CENG311 - Fall 2021
PROGRAMMING ASSIGNMENT 4

mul.c

```
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_8_32_1.out -c 8 -b 32 -a1 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_8_32_2.out -c 8 -b 32 -a 2 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_8_32_4.out -c 8 -b 32 -a 4 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_16_16_1.out -c 16 -b 16 -a 1 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_16_32_1.out -c 16 -b 32 -a 1 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_16_64_1.out -c 16 -b 64 -a 1 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_16_16_2.out -c 16 -b 16 -a 2 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_16_32_2.out -c 16 -b 32 -a 2 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_16_64_2.out -c 16 -b 64 -a 2 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_16_16_4.out -c 16 -b 16 -a 4 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_16_32_4.out -c 16 -b 32 -a 4 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_16_64_4.out -c 16 -b 64 -a 4 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_32_32_1.out -c 32 -b 32 -a 1 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_32_32_2.out -c 32 -b 32 -a 2 -- ./mul1
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_32_32_4.out -c 32 -b 32 -a 4 -- ./mul1
```

mul_block.c

```
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_8_32_1.out2 -c 8 -b 32 -a 1 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_8_32_2.out2 -c 8 -b 32 -a 2 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_8_32_4.out2 -c 8 -b 32 -a 4 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_16_16_1.out2 -c 16 -b 16 -a 1 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_16_32_1.out2 -c 16 -b 32 -a 1 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_16_64_1.out2 -c 16 -b 64 -a 1 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_16_16_2.out2 -c 16 -b 16 -a 2 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_16_32_2.out2 -c 16 -b 32 -a 2 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_16_64_2.out2 -c 16 -b 64 -a 2 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_16_16_4.out2 -c 16 -b 16 -a 4 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_16_32_4.out2 -c 16 -b 32 -a 4 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_16_64_4.out2 -c 16 -b 64 -a 4 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_32_32_1.out2 -c 32 -b 32 -a 1 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_32_32_2.out2 -c 32 -b 32 -a 2 -- ./mul2
./pin -t source/tools/Memory/obj-intel64/dcache.so -o out2put_32_32_4.out2 -c 32 -b 32 -a 4 -- ./mul2
```

mul_block.c

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13033975325   91.98%
# Load-Misses:         1136797730    8.02%
# Load-Accesses:      14170773055  100.00%
#
# Store-Hits:          2038561501   99.77%
# Store-Misses:           4760112    0.23%
# Store-Accesses:      2043321613  100.00%
#
# Total-Hits:         15072536826   92.96%
# Total-Misses:        1141557842    7.04%
# Total-Accesses:     16214094668  100.00%
```

Cache size = 8
Block size = 32
Associativity = 1

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13503591221   95.29%
# Load-Misses:          667181834    4.71%
# Load-Accesses:      14170773055  100.00%
#
# Store-Hits:          2039148441   99.80%
# Store-Misses:           4173173    0.20%
# Store-Accesses:      2043321614  100.00%
#
# Total-Hits:         15542739662   95.86%
# Total-Misses:         671355007    4.14%
# Total-Accesses:     16214094669  100.00%
```

Cache size = 8
Block size = 32
Associativity = 2

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13830188772   97.60%
# Load-Misses:          340584338    2.40%
# Load-Accesses:      14170773110  100.00%
#
# Store-Hits:          2042060897   99.94%
# Store-Misses:           1260754    0.06%
# Store-Accesses:      2043321651  100.00%
#
# Total-Hits:         15872249669   97.89%
# Total-Misses:         341845092    2.11%
# Total-Accesses:     16214094761  100.00%
```

Cache size = 8
Block size = 32
Associativity = 4

For the first three input values above, cache size and block are kept constant and cache associativity values have been increased. Cache associativity being 1 is directed map, cache associativity being 2 is 2-way associativity, and cache associativity being 4 is 4-way associativity. We observe that as the associativity increases, the miss rate decreases, and the hit rate increases. When associativity is 1, the cache indexes are increasing as 0, 1, 2 ... and we can load just one data in each index. In other words, when we want to load data from memory to cache, when the blocks in the cache are full, we lose the data in the first index of the cache to load the new data. This causes the miss rate to increase and the hit rate to decrease. When Associativity is 2, we have 2 blocks in our 0th index to load the data in the cache. In other words, when we want to load the other data from memory to the cache, since we can load 2 data in index 0, our hit count increases, and we reduce the miss rate. Likewise, when associativity is 4, the block of data that we can load in each index increases. For this reason, we are less likely to lose previous data. When we want to reload the same data, that data already exists in the cache, which causes our hit rate to increase and our miss rate to decrease. When the hit rate is high and the miss rate is low, we understand that our cache memory works with a better performance.

Cache size = 16

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13447086417   94.89%
# Load-Misses:          723686632    5.11%
# Load-Accesses:      14170773049  100.00%
#
# Store-Hits:          2039967853   99.84%
# Store-Misses:           3353755    0.16%
# Store-Accesses:      2043321608  100.00%
#
# Total-Hits:         15487054270   95.52%
# Total-Misses:         727040387    4.48%
# Total-Accesses:     16214094657  100.00%
```

