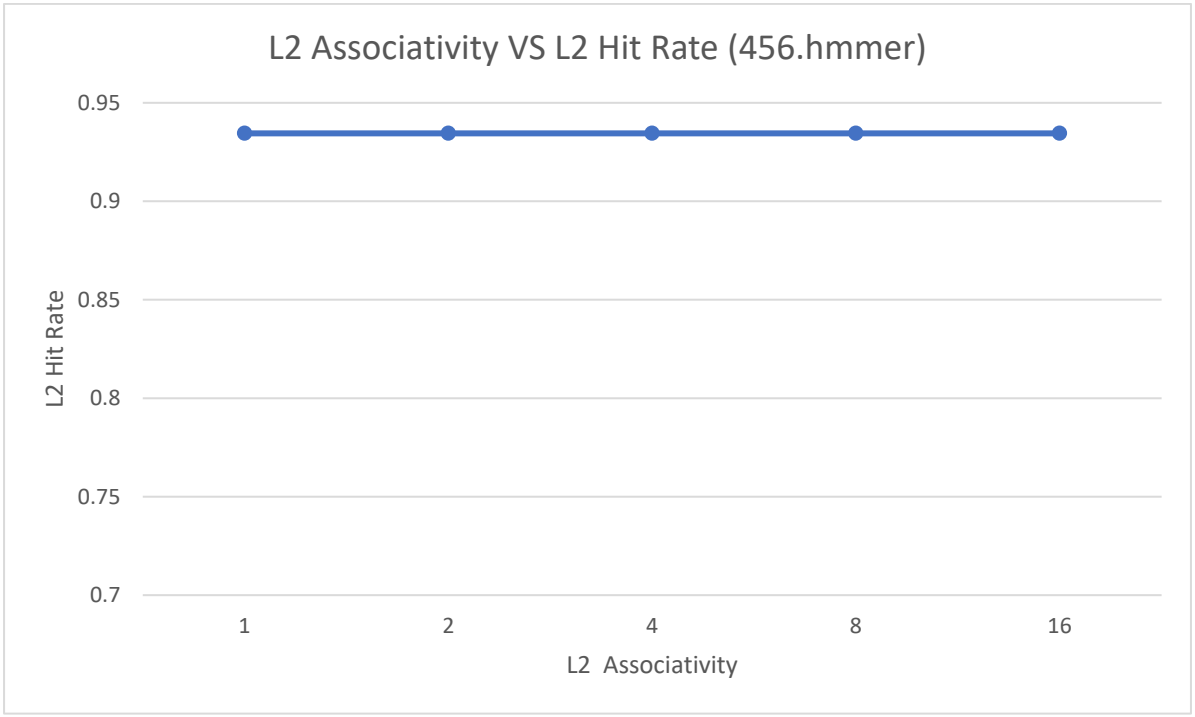
Analysis based on changes in L2 Associativity:

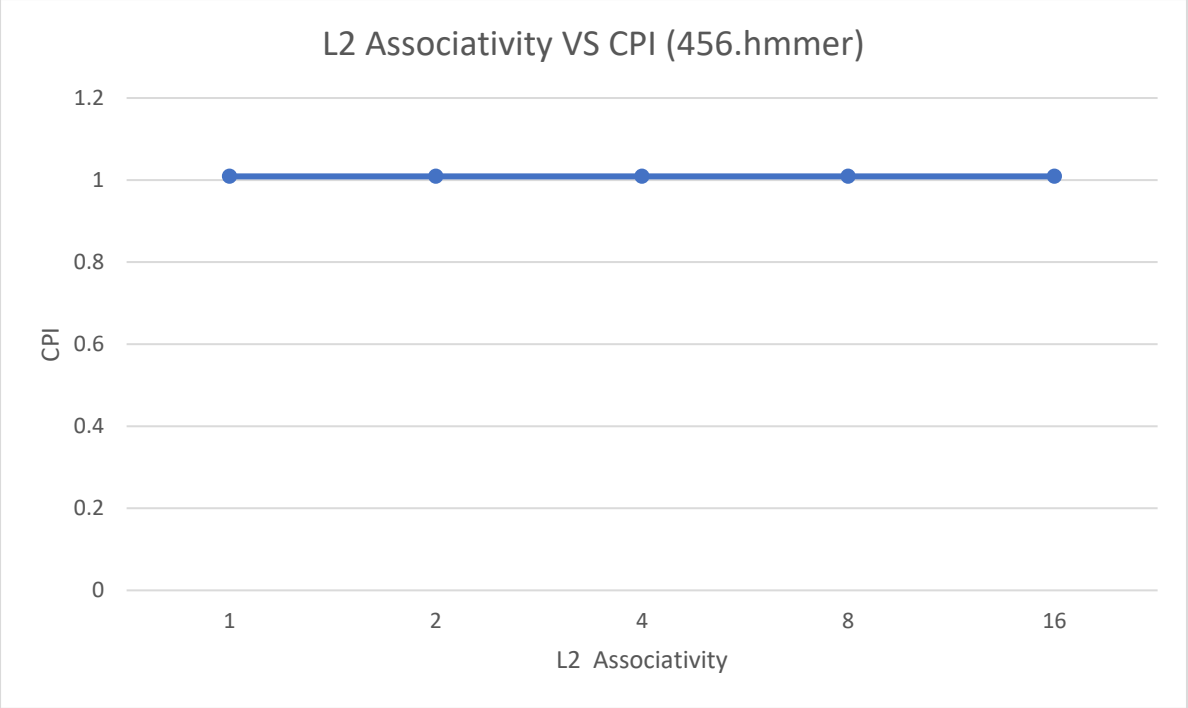
- Default values of Parameters: L1D Size:128KB, L1I Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L1I Associativity:2, Block Size: 64B
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L2 Associativity	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
1	0.998138	0.999984	0.934487	1.009
2	0.998138	0.999984	0.934487	1.009
4	0.998138	0.999984	0.934487	1.009
8	0.998138	0.999984	0.934487	1.009
16	0.998138	0.999984	0.934487	1.009




L2 Associativity VS L1I Hit Rate (456.hmmer)
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
L1I Hit Rate
1
2
4
8
16
L2 Associativity


L2 Associativity VS L2 Hit Rate (456.hmmer)
0.95
0.9
0.85
0.8
0.75
0.7
L2 Hit Rate
1
2
4
8
16
L2 Associativity



Varying Block Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=8

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=16

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=32

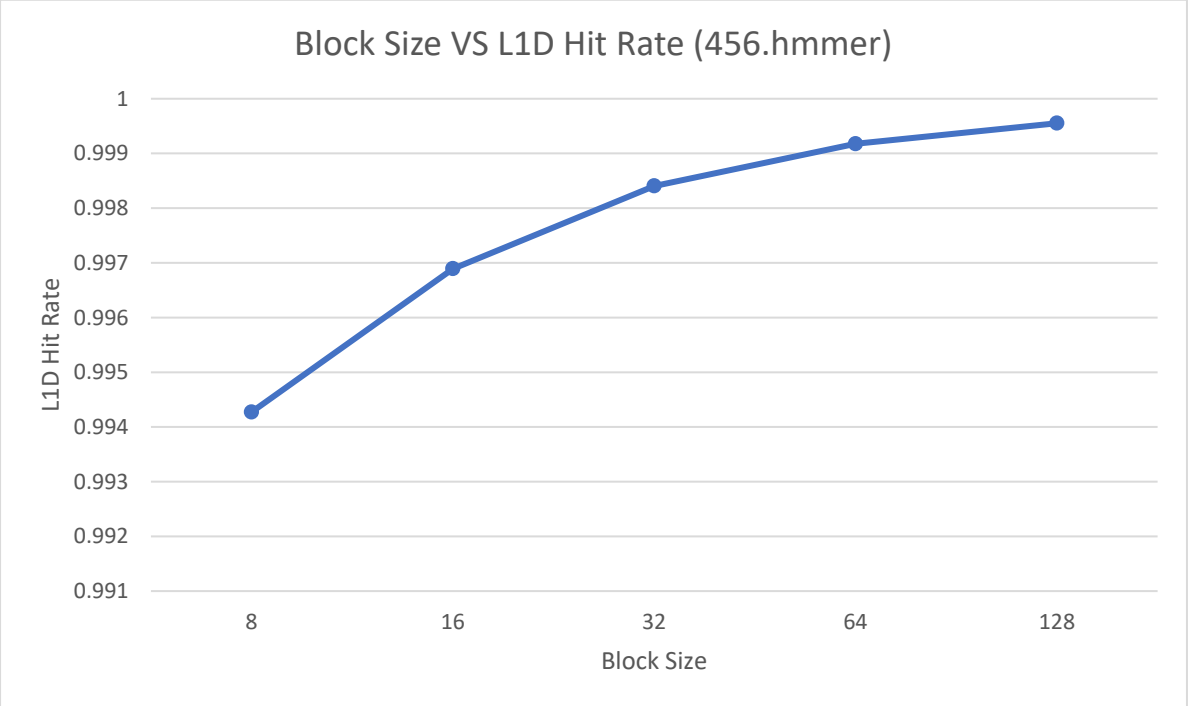
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

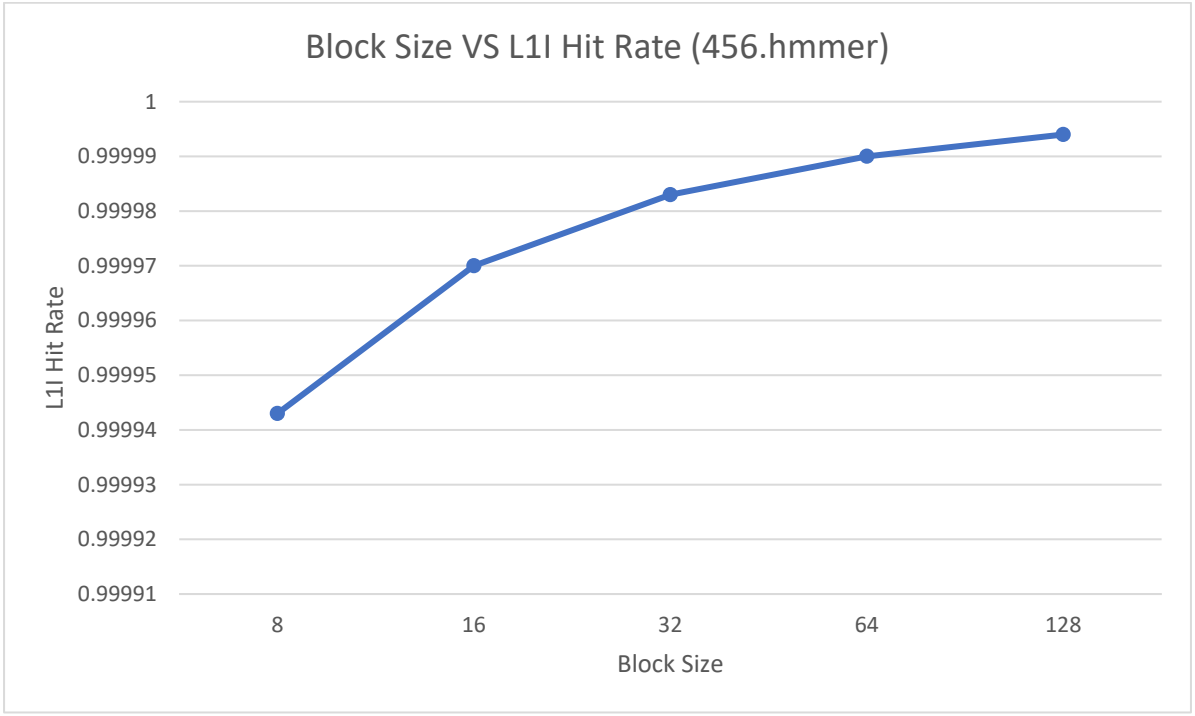
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=128

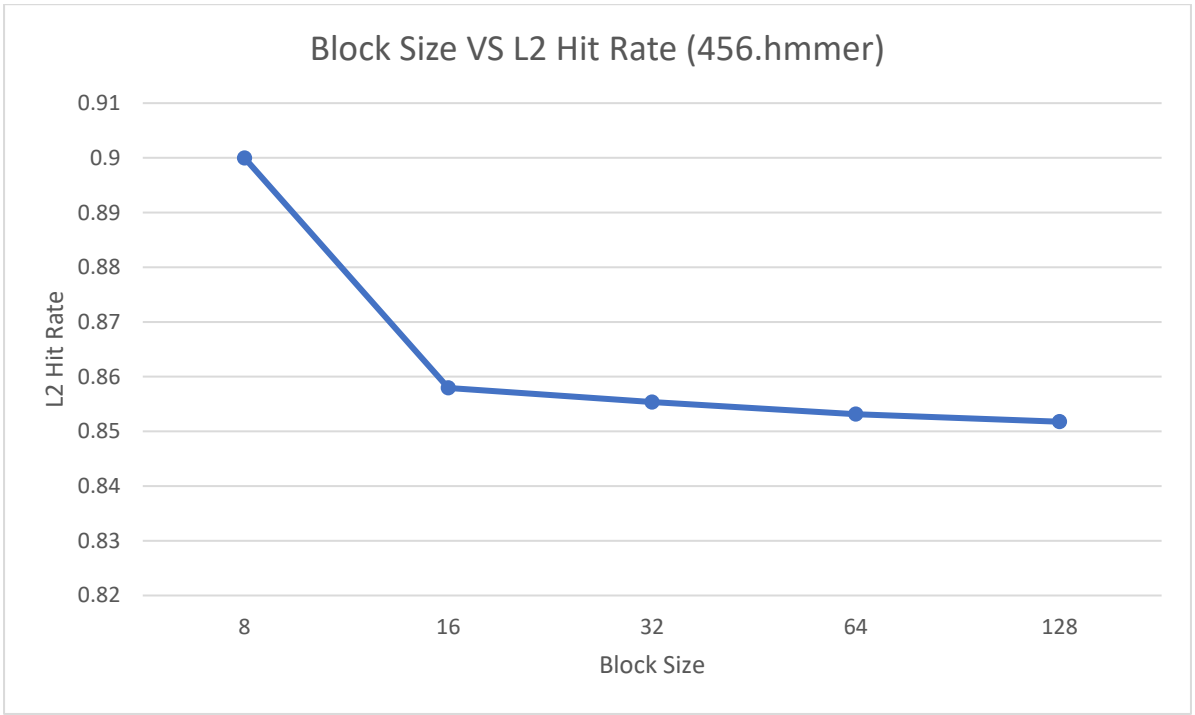
Analysis based on changes in Block Size:

