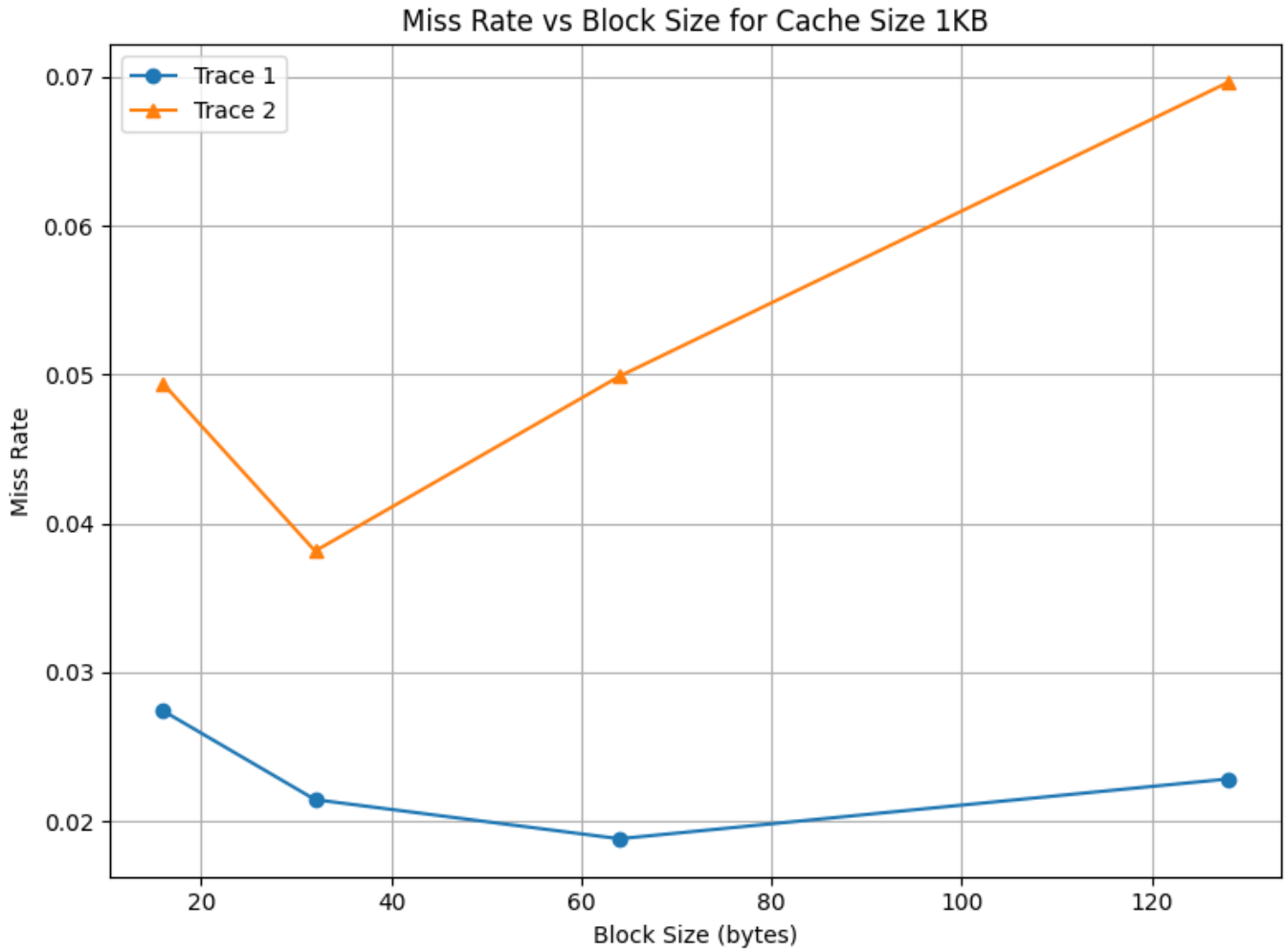
计算机组织与系统结构实习报告 Lab3.1