Block size = 16
Associativity = 1

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13686747964   96.58%
# Load-Misses:          484025094    3.42%
# Load-Accesses:      14170773058  100.00%
#
# Store-Hits:          2039618128   99.82%
# Store-Misses:           3703488    0.18%
# Store-Accesses:      2043321616  100.00%
#
# Total-Hits:         15726366092   96.99%
# Total-Misses:         487728582    3.01%
# Total-Accesses:     16214094674  100.00%
```

Cache size = 16
Block size = 32
Associativity = 1

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13458884849   94.98%
# Load-Misses:          711888209    5.02%
# Load-Accesses:      14170773058  100.00%
#
# Store-Hits:          2038640251   99.77%
# Store-Misses:           4681365    0.23%
# Store-Accesses:      2043321616  100.00%
#
# Total-Hits:         15497525100   95.58%
# Total-Misses:         716569574    4.42%
# Total-Accesses:     16214094674  100.00%
```

Cache size = 16
Block size = 64
Associativity = 1

In the three inputs above, we observe that associativity and cache size are kept constant and block size is increased. We see that by increasing the block size, the miss rate decreases, and the hit rate increases. The reason of this, Increasing the block size reduces the miss rate since we're bringing words into the cache that will be cache hits on subsequent accesses.

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13588339880   95.89%
# Load-Misses:          582433169    4.11%
# Load-Accesses:      14170773049  100.00%
#
# Store-Hits:          2041866361   99.93%
# Store-Misses:           1455246    0.07%
# Store-Accesses:      2043321607  100.00%
#
# Total-Hits:         15630206241   96.40%
# Total-Misses:         583888415    3.60%
# Total-Accesses:     16214094656  100.00%
```

Cache size = 16
Block size = 16
Associativity = 2

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13708207152   96.74%
# Load-Misses:          462565903    3.26%
# Load-Accesses:      14170773055  100.00%
#
# Store-Hits:          2041248044   99.90%
# Store-Misses:           2073569    0.10%
# Store-Accesses:      2043321613  100.00%
#
# Total-Hits:         15749455196   97.13%
# Total-Misses:         464639472    2.87%
# Total-Accesses:     16214094668  100.00%
```

Cache size = 16
Block size = 32
Associativity = 2

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13509637116   95.33%
# Load-Misses:          661135939    4.67%
# Load-Accesses:      14170773055  100.00%
#
# Store-Hits:          2040310816   99.85%
# Store-Misses:           3010798    0.15%
# Store-Accesses:      2043321614  100.00%
#
# Total-Hits:         15549947932   95.90%
# Total-Misses:         664146737    4.10%
# Total-Accesses:     16214094669  100.00%
```

Cache size = 16
Block size = 64
Associativity = 2

In the three inputs above, we observe that associativity and cache size are kept constant and block size is increased. We see that by increasing the block size, the miss rate decreases, and the hit rate increases. The reason of this, Increasing the block size reduces the miss rate since we're bringing words into the cache that will be cache hits on subsequent accesses. In addition, according to our previous analysing, we observe that with the increase in associativity, the miss rate decreases.

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13605766869   96.01%
# Load-Misses:          565006238    3.99%
# Load-Accesses:      14170773107  100.00%
#
# Store-Hits:          2042289061   99.95%
# Store-Misses:           1032587    0.05%
# Store-Accesses:      2043321648  100.00%
#
# Total-Hits:         15648055930   96.51%
# Total-Misses:         566038825    3.49%
# Total-Accesses:     16214094755  100.00%
```

Cache size = 16
Block size = 16
Associativity = 4

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13879450693   97.94%
# Load-Misses:          291322362    2.06%
# Load-Accesses:      14170773055  100.00%
#
# Store-Hits:          2042712423   99.97%
# Store-Misses:           609190    0.03%
# Store-Accesses:      2043321613  100.00%
#
# Total-Hits:         15922163116   98.20%
# Total-Misses:         291931552    1.80%
# Total-Accesses:     16214094668  100.00%
```

Cache size = 16
Block size = 32
Associativity = 4

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13965354740   98.55%
# Load-Misses:          205418318    1.45%
# Load-Accesses:      14170773058  100.00%
#
# Store-Hits:          2042168751   99.94%
# Store-Misses:           1152865    0.06%
# Store-Accesses:      2043321616  100.00%
#
# Total-Hits:         16007523491   98.73%
# Total-Misses:         206571183    1.27%
# Total-Accesses:     16214094674  100.00%
```

Cache size = 16
Block size = 64
Associativity = 4

In the three inputs above, we observe that associativity and cache size are kept constant and block size is increased. We see that by increasing the block size, the miss rate decreases, and the hit rate increases. The reason of this, Increasing the block size reduces the miss rate since we're bringing words into the cache that will be cache hits on subsequent accesses. In addition, according to our previous analysing, we observe that with the increase in associativity, the miss rate decreases.