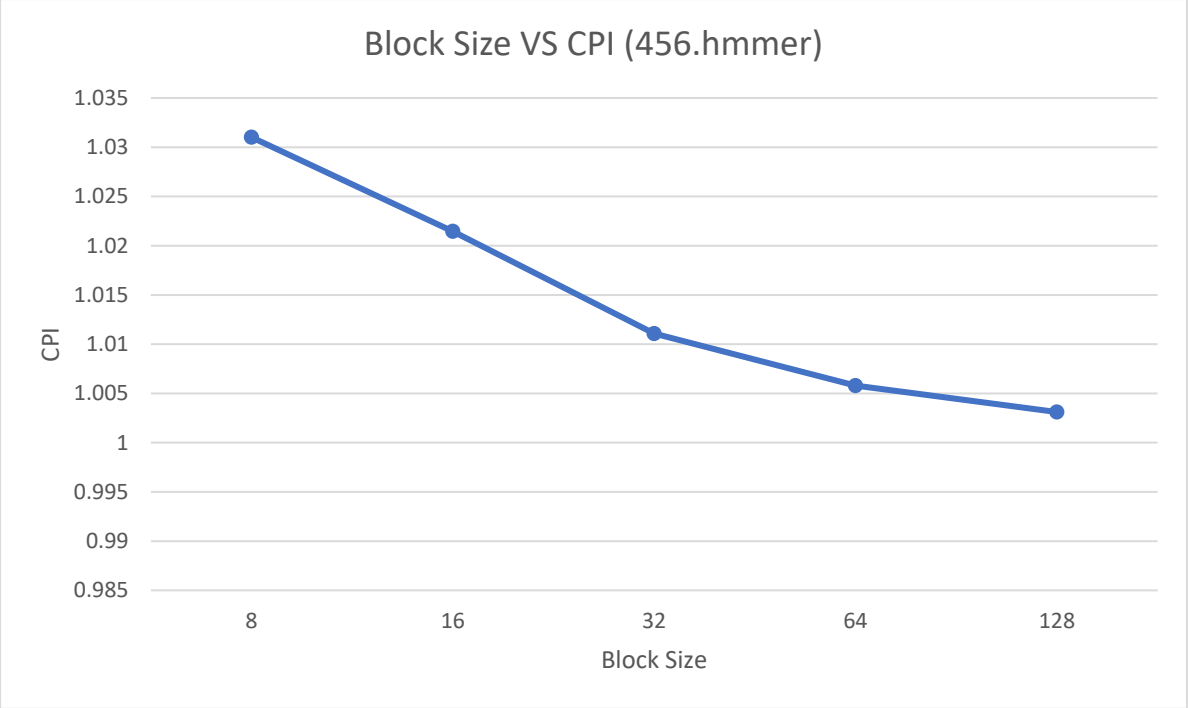
- Default values of Parameters: L1D Size:128KB, L1I Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L1I Associativity:2, L2 Associativity:1.
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

Block size	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
8	0.994273	0.999943	0.899958	1.0310101
16	0.996892	0.99997	0.857936	1.0214519
32	0.998406	0.999983	0.855351	1.01106906
64	0.999177	0.99999	0.853143	1.0057848
128	0.999552	0.999994	0.851764	1.00310956




Block Size VS L1I Hit Rate (456.hmmer)
1
0.99999
0.99998
0.99997
0.99996
0.99995
0.99994
0.99993
0.99992
0.99991
L1I Hit Rate
8
16
32
64
128
Block Size


Block Size VS L2 Hit Rate (456.hmmer)
0.91
0.9
0.89
0.88
0.87
0.86
0.85
0.84
0.83
0.82
L2 Hit Rate
8
16
32
64
128
Block Size



Bechmark 458.sjeng:

Varying L1D Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=32kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=64kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

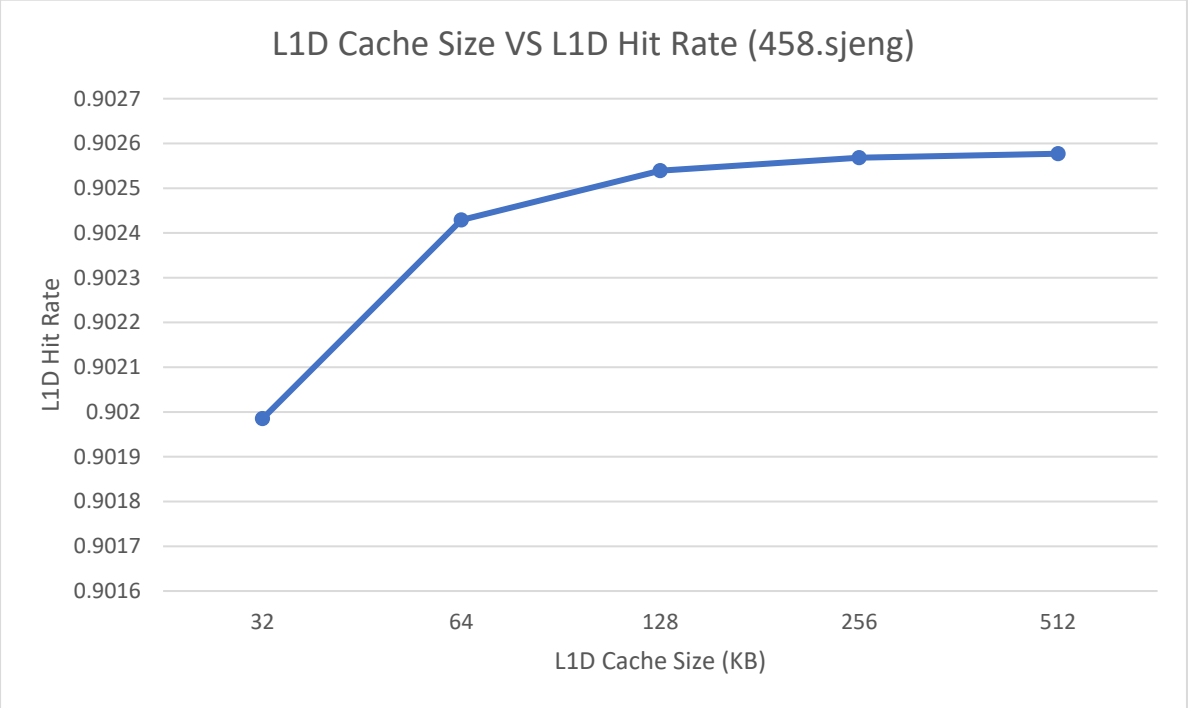
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=256kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

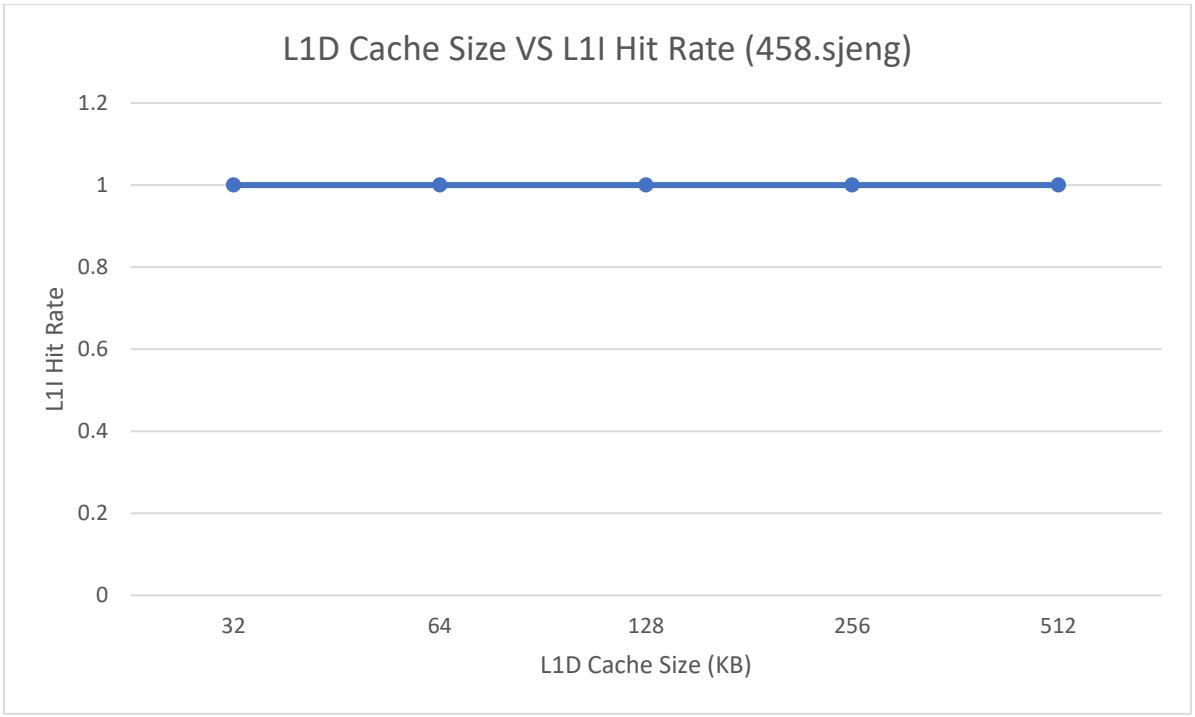
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=512kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

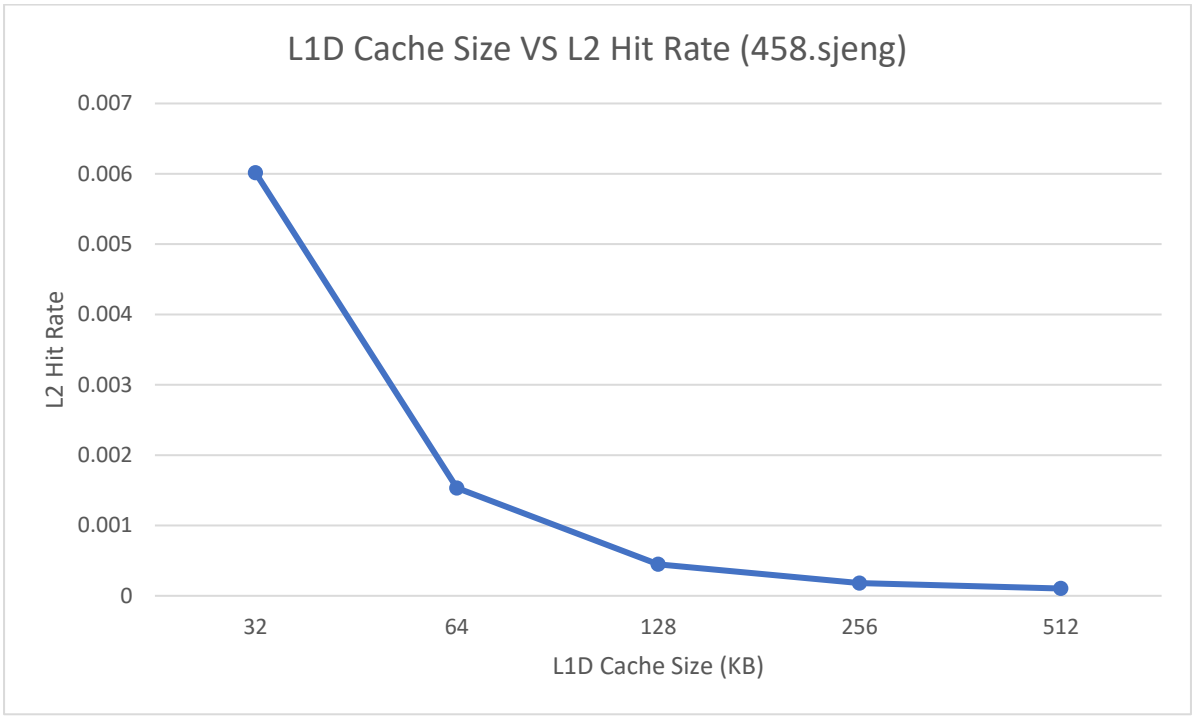
Analysis based on changes in L1D Cache Size:

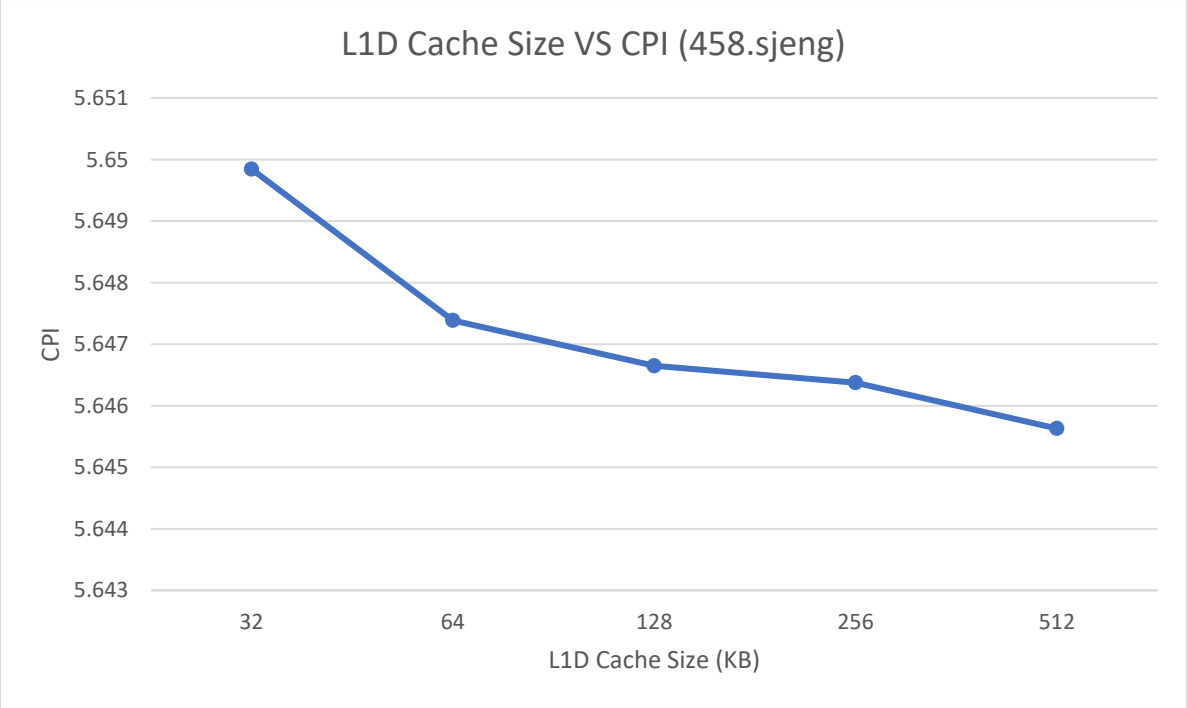
- Default values of Parameters: L1I Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L1I Associativity:2, L2 Associativity:1, Block Size:64B.
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1D size (KB)	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
32	0.901985	0.999984	0.006014	5.6498437
64	0.902429	0.999984	0.001532	5.6473865
128	0.902539	0.999984	0.000448	5.64664854
256	0.902568	0.999984	0.000181	5.64637384
512	0.902577	0.999984	0.000104	5.64562982