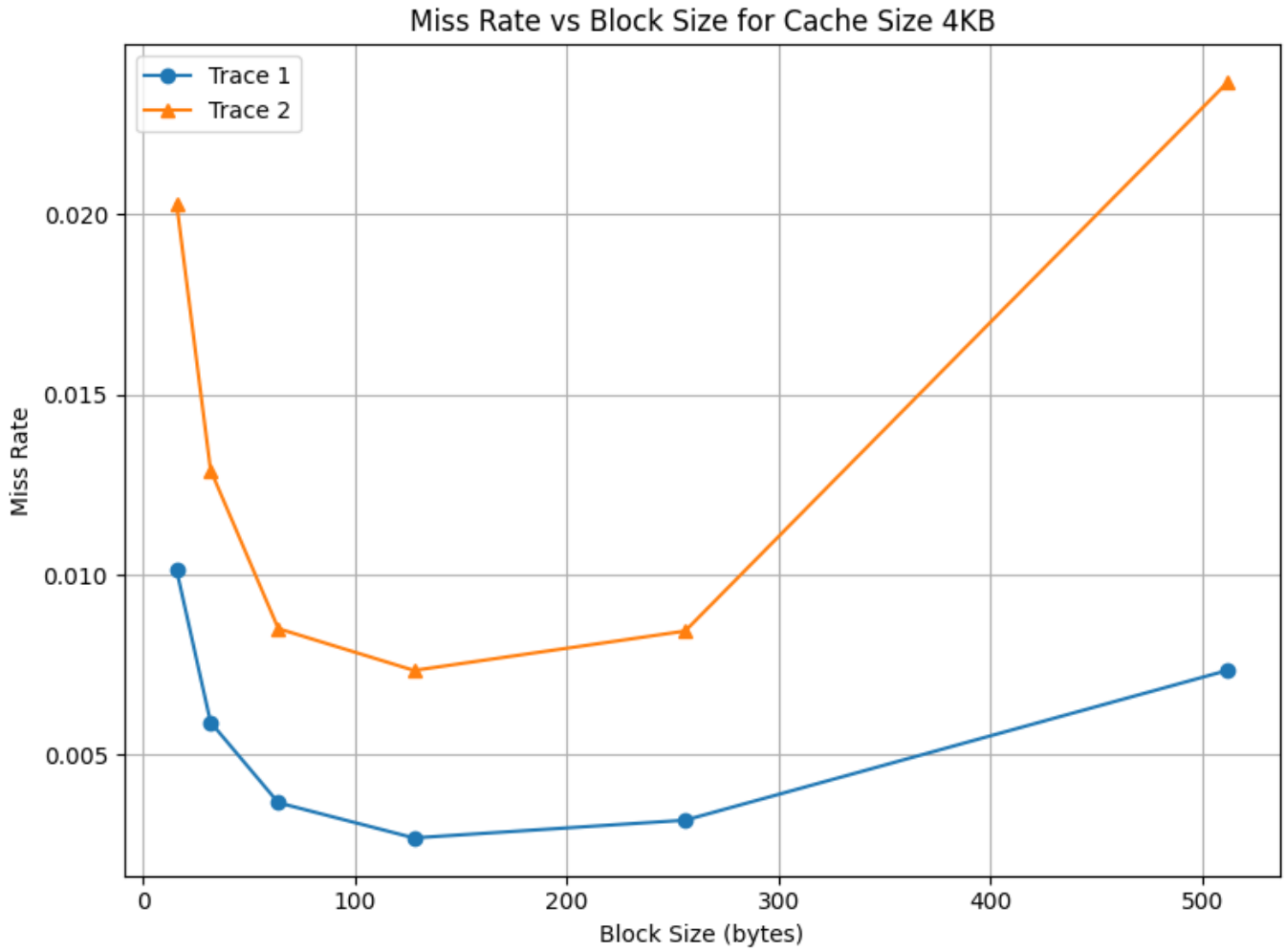
学号:2200013093 姓名:张佳豪 大班教师:陆俊林