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13802418752   97.40%
# Load-Misses:          368354306    2.60%
# Load-Accesses:      14170773058  100.00%
#
# Store-Hits:          2041732968   99.92%
# Store-Misses:           1588648    0.08%
# Store-Accesses:      2043321616  100.00%
#
# Total-Hits:         15844151720   97.72%
# Total-Misses:         369942954    2.28%
# Total-Accesses:     16214094674  100.00%
```

Cache size = 32
Block size = 32
Associativity = 1

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13890439533   98.02%
# Load-Misses:          280333516    1.98%
# Load-Accesses:      14170773049  100.00%
#
# Store-Hits:          2042509659   99.96%
# Store-Misses:            811949    0.04%
# Store-Accesses:      2043321608  100.00%
#
# Total-Hits:         15932949192   98.27%
# Total-Misses:         281145465    1.73%
# Total-Accesses:     16214094657  100.00%
```

Cache size = 32
Block size = 32
Associativity = 2

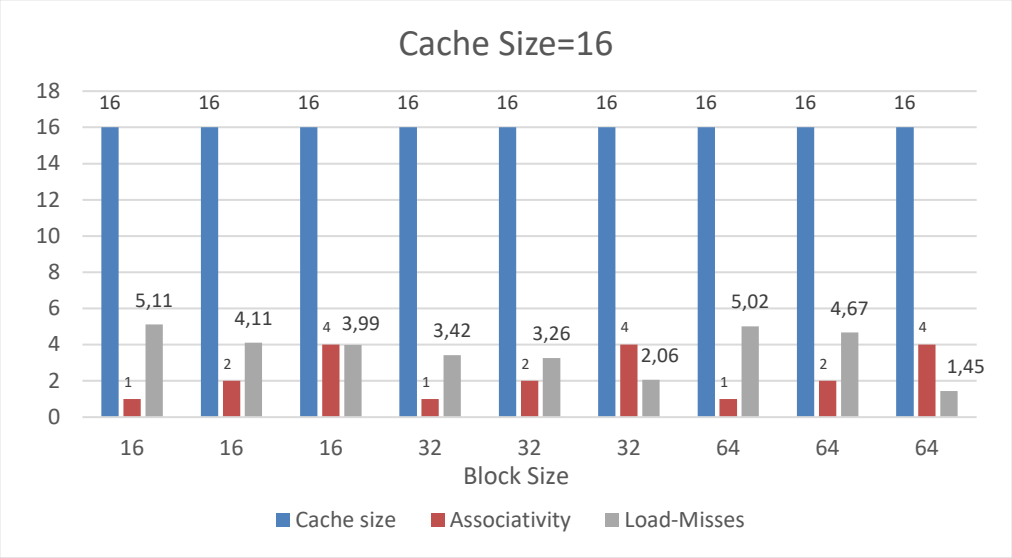
```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          13896900164   98.07%
# Load-Misses:          273872888    1.93%
# Load-Accesses:      14170773052  100.00%
#
# Store-Hits:          2042799196   99.97%
# Store-Misses:            522416    0.03%
# Store-Accesses:      2043321612  100.00%
#
# Total-Hits:         15939699360   98.31%
# Total-Misses:         274395304    1.69%
# Total-Accesses:     16214094664  100.00%
```

Cache size = 32
Block size = 32
Associativity = 4

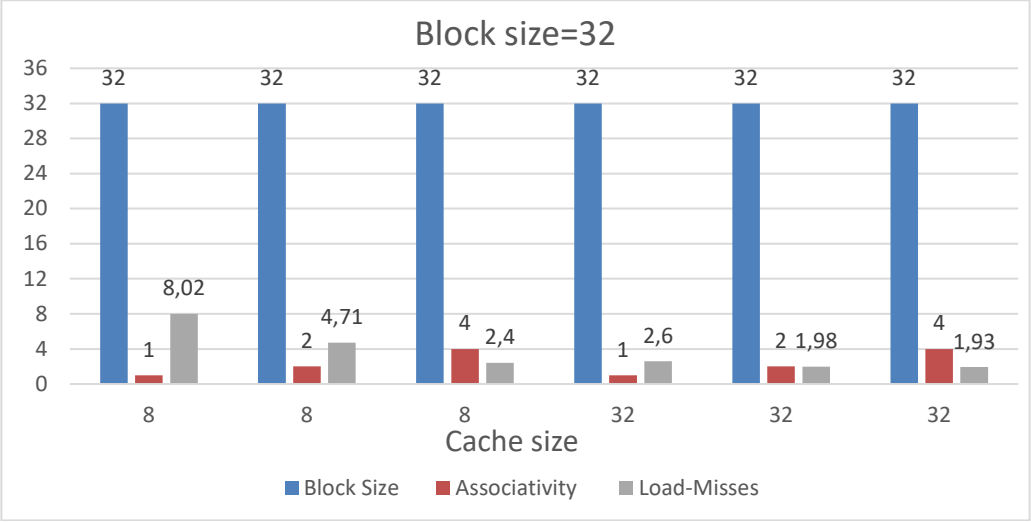
For the first three input values above, cache size and block are kept constant and cache associativity values have been increased. In addition to the first analysing, the cache size is increased. Thus, it has caused decreasing miss rate with the associativity.