L1D Cache Size VS L1I Hit Rate (458.sjeng)
1.2
1
0.8
0.6
0.4
0.2
0
L1I Hit Rate
32
64
128
256
512
L1D Cache Size (KB)


L1D Cache Size VS L2 Hit Rate (458.sjeng)
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
L2 Hit Rate
32
64
128
256
512
L1D Cache Size (KB)



Varying L1I Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=32kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=64kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=256kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

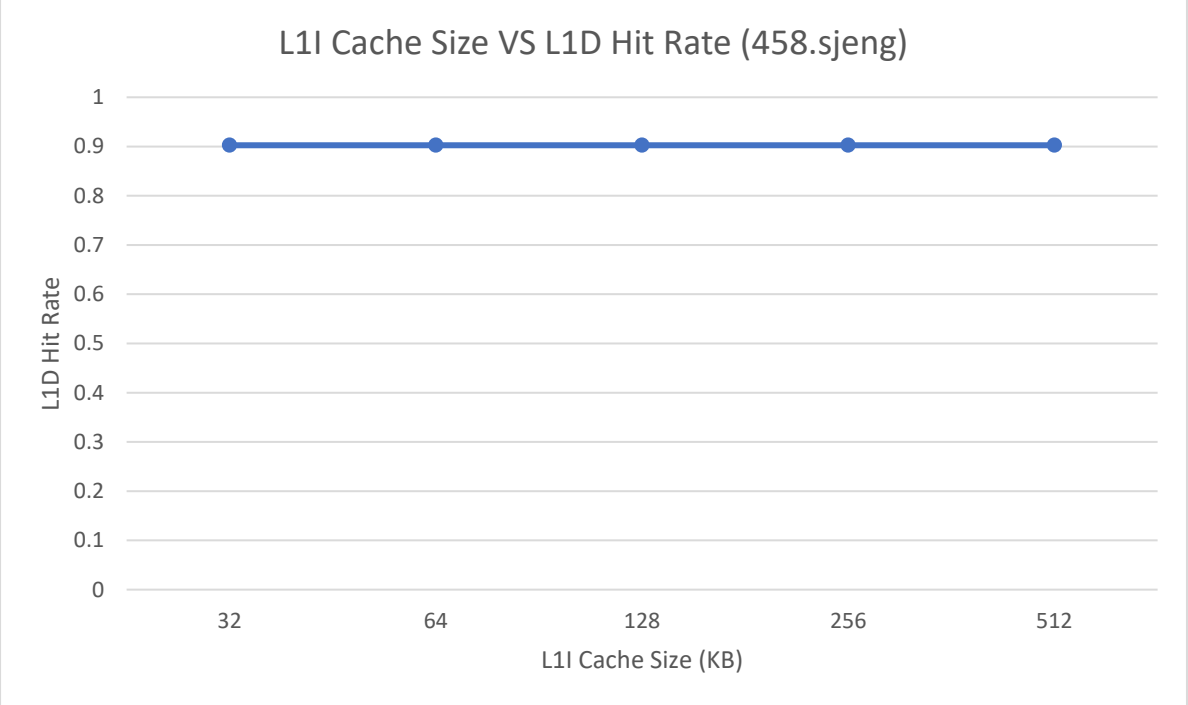
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --

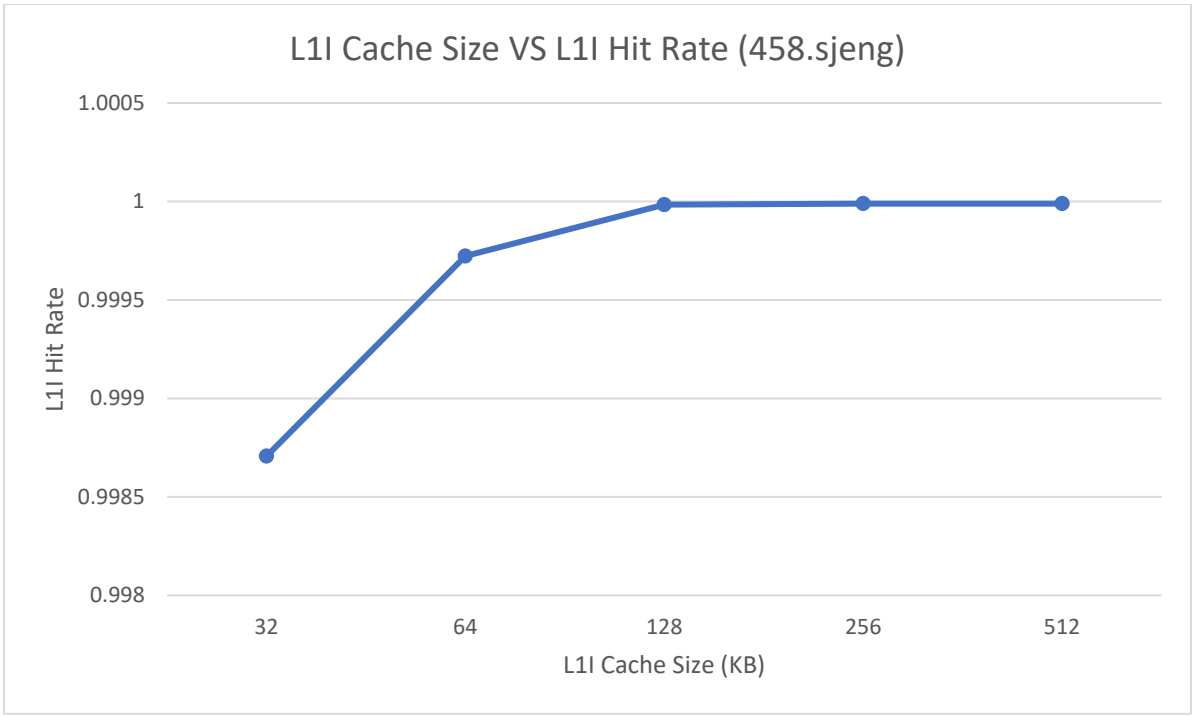
cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=512kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

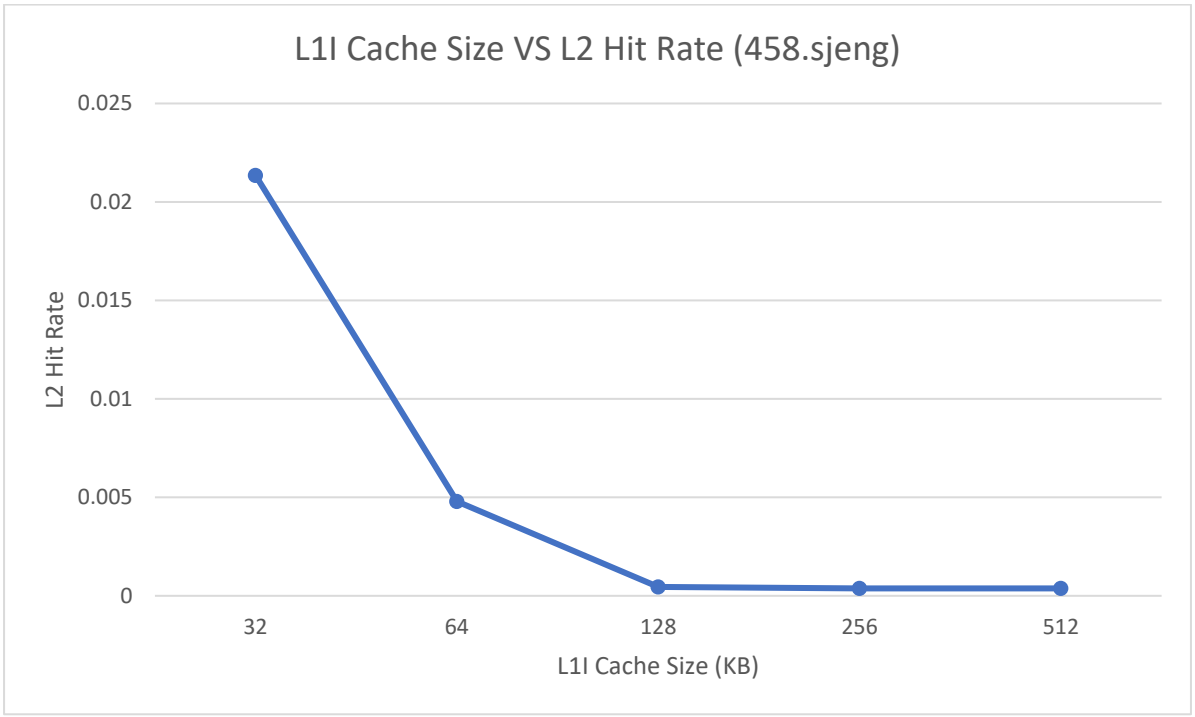
Analysis based on changes in L1I Cache Size:

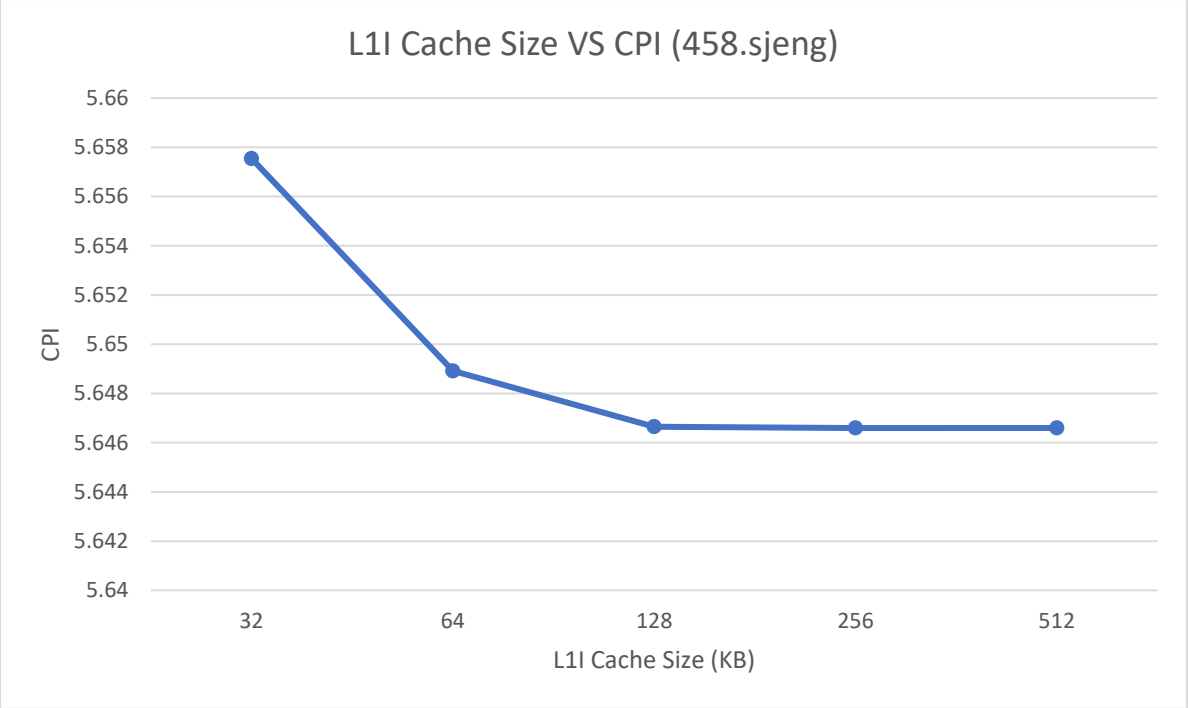
- Default values of Parameters: L1D Size: 128KB, L2 Cache Size:1MB, L1D Associativity: 2, L1I Associativity:2, L2 Associativity:1, Block Size:64B.
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1I size (KB)	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
32	0.902539	0.998707	0.021344	5.65754022
64	0.902539	0.999723	0.004788	5.64891182
128	0.902539	0.999984	0.000448	5.64664854
256	0.902539	0.999989	0.000377	5.64659604
512	0.902539	0.999989	0.000377	5.64659554




L1I Cache Size VS L1I Hit Rate (458.sjeng)
1.0005
1
0.9995
0.999
0.9985
0.998
L1I Hit Rate
32
64
128
256
512
L1I Cache Size (KB)


L1I Cache Size VS L2 Hit Rate (458.sjeng)
0.025
0.02
0.015
0.01
0.005
0
L2 Hit Rate
32
64
128
256
512
L1I Cache Size (KB)



Varying L2 Cache Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=256kB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=512kB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=2MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