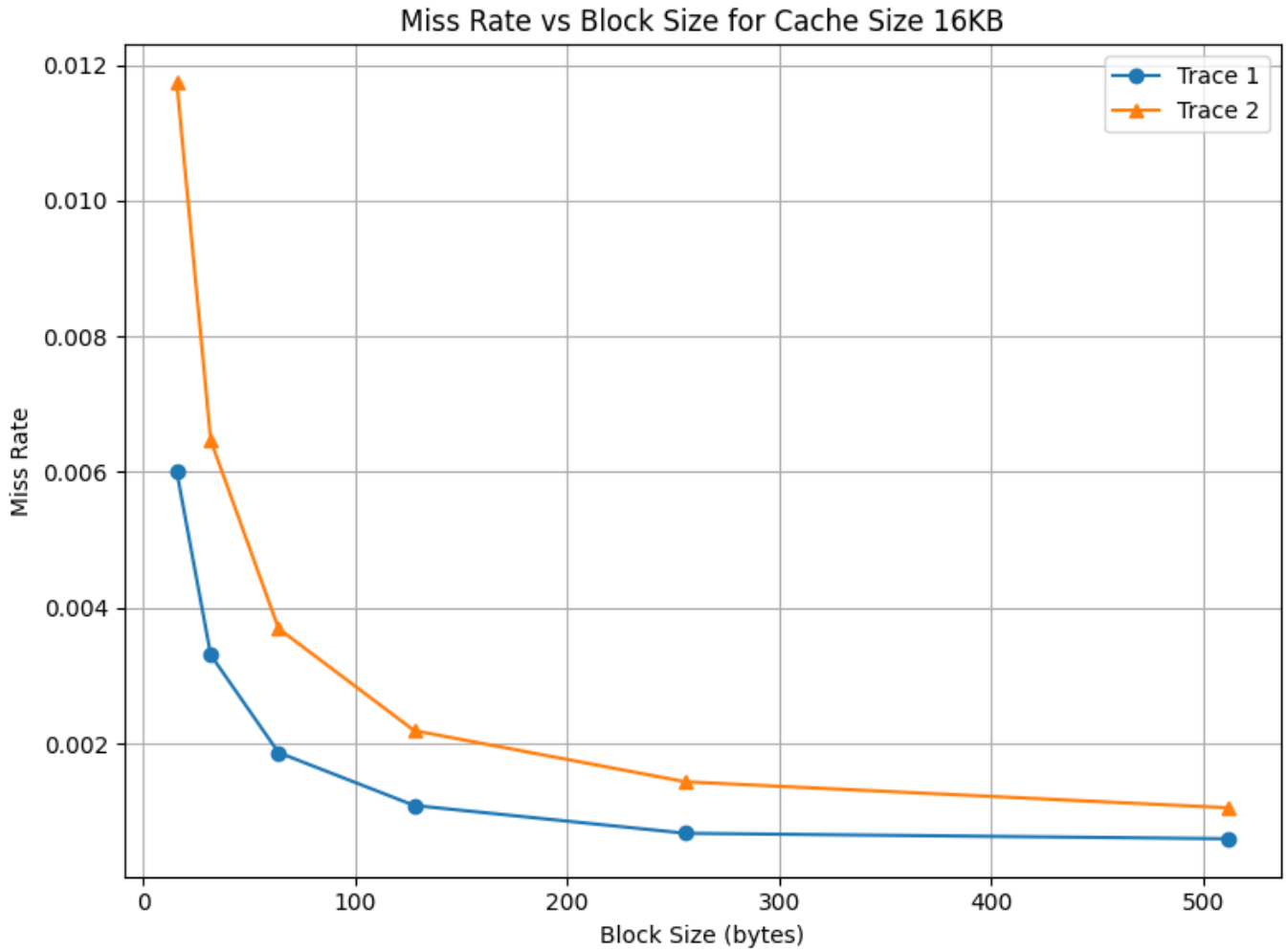
PART 1、单层Cache模拟(100分)