Graphs for mul_block.c

Block Size	Cache size	Asso-ciativity	Load-Misses
16	16	1	5,11
16	16	2	4,11
16	16	4	3,99
32	16	1	3,42
32	16	2	3,26
32	16	4	2,06
64	16	1	5,02
64	16	2	4,67
64	16	4	1,45



Block Size	Cache size	Asso-ciativity	Load-Misses
32	8	1	8,02
32	8	2	4,71
32	8	4	2,4
32	32	1	2,6
32	32	2	1,98
32	32	4	1,93



As you see for Block size=32, load misses are decreasing when associativity is increase for cache size is 8. And the same as a cache size 8, for cache size 32, load misses are decreasing when associativity is increase.

Well, **we can understand if we look the two graph above, when cache size is increasing, load miss is decreasing**(when Block Size is the same). **Because bigger cache size can exploit temporal locality better (but not always.)**

mul.c

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11450919270   87.96%
# Load-Misses:         1567243449   12.04%
# Load-Accesses:      13018162719  100.00%
#
# Store-Hits:          2005581679   99.92%
# Store-Misses:           1531709    0.08%
# Store-Accesses:      2007113388  100.00%
#
# Total-Hits:         13456500949   89.56%
# Total-Misses:        1568775158   10.44%
# Total-Accesses:     15025276107  100.00%
```

Cache size = 8
Block size = 32
Associativity = 1

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11496755418   88.31%
# Load-Misses:         1521407310   11.69%
# Load-Accesses:      13018162728  100.00%
#
# Store-Hits:          2005599109   99.92%
# Store-Misses:           1514287    0.08%
# Store-Accesses:      2007113396  100.00%
#
# Total-Hits:         13502354527   89.86%
# Total-Misses:        1522921597   10.14%
# Total-Accesses:     15025276124  100.00%
```

Cache size = 8
Block size = 32
Associativity = 2

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11497896209   88.32%
# Load-Misses:         1520266510   11.68%
# Load-Accesses:      13018162719  100.00%
#
# Store-Hits:          2005599217   99.92%
# Store-Misses:           1514170    0.08%
# Store-Accesses:      2007113387  100.00%
#
# Total-Hits:         13503495426   89.87%
# Total-Misses:        1521780680   10.13%
# Total-Accesses:     15025276106  100.00%
```

Cache size = 8
Block size = 32
Associativity = 4

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11207587433   86.09%
# Load-Misses:         1810575289   13.91%
# Load-Accesses:      13018162722  100.00%
#
# Store-Hits:          2005089583   99.90%
# Store-Misses:           2023807    0.10%
# Store-Accesses:      2007113390  100.00%
#
# Total-Hits:         13212677016   87.94%
# Total-Misses:        1812599096   12.06%
# Total-Accesses:     15025276112  100.00%
```

Cache size = 16
Block size = 16
Associativity = 1

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11502273549   88.36%
# Load-Misses:         1515889179   11.64%
# Load-Accesses:      13018162728  100.00%
#
# Store-Hits:          2005591471   99.92%
# Store-Misses:           1521925    0.08%
# Store-Accesses:      2007113396  100.00%
#
# Total-Hits:         13507865020   89.90%
# Total-Misses:        1517411104   10.10%
# Total-Accesses:     15025276124  100.00%
```

Cache size = 16
Block size = 32
Associativity = 1

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11728156292   90.09%
# Load-Misses:         1290006418    9.91%
# Load-Accesses:      13018162710  100.00%
#
# Store-Hits:          2005834616   99.94%
# Store-Misses:           1278775    0.06%
# Store-Accesses:      2007113391  100.00%
#
# Total-Hits:         13733990908   91.41%
# Total-Misses:        1291285193    8.59%
# Total-Accesses:     15025276101  100.00%
```