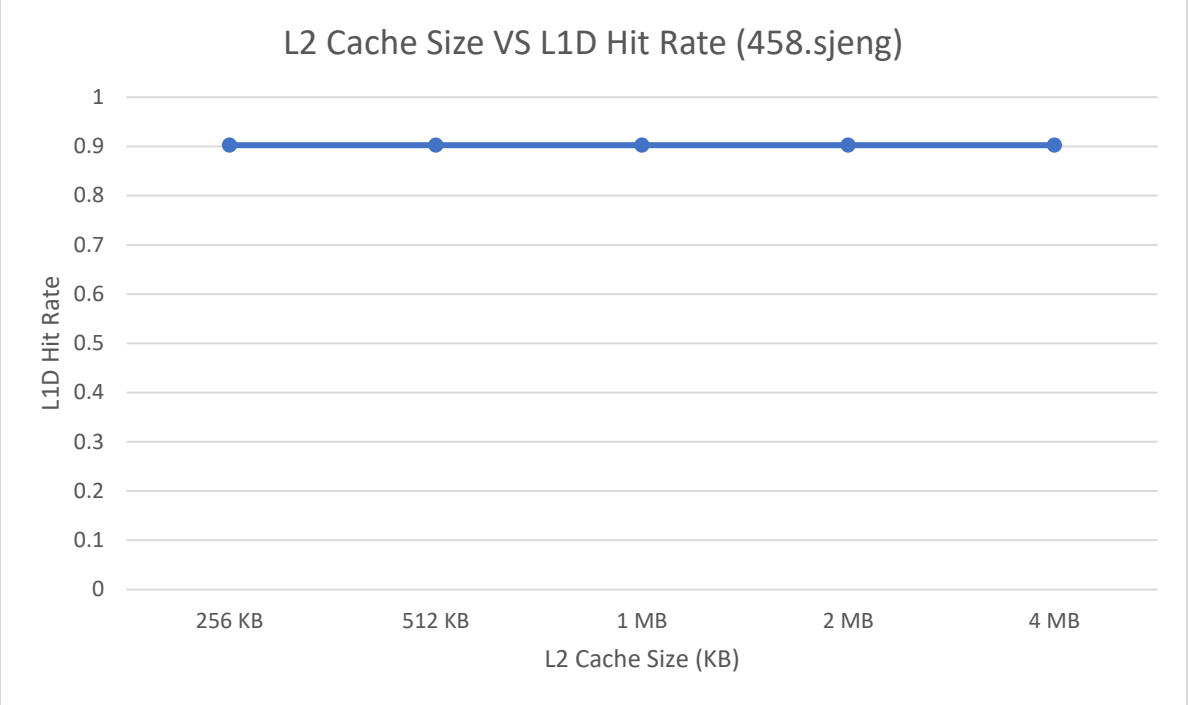
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --

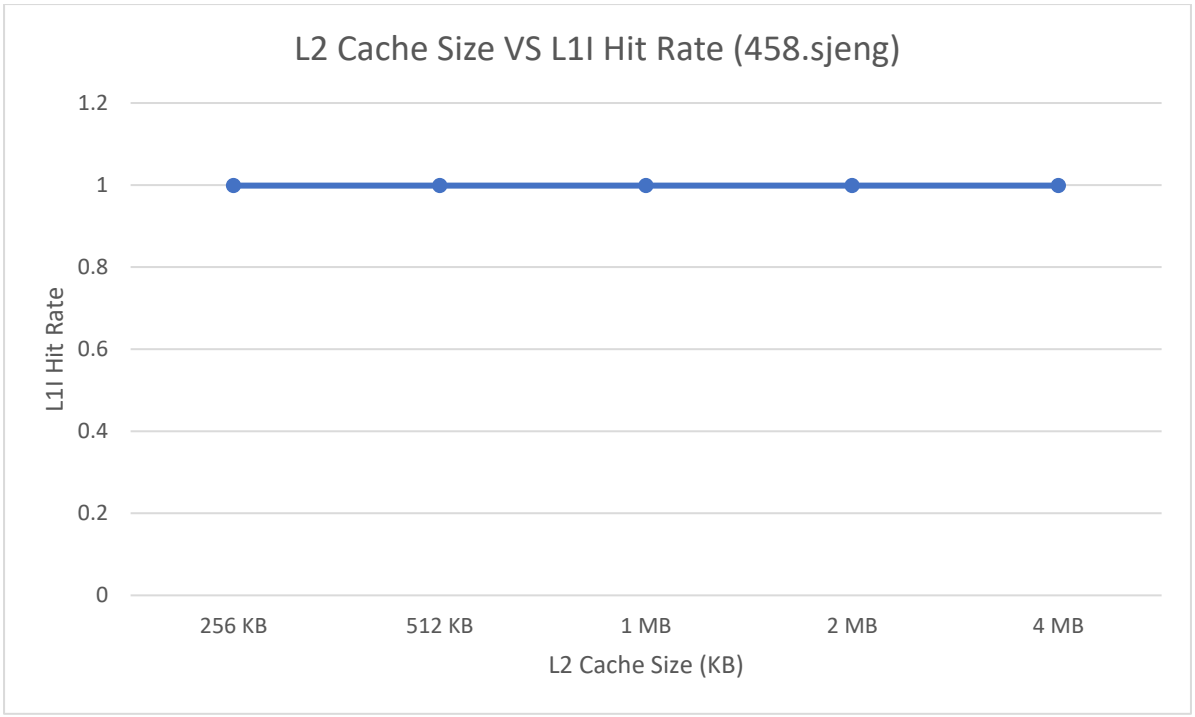
cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=4MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

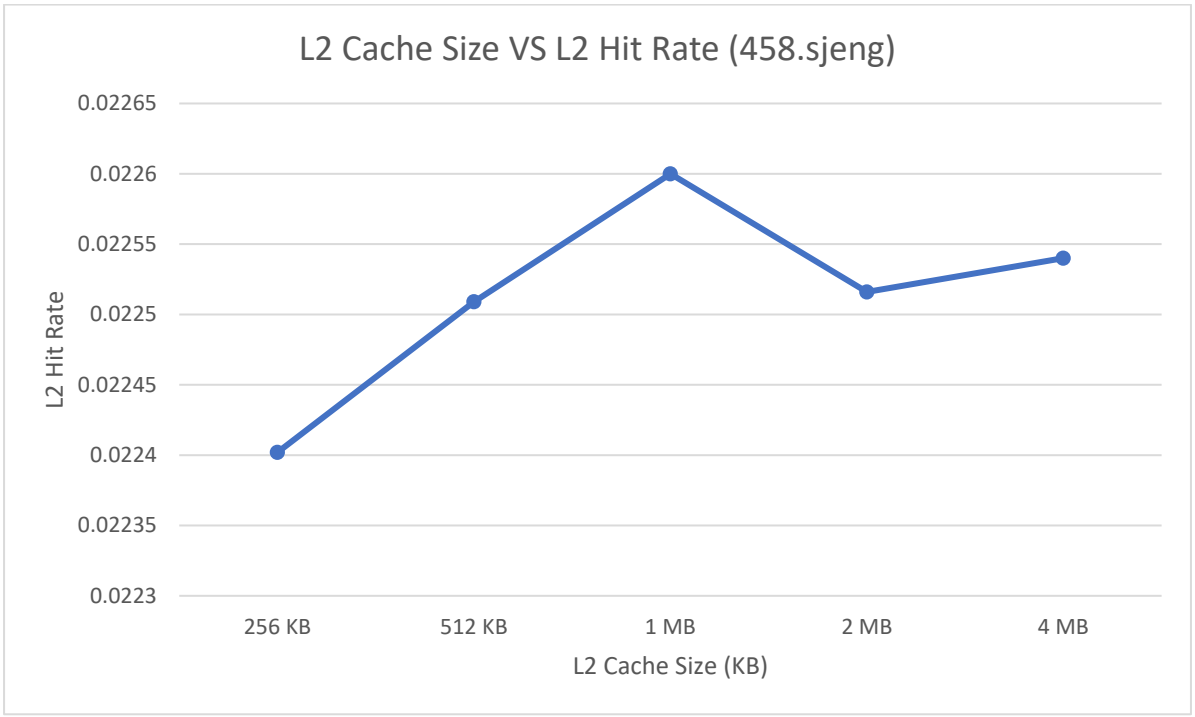
Analysis based on changes in L2 Cache Size:

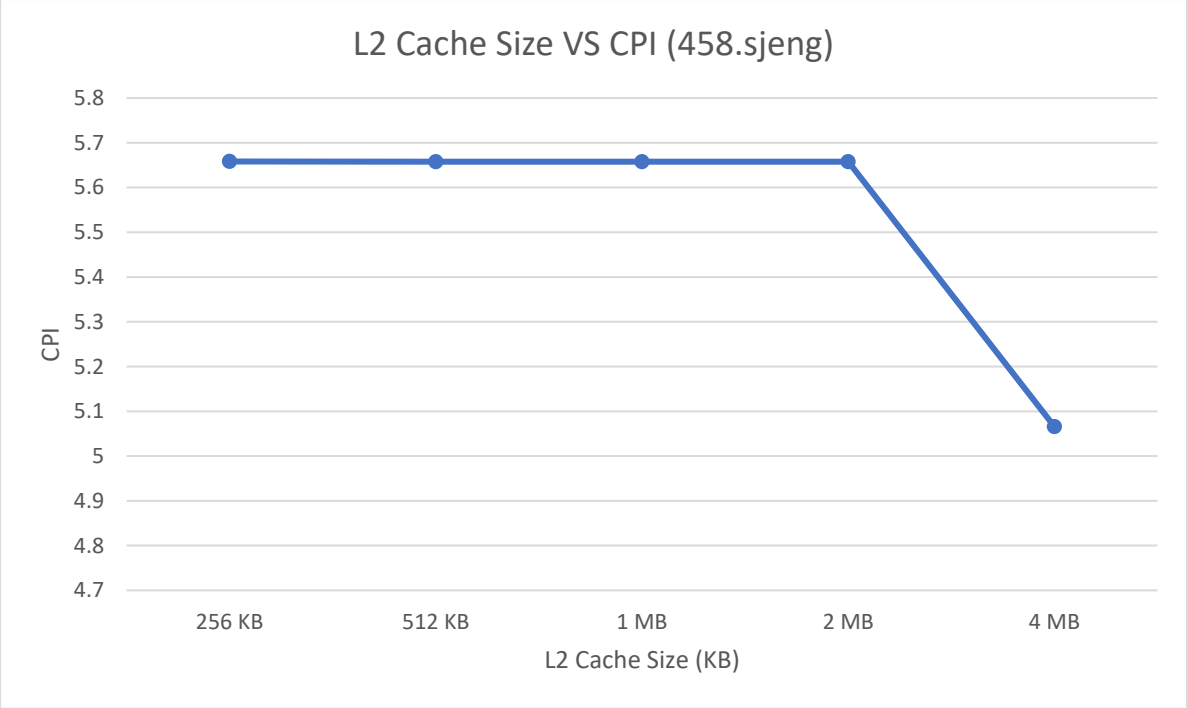
- Default values of Parameters: L1D Size: 128KB, L1I Cache Size:128KB, L1D Associativity: 2, L1I Associativity:2, L2 Associativity:1, Block Size:64B.
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L2 size (KB/MB)	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
256 KB	0.902429	0.998707	0.022402	5.6582
512 KB	0.902429	0.998707	0.022509	5.6577
1 MB	0.902429	0.998707	0.0226	5.6577
2 MB	0.902429	0.998707	0.022516	5.6577
4 MB	0.902429	0.998707	0.02254	5.0657




L2 Cache Size VS L1I Hit Rate (458.sjeng)
1.2
1
0.8
0.6
0.4
0.2
0
L1I Hit Rate
256 KB
512 KB
1 MB
2 MB
4 MB
L2 Cache Size (KB)


L2 Cache Size VS L2 Hit Rate (458.sjeng)
0.02265
0.0226
0.02255
0.0225
0.02245
0.0224
0.02235
0.0223
L2 Hit Rate
256 KB
512 KB
1 MB
2 MB
4 MB
L2 Cache Size (KB)



Varying L1D Associativity:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=1 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=4 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=8 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

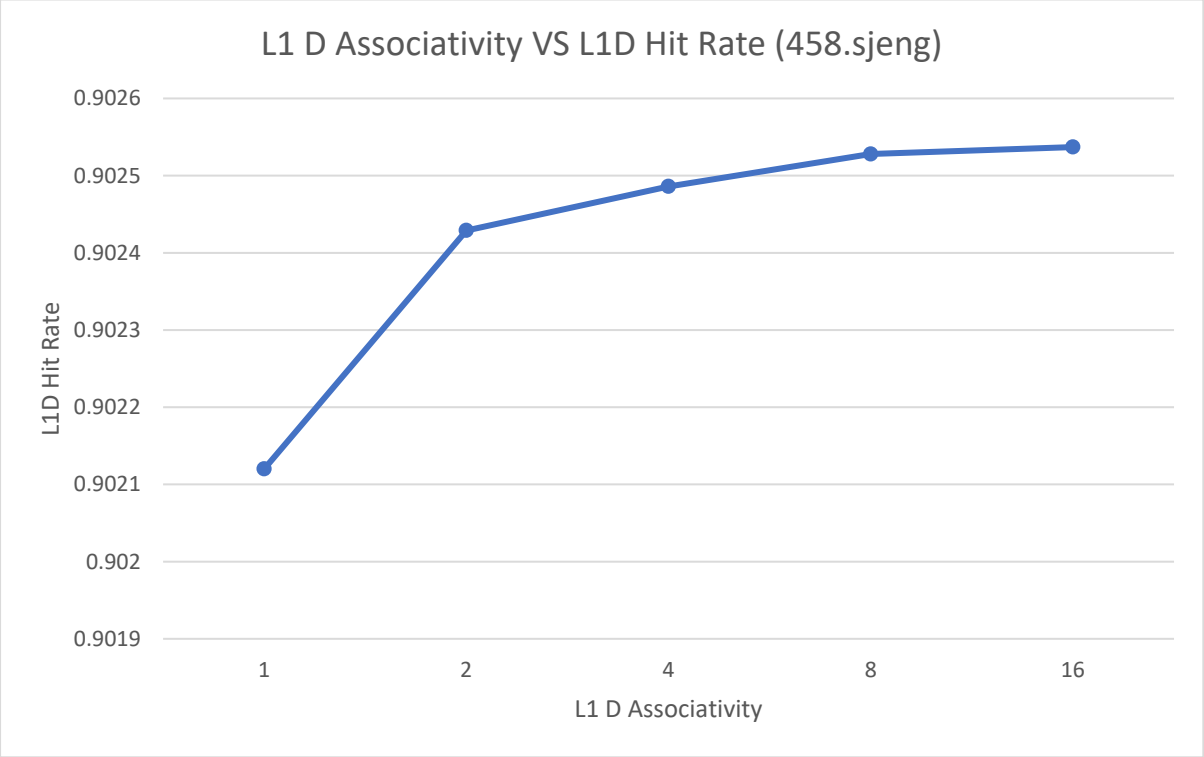
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --