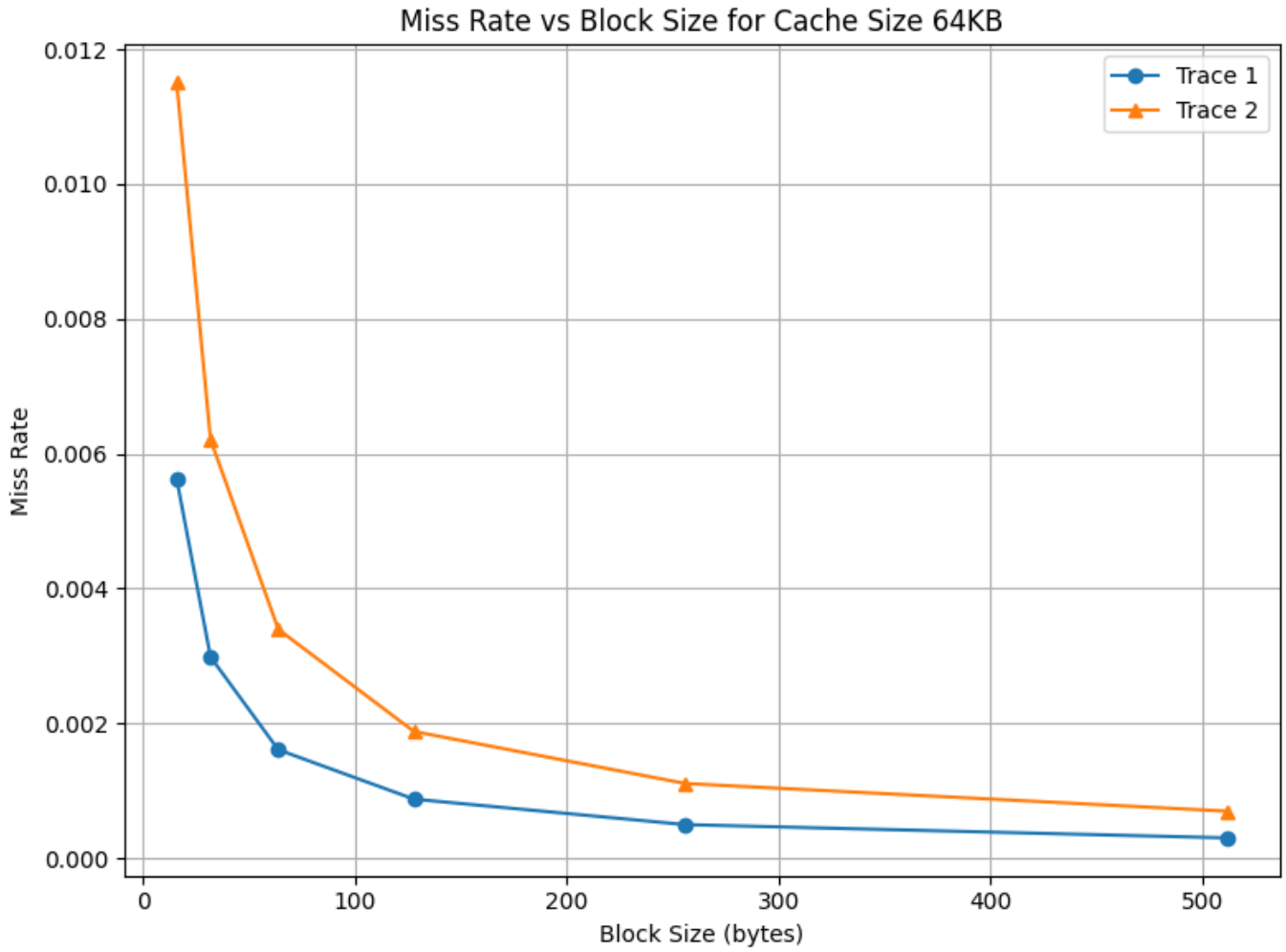
> 代码运行说明在`scripts/lab3`的`README.md`中. 具体的配置见`scripts/lab3/part1.ipynb`