Cache size = 16
Block size = 64
Associativity = 1

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11013116713   84.60%
# Load-Misses:         2005046015   15.40%
# Load-Accesses:      13018162728  100.00%
#
# Store-Hits:          2005093176   99.90%
# Store-Misses:           2020220    0.10%
# Store-Accesses:      2007113396  100.00%
#
# Total-Hits:         13018209889   86.64%
# Total-Misses:        2007066235   13.36%
# Total-Accesses:     15025276124  100.00%
```

Cache size = 16
Block size = 16
Associativity = 2

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11506580913   88.39%
# Load-Misses:         1511581815   11.61%
# Load-Accesses:      13018162728  100.00%
#
# Store-Hits:          2005600850   99.92%
# Store-Misses:           1512546    0.08%
# Store-Accesses:      2007113396  100.00%
#
# Total-Hits:         13512181763   89.93%
# Total-Misses:        1513094361   10.07%
# Total-Accesses:     15025276124  100.00%
```

Cache size = 16
Block size = 32
Associativity = 2

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11749594440   90.26%
# Load-Misses:         1268568288    9.74%
# Load-Accesses:      13018162728  100.00%
#
# Store-Hits:          2005856297   99.94%
# Store-Misses:           1257099    0.06%
# Store-Accesses:      2007113396  100.00%
#
# Total-Hits:         13755450737   91.55%
# Total-Misses:        1269825387    8.45%
# Total-Accesses:     15025276124  100.00%
```

Cache size = 16
Block size = 64
Associativity = 2

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11008273341   84.56%
# Load-Misses:         2009889387   15.44%
# Load-Accesses:      13018162728  100.00%
#
# Store-Hits:          2005093100   99.90%
# Store-Misses:           2020296    0.10%
# Store-Accesses:      2007113396  100.00%
#
# Total-Hits:         13013366441   86.61%
# Total-Misses:        2011909683   13.39%
# Total-Accesses:     15025276124  100.00%
```

Cache size = 16
Block size = 16
Associativity = 4

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11506803789   88.39%
# Load-Misses:         1511358933   11.61%
# Load-Accesses:      13018162722  100.00%
#
# Store-Hits:          2005601160   99.92%
# Store-Misses:           1512230    0.08%
# Store-Accesses:      2007113390  100.00%
#
# Total-Hits:         13512404949   89.93%
# Total-Misses:        1512871163   10.07%
# Total-Accesses:     15025276112  100.00%
```

Cache size = 16
Block size = 32
Associativity = 4

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11755579834   90.30%
# Load-Misses:         1262582882    9.70%
# Load-Accesses:      13018162716  100.00%
#
# Store-Hits:          2005855442   99.94%
# Store-Misses:           1257942    0.06%
# Store-Accesses:      2007113384  100.00%
#
# Total-Hits:         13761435276   91.59%
# Total-Misses:        1263840824    8.41%
# Total-Accesses:     15025276100  100.00%
```

Cache size = 16
Block size = 64
Associativity = 4

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11864314921   91.14%
# Load-Misses:         1153847798    8.86%
# Load-Accesses:      13018162719  100.00%
#
# Store-Hits:          2005850727   99.94%
# Store-Misses:           1262661    0.06%
# Store-Accesses:      2007113388  100.00%
#
# Total-Hits:         13870165648   92.31%
# Total-Misses:        1155110459    7.69%
# Total-Accesses:     15025276107  100.00%
```

Cache size = 32
Block size = 32
Associativity = 1

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11767126417   90.39%
# Load-Misses:         1251036302    9.61%
# Load-Accesses:      13018162719  100.00%
#
# Store-Hits:          2005608580   99.93%
# Store-Misses:           1504807    0.07%
# Store-Accesses:      2007113387  100.00%
#
# Total-Hits:         13772734997   91.66%
# Total-Misses:        1252541109    8.34%
# Total-Accesses:     15025276106  100.00%
```