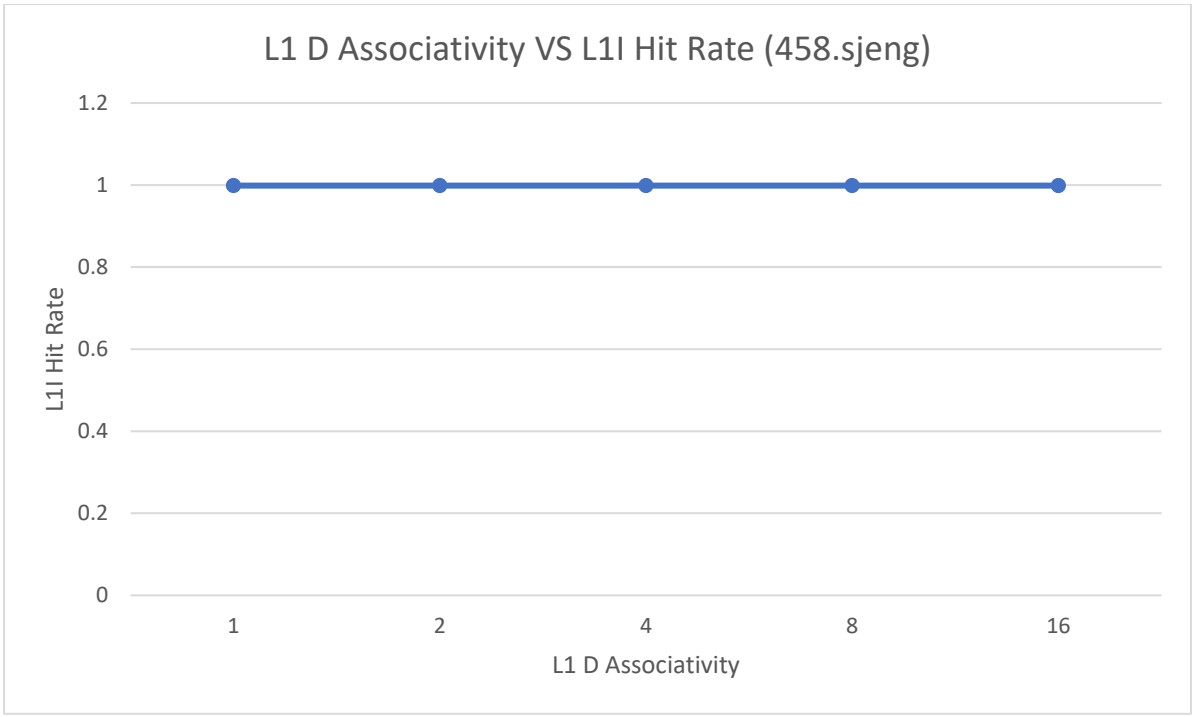
cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=16 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

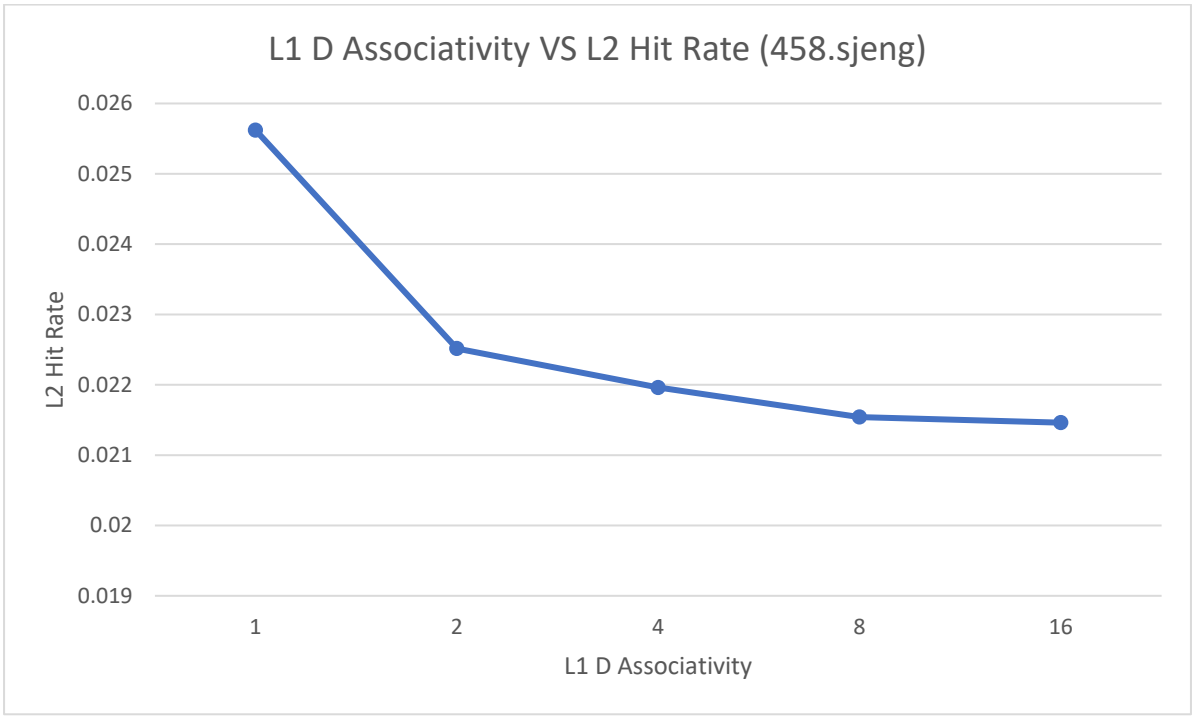
Analysis based on changes in L1D Associativity:

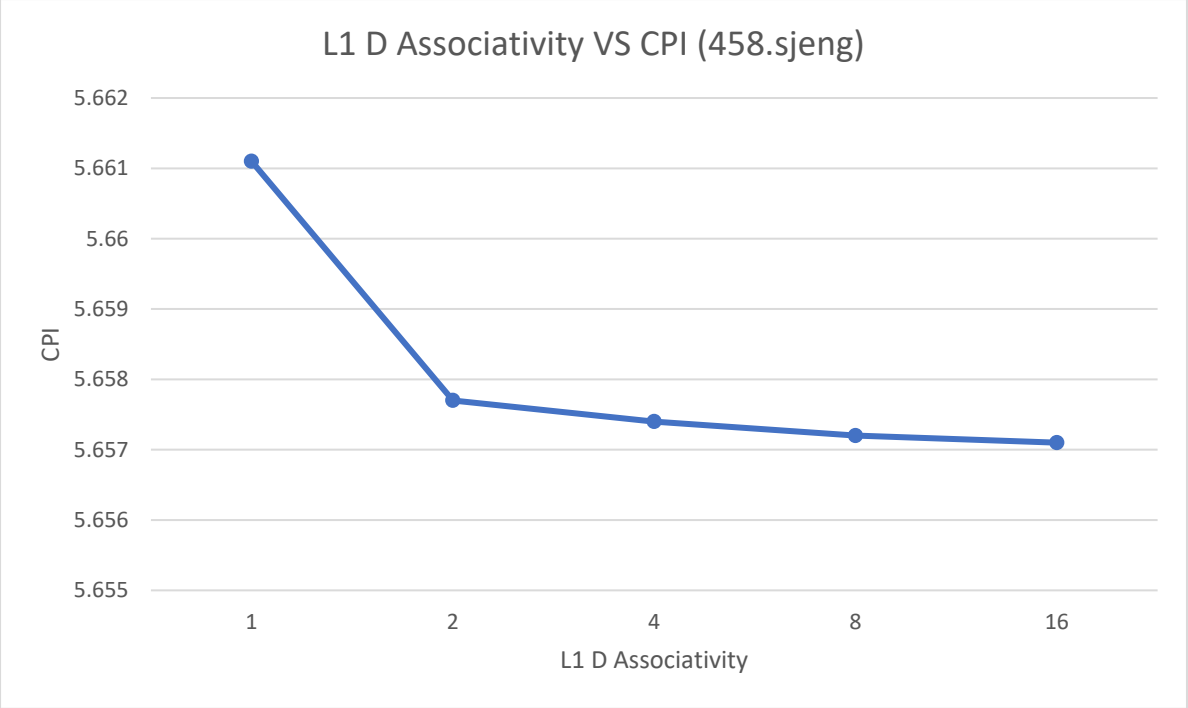
- Default values of Parameters: L1D Size: 128KB, L1I Cache Size:128KB, L2 Cache Size:1MB, L1I Associativity:2, L2 Associativity:1, Block Size:64B.
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1D Associativity	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
1	0.90212	0.998707	0.02562	5.6611
2	0.902429	0.998707	0.022516	5.6577
4	0.902486	0.998707	0.021962	5.6574
8	0.902528	0.998707	0.021542	5.6572
16	0.902537	0.998707	0.021462	5.6571




L1 D Associativity VS L1I Hit Rate (458.sjeng)
1.2
1
0.8
0.6
0.4
0.2
0
L1I Hit Rate
1
2
4
8
16
L1 D Associativity


L1 D Associativity VS L2 Hit Rate (458.sjeng)
0.026
0.025
0.024
0.023
0.022
0.021
0.02
0.019
L2 Hit Rate
1
2
4
8
16
L1 D Associativity



Varying L1I Associativity:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=1 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=4 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=8 --l2_assoc=1 --cacheline_size=64

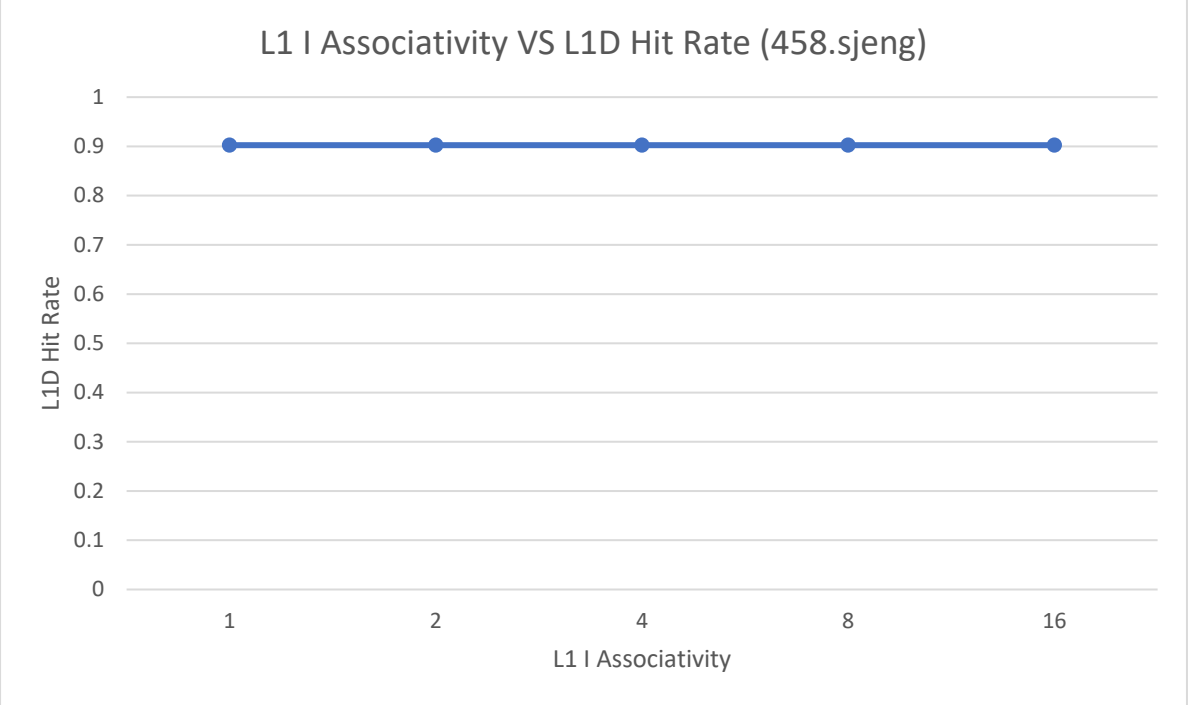
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --

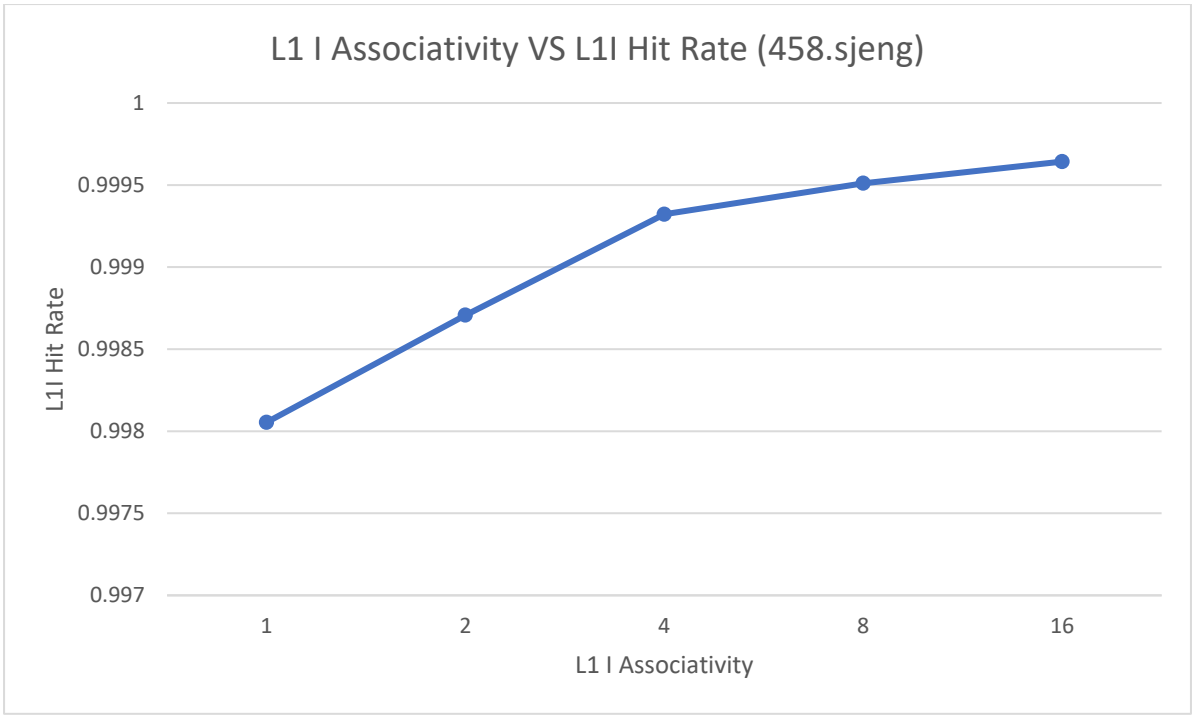
cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=16 --l2_assoc=1 --cacheline_size=64