1. 在不同的 Cache Size(1KB ~ 64KB) 的条件下, Miss Rate 随 Block Size变化的趋势,收集数据并绘制折线图。并说明变化原因。至少有4个不同的size大小对应的折线图。(40分)


Miss Rate vs Block Size for Cache Size 1KB
Trace 1
Trace 2
Miss Rate
0.07
0.06
0.05
0.04
0.03
0.02
20
40
60
80
100
120
Block Size (bytes)

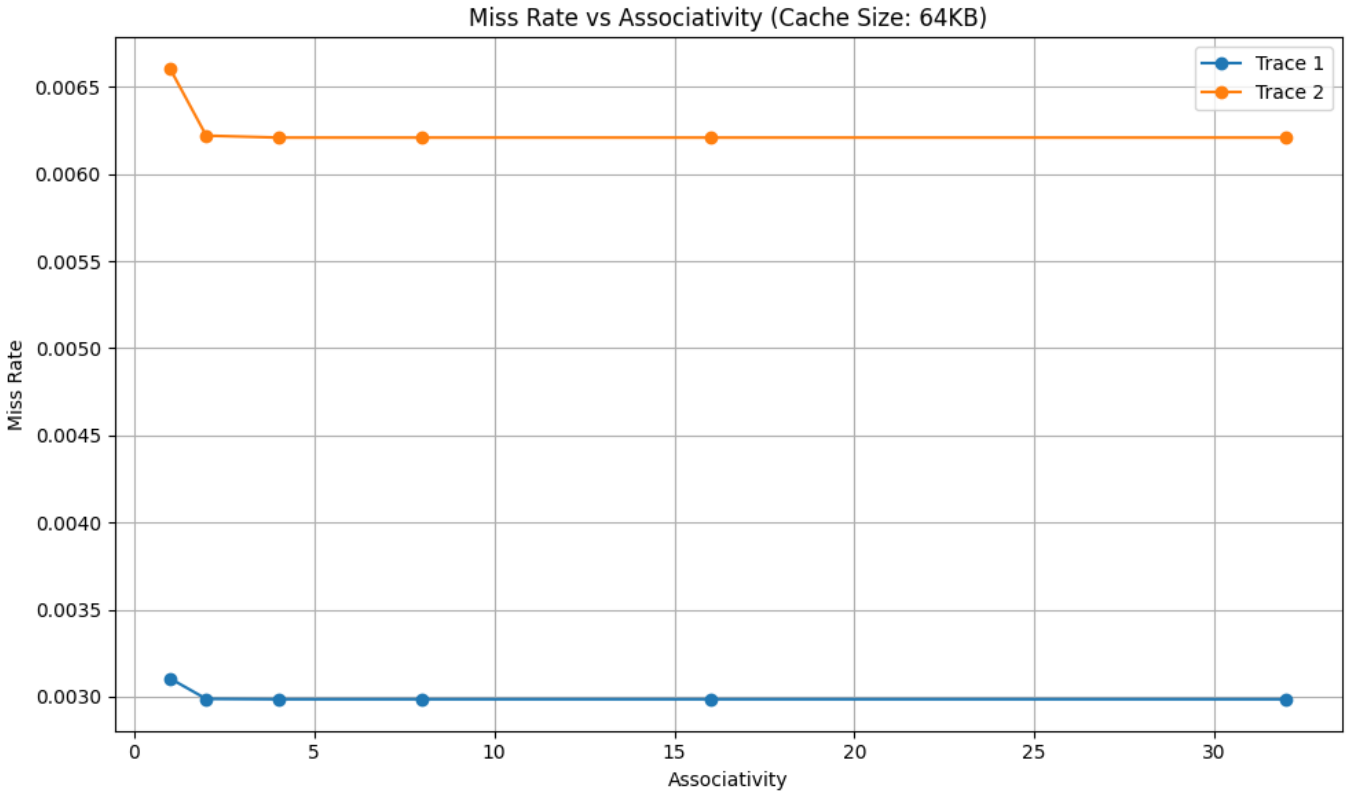

Miss Rate vs Block Size for Cache Size 4KB
Trace 1
Trace 2
Miss Rate
0.020
0.015
0.010
0.005
0
100
200
300
400
500
Block Size (bytes)