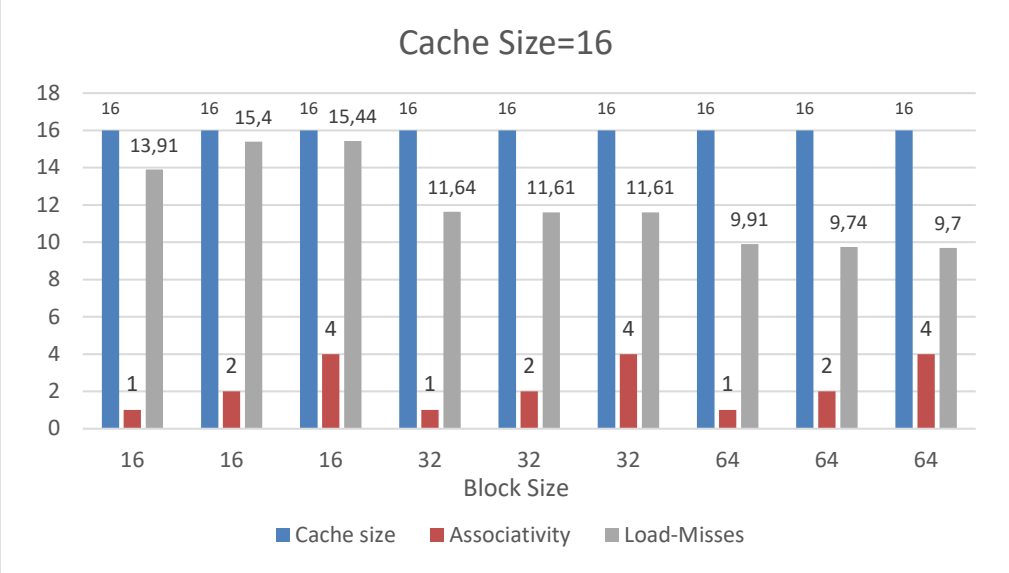
Cache size = 32
Block size = 32
Associativity = 2

```
PIN:MEMLATENCIES 1.0. 0x0
#
# DCACHE stats
#
# L1 Data Cache:
# Load-Hits:          11596489833   89.08%
# Load-Misses:         1421672886   10.92%
# Load-Accesses:      13018162719  100.00%
#
# Store-Hits:          2005609574   99.93%
# Store-Misses:           1503814    0.07%
# Store-Accesses:      2007113388  100.00%
#
# Total-Hits:         13602099407   90.53%
# Total-Misses:        1423176700    9.47%
# Total-Accesses:     15025276107  100.00%
```

Cache size = 32
Block size = 32
Associativity = 4

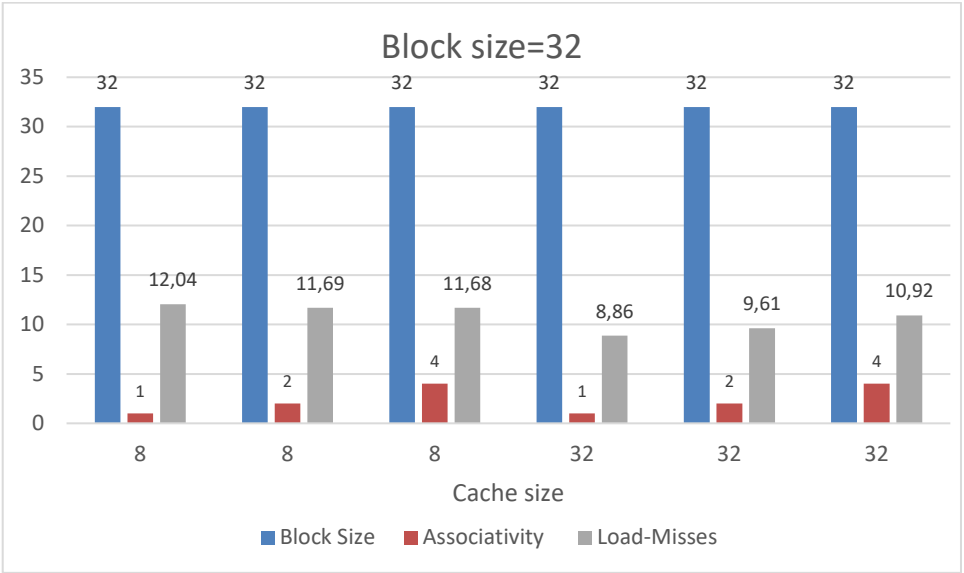
Graphs for mull.c

Block Size	Cache size	Asso-ciativity	Load-Misses
16	16	1	13,91
16	16	2	15,4
16	16	4	15,44
32	16	1	11,64
32	16	2	11,61
32	16	4	11,61
64	16	1	9,91
64	16	2	9,74
64	16	4	9,7



For 16 block size, miss rate is increased because of the array size is greater than block size. Thus, even the associativity is increased, miss rate is not decreased. However, other block sizes, the miss rate is decreased because of the array size is not bigger than block size and, the associativity is increased. Thus, when the block size is increased, the miss rate is decreased because of spatial locality.

Block Size	Cache size	Asso-ciativity	Load-Misses
32	8	1	12,04
32	8	2	11,69
32	8	4	11,68
32	32	1	8,86
32	32	2	9,61
32	32	4	10,92



High cache size does not always give positive results for miss rate. In this example, where the cache size is 32 although associativity has increased, the miss rate has also increased. Due to the size of the array in the process, when the cache was big for you, it had a negative effect on the miss rate and caused the miss rate to increase.