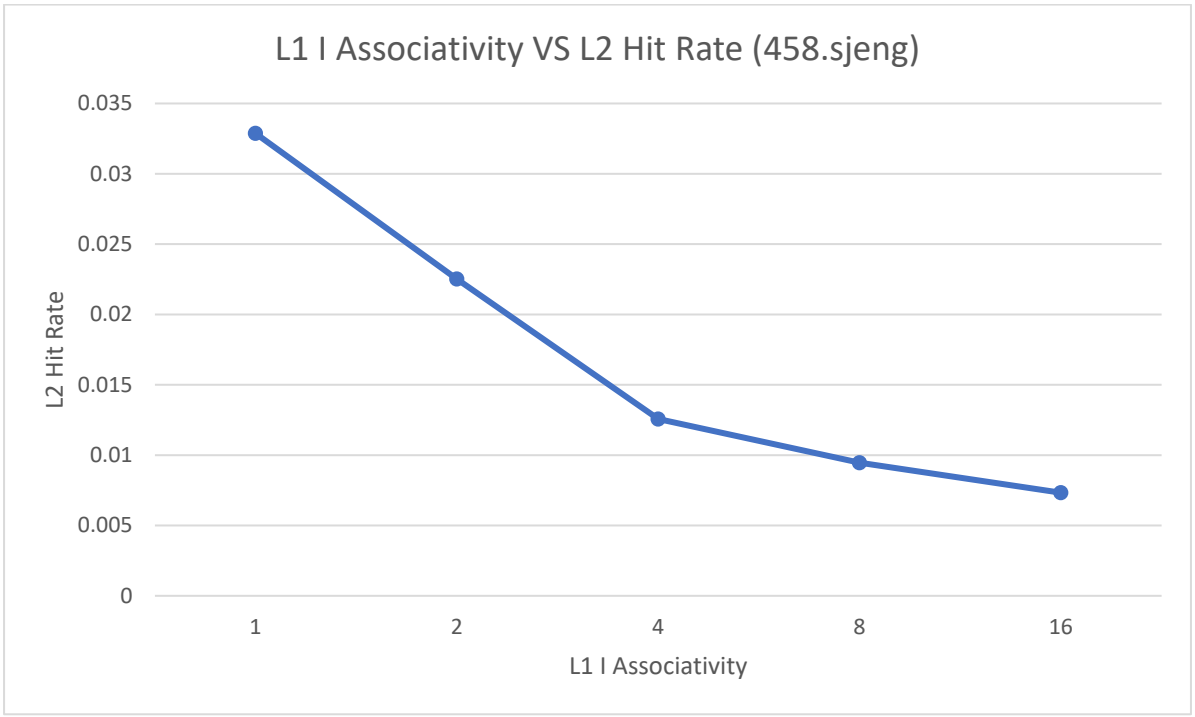
Analysis based on changes in L1I Associativity:

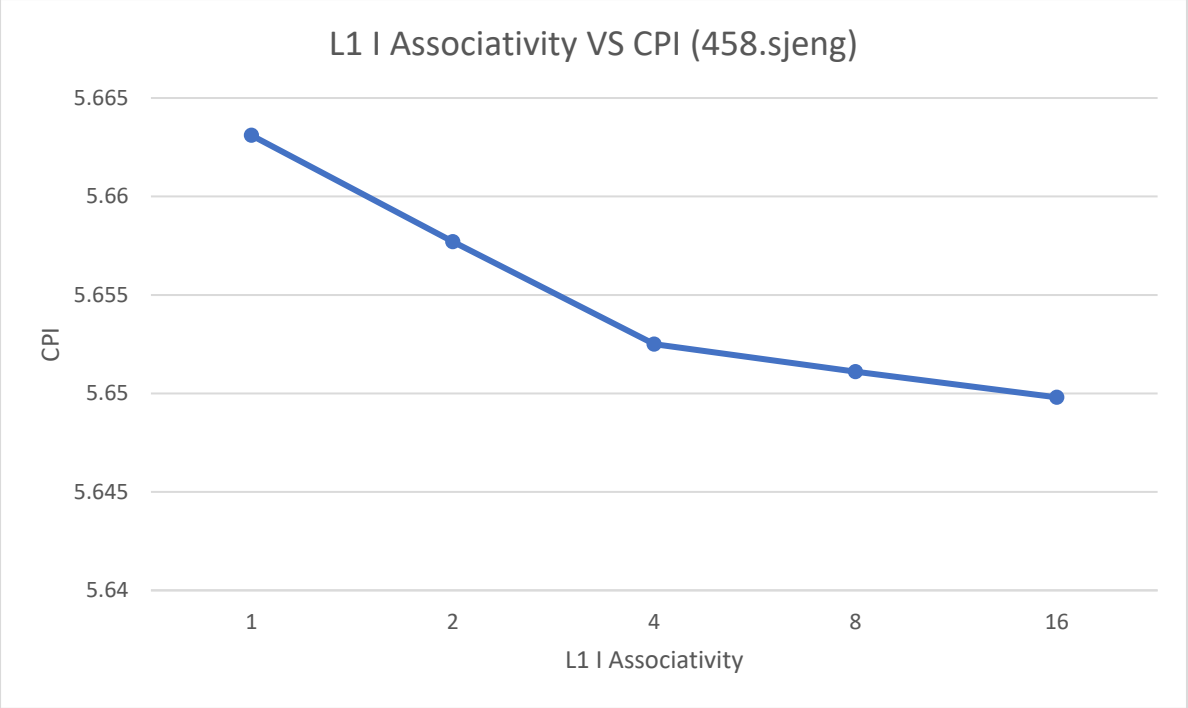
- Default values of Parameters: L1D Size: 128KB, L1I Cache Size:128KB, L2 Cache Size:1MB, L1D Associativity:2, L2 Associativity:1, Block Size:64B.
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L1I Associativity	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
1	0.902429	0.998054	0.032872	5.6631
2	0.902429	0.998707	0.022516	5.6577
4	0.902429	0.999322	0.012567	5.6525
8	0.902429	0.999511	0.009462	5.6511
16	0.902429	0.999643	0.00733	5.6498




L1 I Associativity VS L1I Hit Rate (458.sjeng)
1
0.9995
0.999
0.9985
0.998
0.9975
0.997
L1I Hit Rate
1
2
4
8
16
L1 I Associativity


L1 I Associativity VS L2 Hit Rate (458.sjeng)
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
L2 Hit Rate
1
2
4
8
16
L1 I Associativity



Varying L2 Associativity:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=2 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=4 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=8 --cacheline_size=64

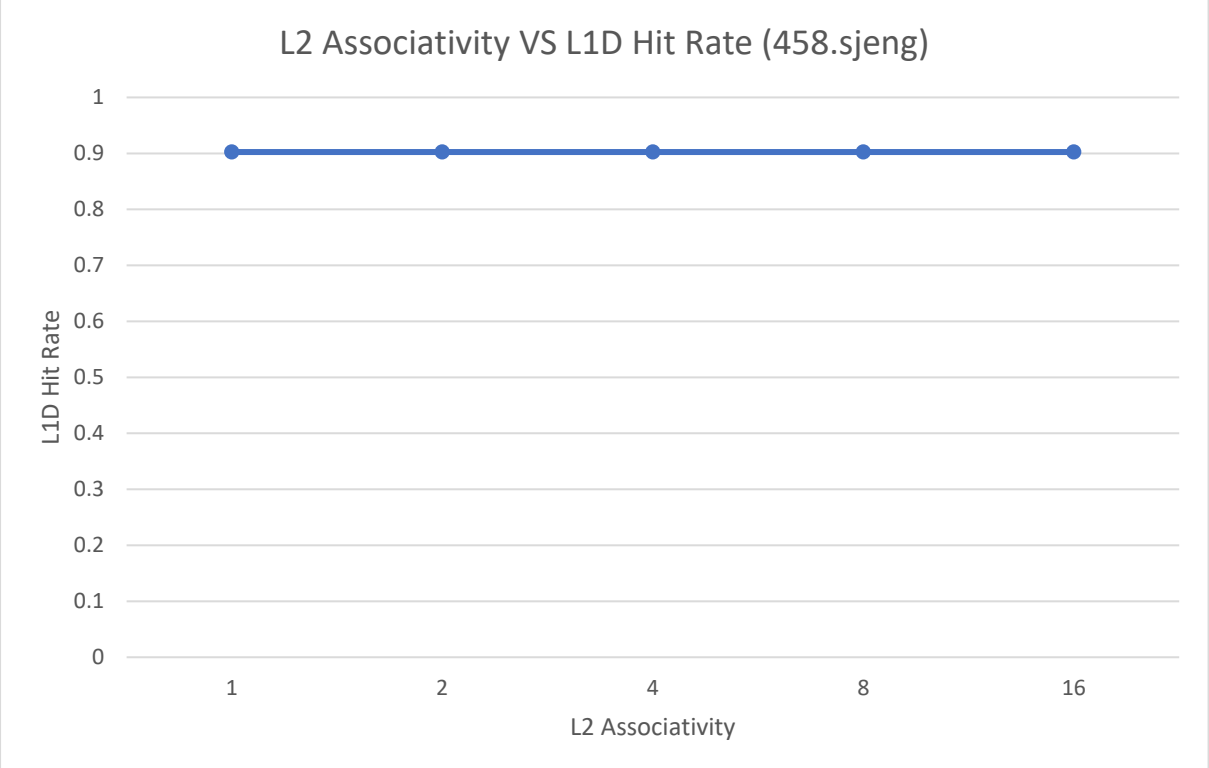
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --

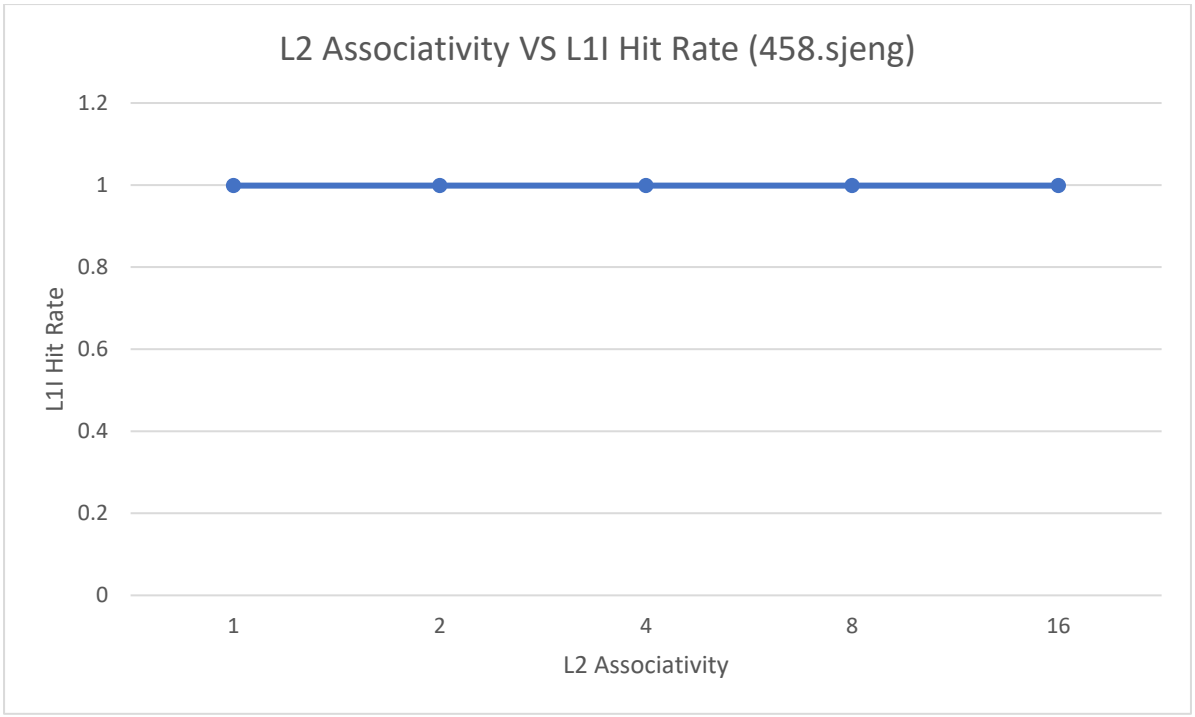
cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=16 --cacheline_size=64

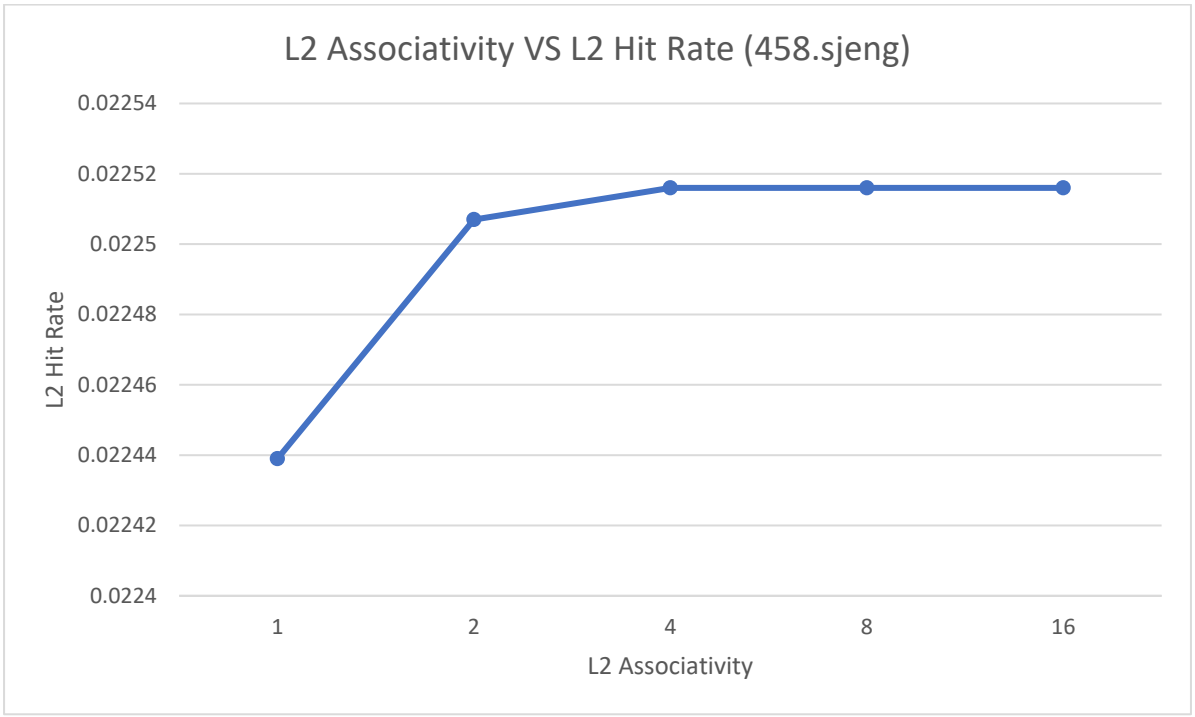
Analysis based on changes in L2 Associativity:

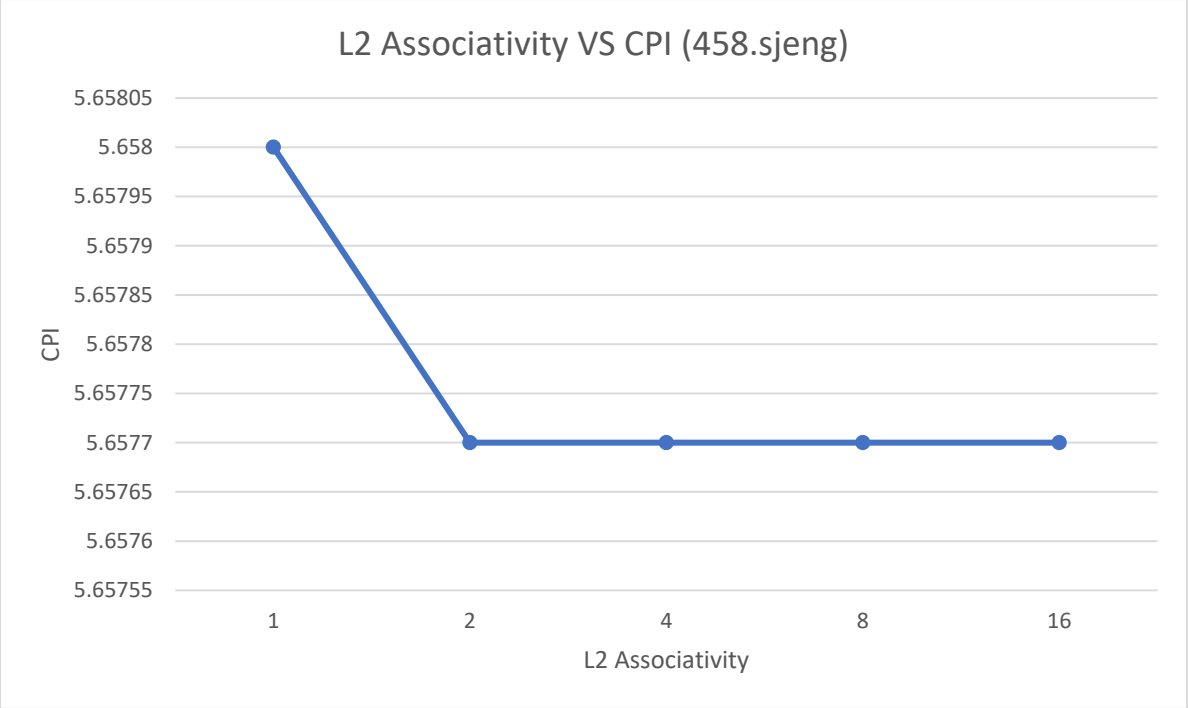
- Default values of Parameters: L1D Size: 128KB, L1I Cache Size:128KB, L2 Cache Size:1MB, L1D Associativity:2, L1I Associativity:2, Block Size:64B.
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

L2 Associativity	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
1	0.902429	0.998707	0.022439	5.658
2	0.902429	0.998707	0.022507	5.6577
4	0.902429	0.998707	0.022516	5.6577
8	0.902429	0.998707	0.022516	5.6577
16	0.902429	0.998707	0.022516	5.6577




L2 Associativity VS L1I Hit Rate (458.sjeng)
1.2
1
0.8
0.6
0.4
0.2
0
L1I Hit Rate
1
2
4
8
16
L2 Associativity


L2 Associativity VS L2 Hit Rate (458.sjeng)
0.02254
0.02252
0.0225
0.02248
0.02246
0.02244
0.02242
0.0224
L2 Hit Rate
1
2
4
8
16
L2 Associativity



Varying Block Size:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=8

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=16

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=32

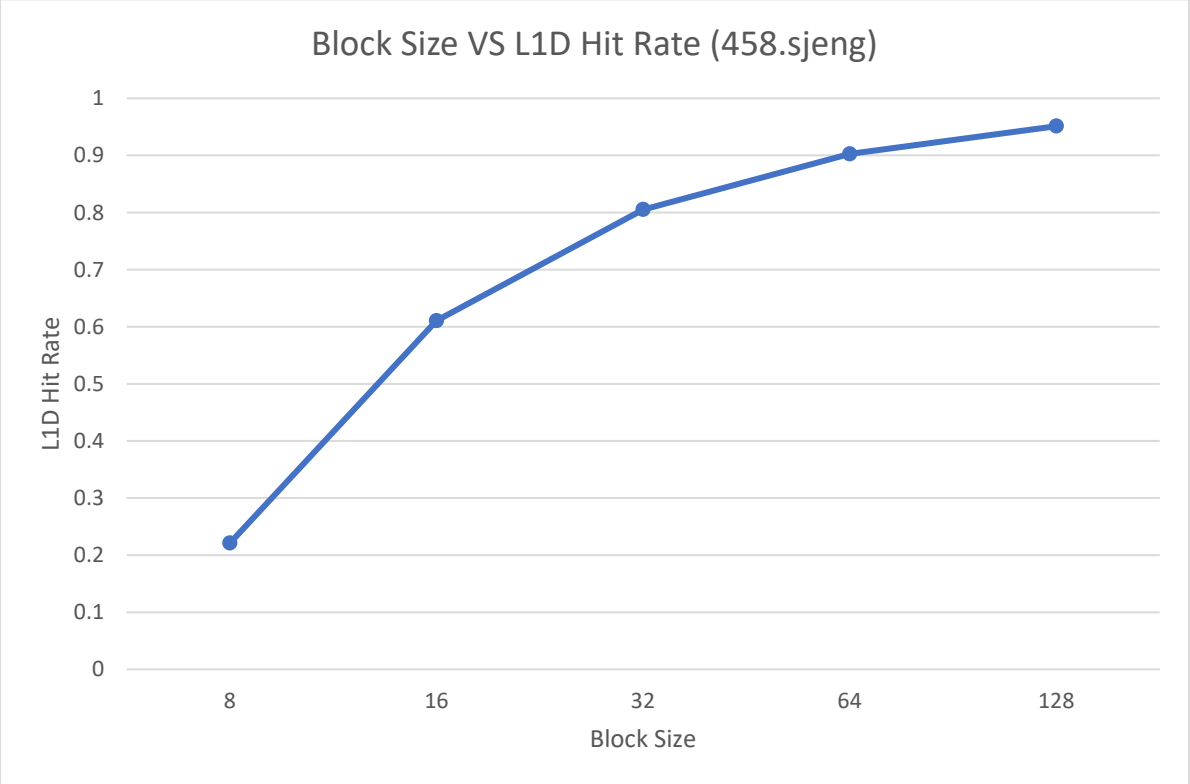
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

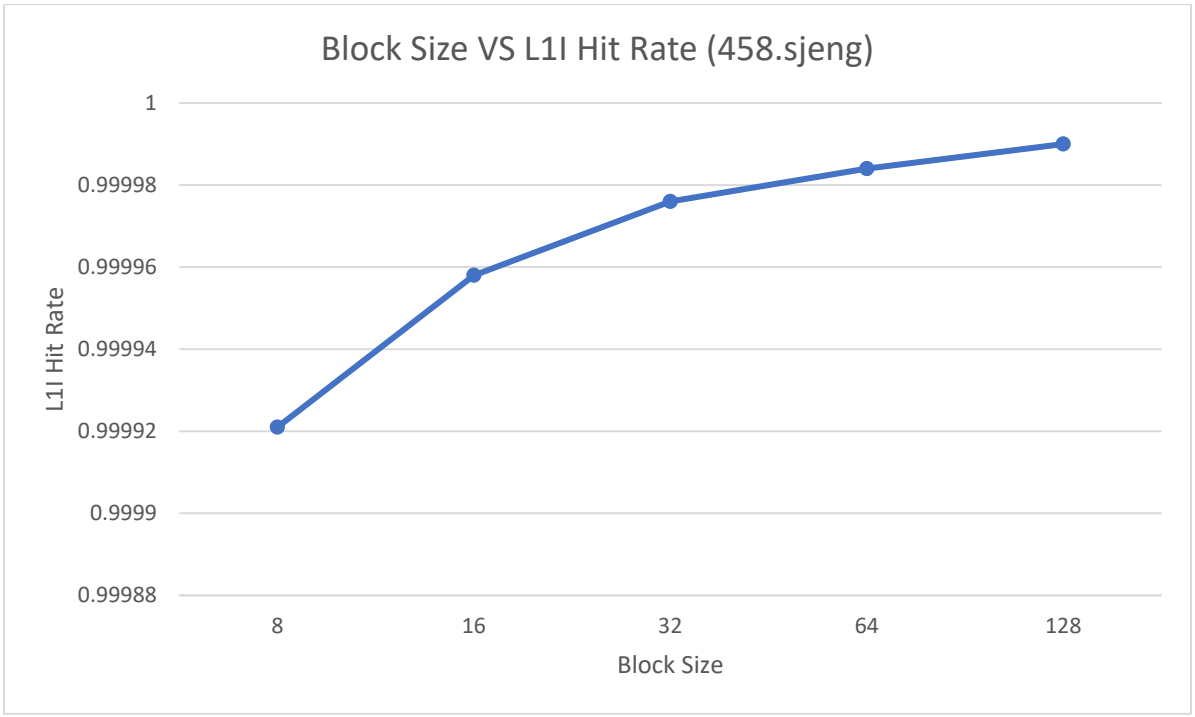
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=128kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=128

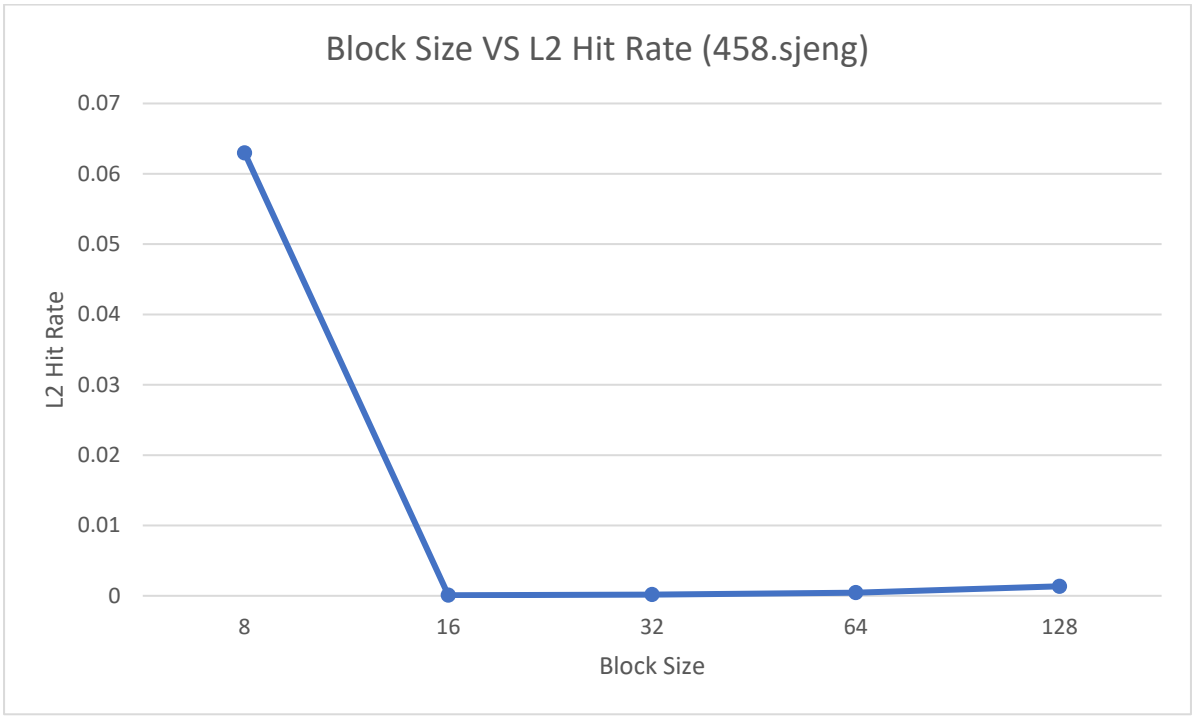
Analysis based on changes in Block Size:

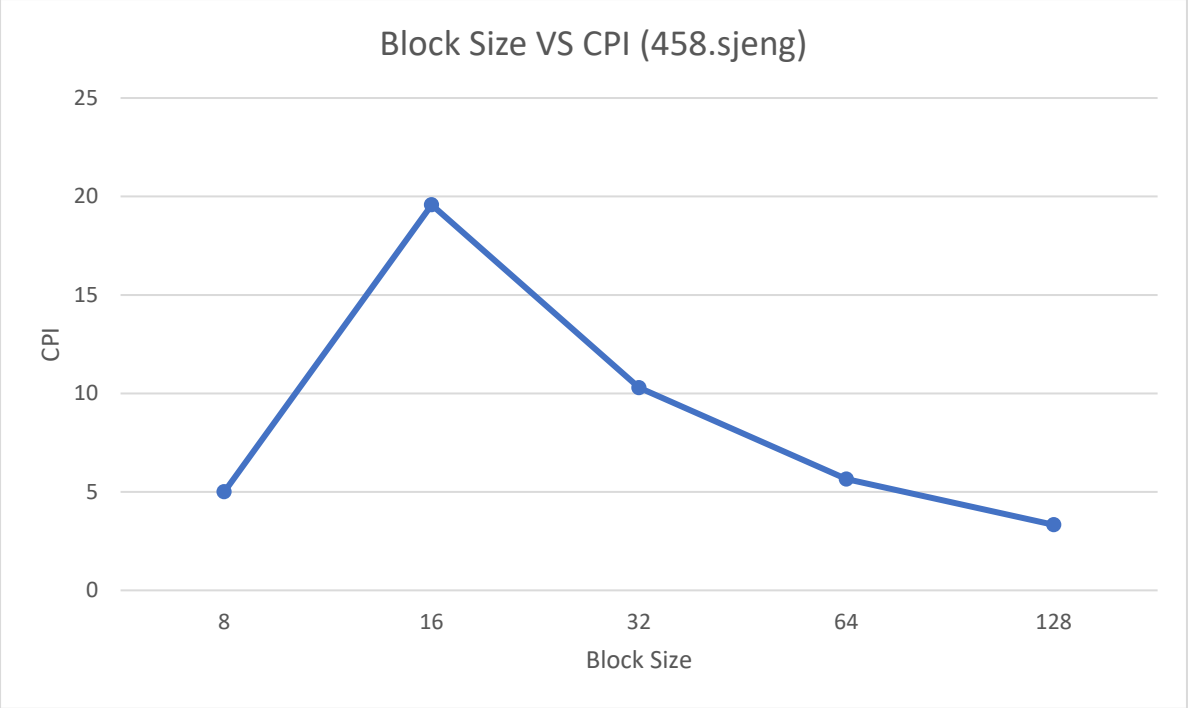
- Default values of Parameters: L1D Size: 128KB, L1I Cache Size:128KB, L2 Cache Size:1MB, L1D Associativity:2, L1I Associativity:2, L2 Associavity:1
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.
- Results:

Block size	L1D Hit rate	L1I Hit rate	L2 Hit rate	CPI
8	0.221085	0.999921	0.062964	5.00193712
16	0.610492	0.999958	0.000083	19.5751373
32	0.805194	0.999976	0.000177	10.2894330
64	0.902539	0.999984	0.000448	5.64664854
128	0.951191	0.99999	0.001353	3.32537064




Block Size VS L1I Hit Rate (458.sjeng)
1
0.99998
0.99996
0.99994
0.99992
0.9999
0.99988
L1I Hit Rate
8
16
32
64
128
Block Size


Block Size VS L2 Hit Rate (458.sjeng)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
L2 Hit Rate
8
16
32
64
128
Block Size



Varying Benchmark:

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/401.bzip2/src/benchmark -o ./benchmark/401.bzip2/data/input.program -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=32kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAPp2 ./configs/example/se.py -c ./benchmark/456.hmmer/src/benchmark -o ./benchmark/456.hmmer/data/bombesin.hmm.new -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=32kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

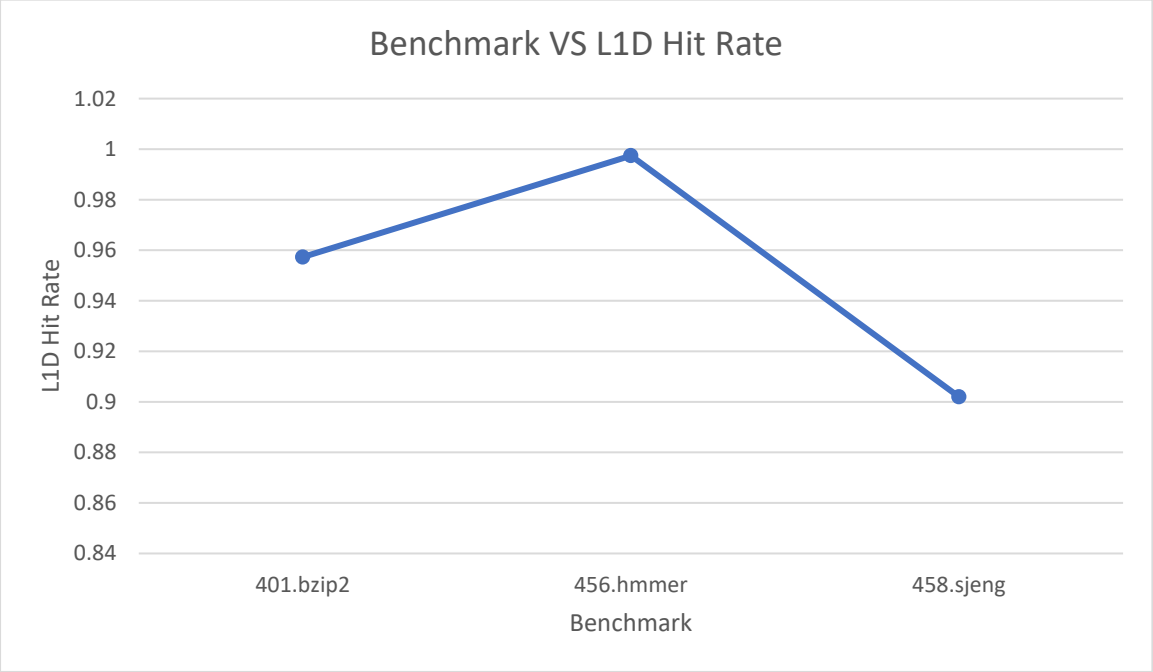
time ./build/X86/gem5.opt -d /home/csgrad/karanman/CAp2 ./configs/example/se.py -c ./benchmark/458.sjeng/src/benchmark -o ./benchmark/458.sjeng/data/test.txt -I 100000000 --cpu-type=TimingSimpleCPU --caches --l2cache --l1d_size=32kB --l1i_size=128kB --l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

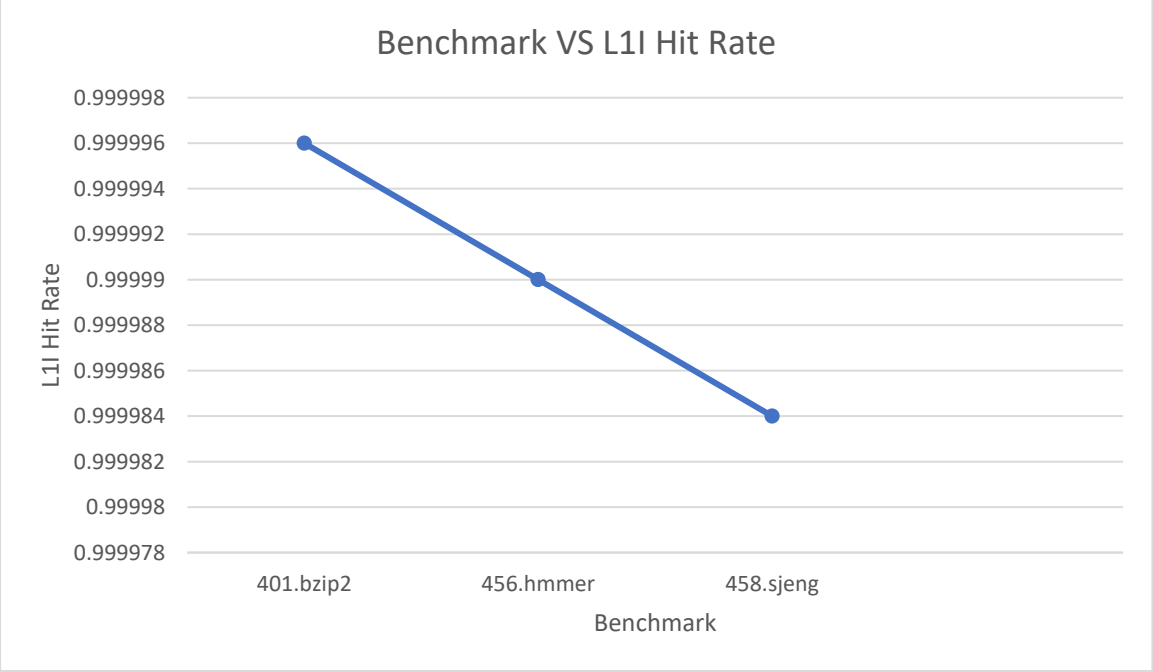
Analysis based on changes in Benchmark:

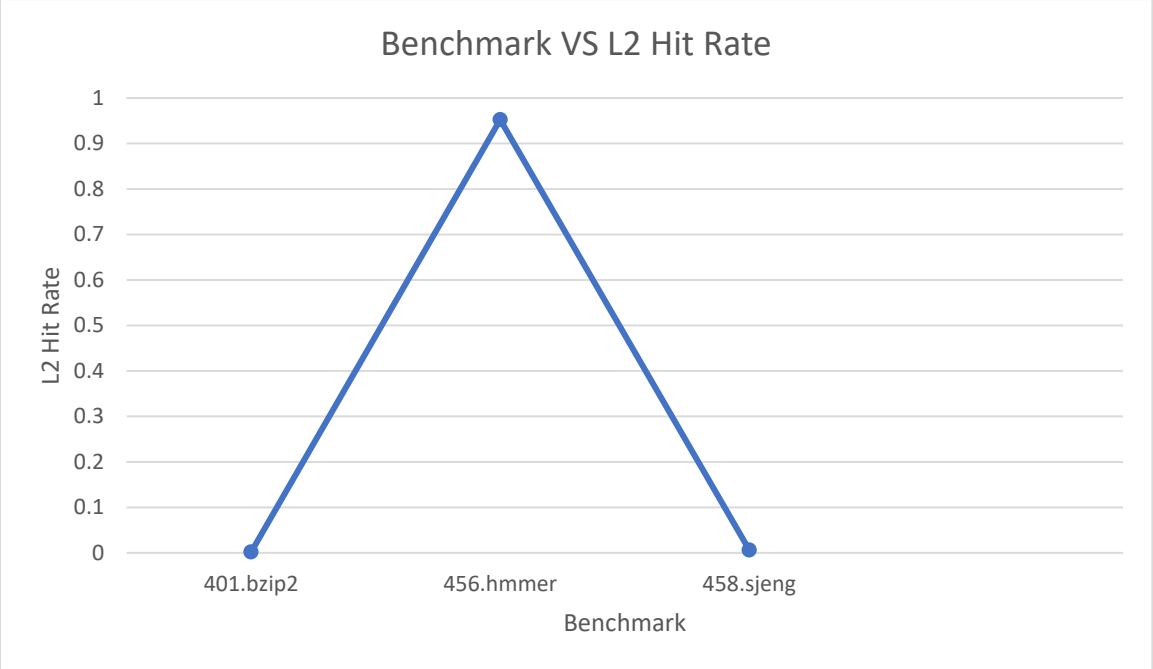
- Default values of Parameters: L1D Size: 128KB, L1I Cache Size:128KB, L2 Cache Size:1MB, L1D Associativity:2, L1I Associativity:2, L2 Associavity:1, Block Size:64B.
- In the graphs shown below the similar trends are seen as shown above. Please see through the analysis pointers discusses above and at the end of report.

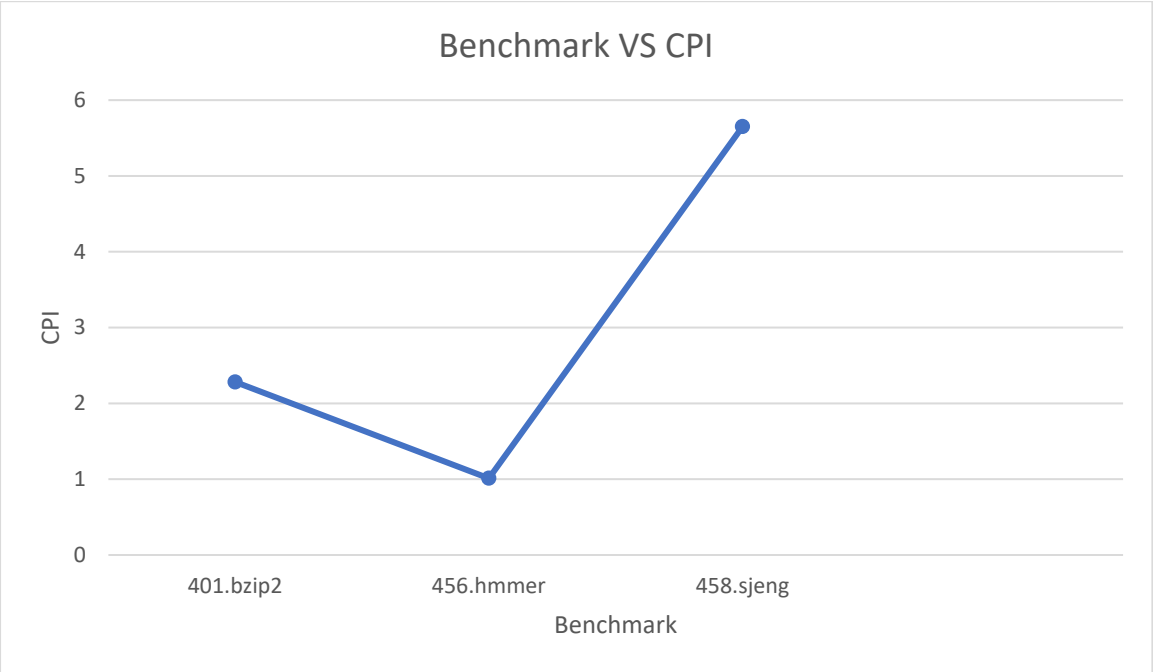
- Results:

Benchmark	L1 D Hit rate	L1 I Hit rate	L2 Hit rate	CPI
401.bzip2	0.957281	0.999996	0.001898	2.28182568
456.hmmer	0.997408	0.99999	0.95227	1.01120492
458.sjeng	0.901985	0.999984	0.006014	5.6498437









Final Analysis and Inferences:

- By carefully observing all the graphs we can make some general inferences that we observed by changing a single parameter and how it affects hit rates and CPI.
- As the size of the cache increases the miss rate decreases and hit rate increases because as the cache size is large there is a less chance of conflict. The same has been reflected in the values we have got above by running the commands.
- For example, for every benchmark when the size of L1I cache is increased, L1I hit rate of L1I increases. Same is the case with L1D cache and L2 cache.

- The same was the case with associativity of a cache. Mostly, it was seen that as the associativity increases the hit rate increases.
- For example, when the L1I associativity is kept higher, the hit rate of L1I increases. The same was seen with L1D and L2 cache.
- This is because, as we increase associativity, each set has more block so there is less chance of conflict between 2 addresses which both belong to the same set. However, change in associativity and size of L2 cache does not effect hit and miss rate of L1D and L1I cache.
- As for the CPI we know from the formula that it depends on the value of miss_num. So miss rate and CPI are proportional. So any change in miss rate will affect similar change in CPI.
- The graphs of all the hit rates and CPI according to the values of all the parameter for all 3 benchmark were shown above.