Miss Rate vs Block Size for Cache Size 16KB
Trace 1
Trace 2
Miss Rate
0.012
0.010
0.008
0.006
0.004
0.002
0
100
200
300
400
500
Block Size (bytes)


Miss Rate vs Block Size for Cache Size 64KB
Trace 1
Trace 2
Miss Rate
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
100
200
300
400
500
Block Size (bytes)

对CacheSize1KB和4KB可以看出先MissRate降低后升高,先降低主要是利用到了空间局部性，后升高是由于blocksize过大导致组减少，产生更多冲突未命中. 但是对较大的CacheSize(16KB和64KB)来说，MissRate随着BlockSize的增大而降低，这是由于Cache足够大已经很好地规避了冲突未命中。

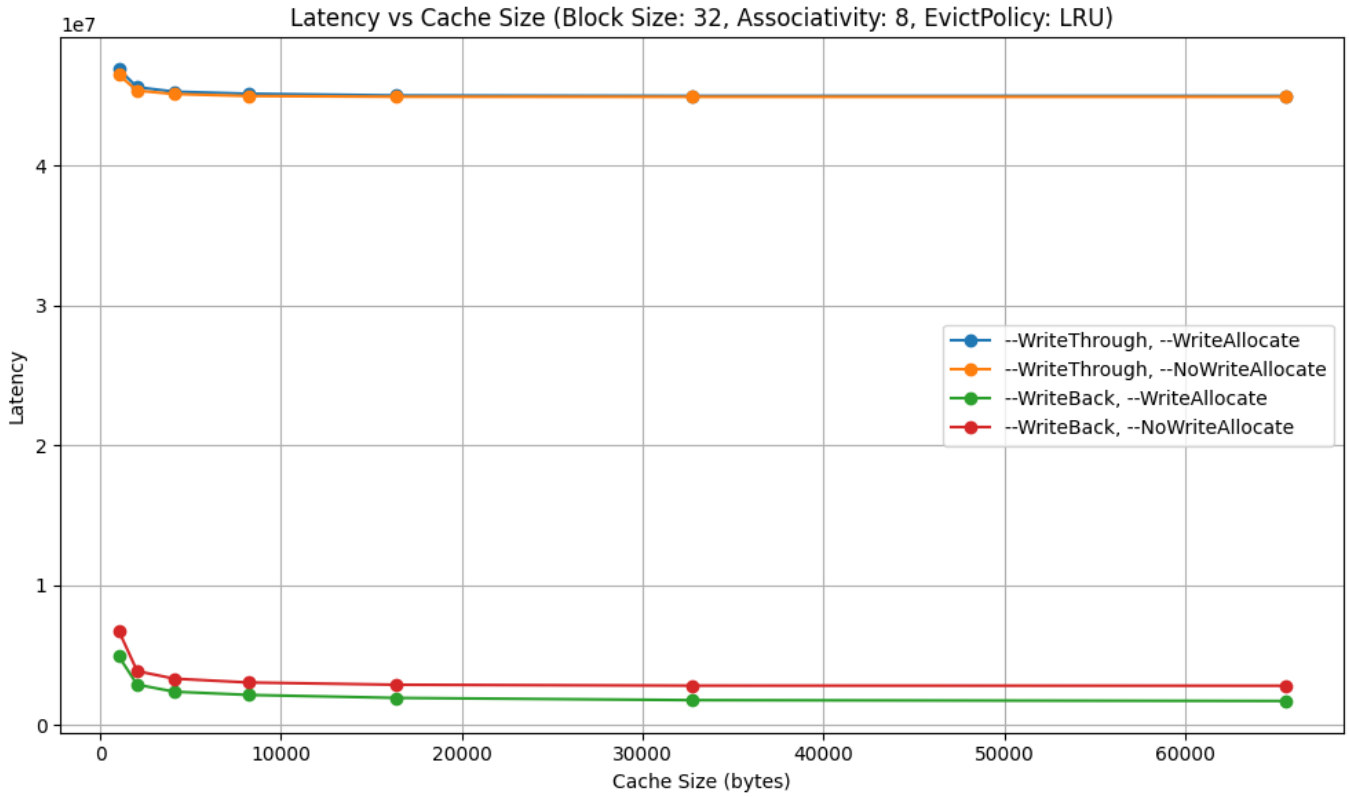
2. 在给定 Cache Size 的条件下,如64KB(L1 Cache)或512KB(L2 Cache), Miss Rate 随Associativity(1-32) 变化的趋势,收集数据并绘制折线图。并说明变化原因。至少有2、4、8、16对应的折线图。(40分)