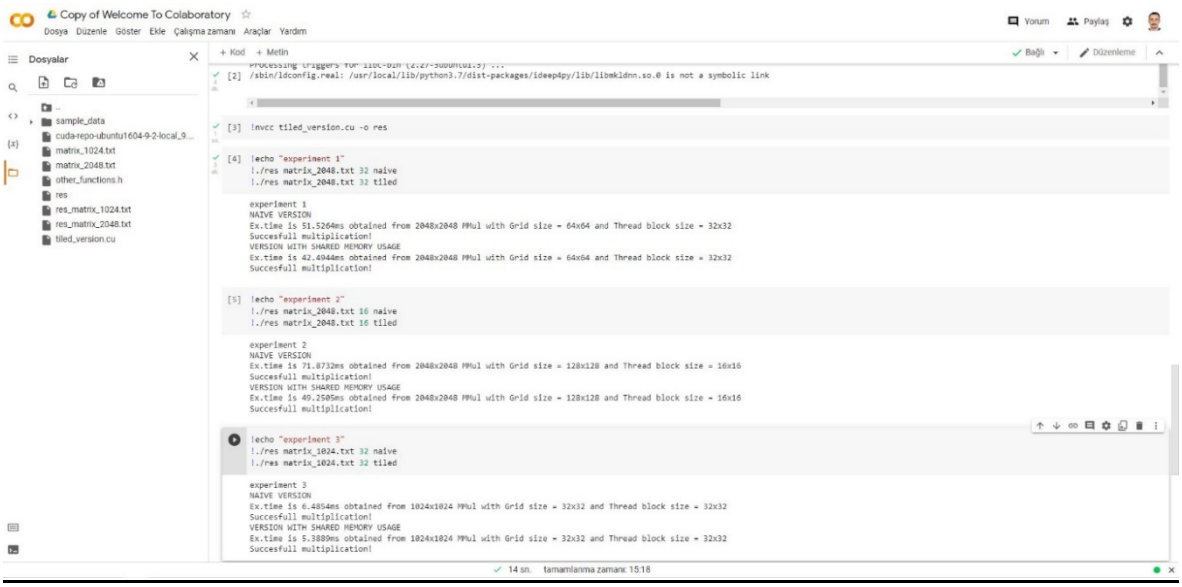
As I mentioned above, **we can understand if we look the two graphs above, when cache size is increasing, load miss is decreasing** (when Block Size is the same). **Because bigger cache size can exploit temporal locality better.**

L1 Size = [KB], Block size = [B], L1 Associativity

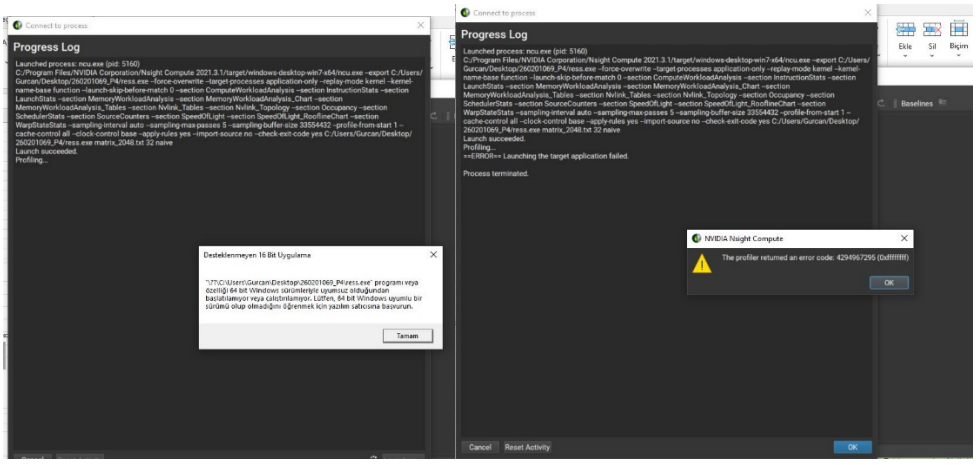
OPTIONAL PART

As we learned in the last lab lesson, I set the Run time to GPU in Google Collaborate. I have uploaded the necessary files. Then I ran the given commands in the cells in order.

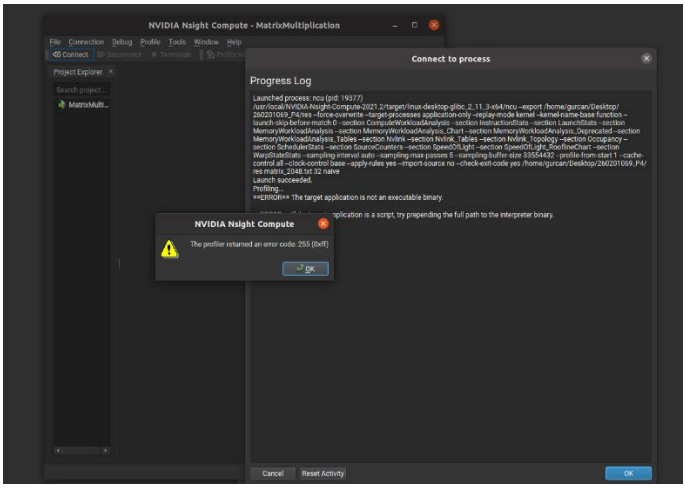
```
!apt-get --purge remove cuda nvidia* libnvidia-*
!dpkg -l | grep cuda- | awk '{print $2}' | xargs -n1 dpkg --purge
!apt-get remove cuda-*
!apt autoremove
!apt-get update

!wget https://developer.nvidia.com/compute/cuda/9.2/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64 -O cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb
!dpkg -i cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb
!apt-key add /var/cuda-repo-9-2-local/7fa2af80.pub
!apt-get update
!apt-get install cuda-9.2

!nvcc tiled_version.cu -o res

!echo "experiment 1"
!./res matrix_2048.txt 32 naive
!./res matrix_2048.txt 32 tiled

!echo "experiment 2"
!./res matrix_2048.txt 16 naive
!./res matrix_2048.txt 16 tiled

!echo "experiment 3"
!./res matrix_1024.txt 32 naive
!./res matrix_1024.txt 32 tiled

!echo "experiment 4"
!./res matrix_1024.txt 16 naive
!./res matrix_1024.txt 16 tiled
```



In Windows Operating system, I tried to run Nvdia Compute; I had this error;



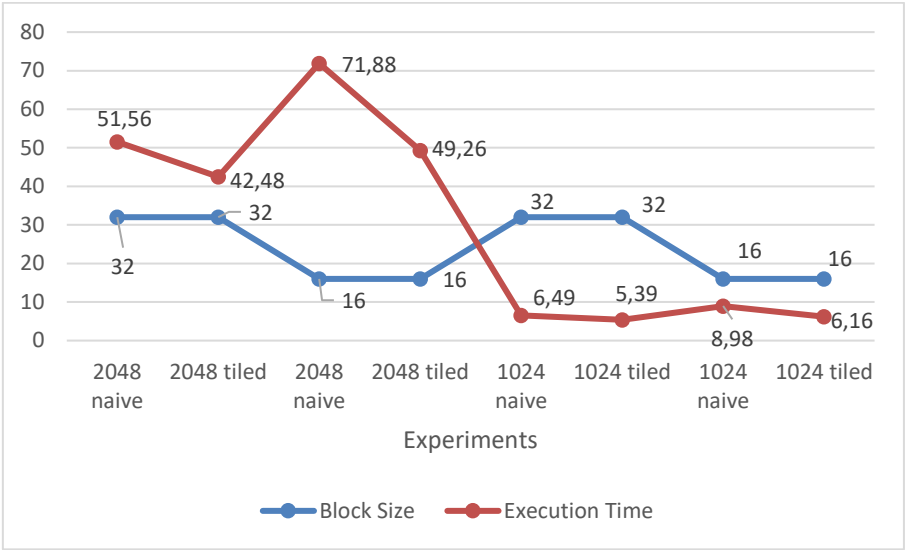
Then I tried to run Nvdia Nsight Compute Compute in linux operating system. But I had this error.



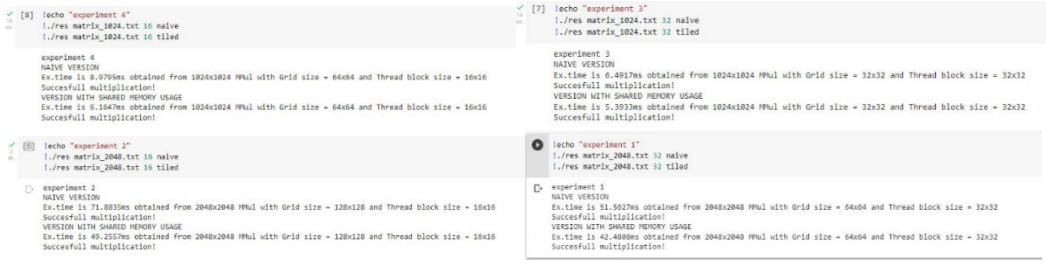
Then I decided to draw myself from teaching our lecture video and our textbook.

This is my result.

Experiments	Block Size	Execution Time
2048 naive	32	51,56
2048 tiled	32	42,48
2048 naive	16	71,88
2048 tiled	16	49,26
1024 naive	32	6,49
1024 tiled	32	5,39
1024 naive	16	8,98
1024 tiled	16	6,16



We understand if we look the graph, when block size is decrease, execution time is increasing.



REFERANCES

1. CENG311 Computer Architecture Cache Performance: Lecture PDF by Işıl ÖZ,(IZTECH, Fall 2021 07 January 2022)
2. Design of Digital Circuits - Lecture 23b: Memory Organization &Technology by Onur MUTLU (ETH Zürich, Spring 2018) https://www.youtube.com/watch?v=sweCA3836C0
3. Design of Digital Circuits - Lecture 25a: More Caches by Onur MUTLU by Onur MUTLU (ETH Zürich, Spring 2018) https://www.youtube.com/watch?v=kMUZKjaPNWo
4. Computer Organization and Design: The Hardware/Software Interface by Hennessy/Patterson, 5th Edition.