可以看到MissRate随着Associativity的增大而降低，主要是由于Associativity增大冲突未命中减少. 但是在64KB的CacheSize下，Associativity增大到4时，MissRate已经趋于平稳，说明Associativity增大到4后组内冲突已经很小了，并无法减少强制未命中的发生。

3. 比较 Write Through 和 Write Back、Write Allocate 和 No-write allocate 的总访问延时的差异。 (20分)

> 设置的hit latency是3 cycles, miss penalty是220 cycles(和指导中要求不同但更贴合实际) Latency 可能略大于 hit latency + miss rate * miss penalty，比如说write through + write allocate需要先读主存再写主存造成两次miss penalty



可以看到WriteBack优势明显，这是因为WriteBack只在Cache中写入数据，只有在Cache替换时才会写回主存，而WriteThrough每次都要写回主存，造成了更大的延迟。 而WriteAllocate和No-write allocate的差别不大，也不确定，和访存模式有关.

PART 2. 与lab2中的处理器性能模拟器联调。(可选)

该测试中cache的配置如下:

Level	Capacity	Associativity	Block size(Bytes)	WriteUp Polity	Hit Latency	Bus Latency
L1	32KB	8ways	64	write Through	1 cpu cycle	0 cpu cycle
L2	256KB	8ways	64	write Through	8 cpu cycles	6 cpu cycles
LLC	8MB	8ways	64	write Back	20 cpu cycles	20 cpu cycles
Main Memory	128MB	-	-	--	200 cpu cycles	20 cpu cycles

加入cache前(所有访问1周期):

```
make T=huge-matrix-vec-mul PERF=pipeline_pro
```

结果如下:

```
-------Build Simulator-------
make[1]: Entering directory '/workspaces/Workspace/sim'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/workspaces/Workspace/sim'
-------Build Test-------
make[1]: Entering directory '/workspaces/Workspace/test'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/workspaces/Workspace/test'
-------Start Simulation-------
[INFO] (src/memory.c:98) Physical Memory Range:[0000000008000000,
000000000fffffff].
[INFO] (src/memory.c:104) The image is test/build/huge-matrix-vec-mul.bin, size =
67472.
HIT GOOD TRAP!
Performance Profiler: Pipeline
Dynamic instructions: 1436212
Dynamic cycles: 2296004
CPI: 1.60
Control Hazard Stall Cycles: 774
Data Hazard Stall Cycles: 215046
[INFO] (src/cpu.c:60) Program ended at pc 0800032c, with exit code 0.
```

加入cache后:

```
make T=huge-matrix-vec-mul PERF=pipeline_pro CACHE=ON
```

结果如下

```
-------Build Simulator-------
make[1]: Entering directory '/workspaces/Workspace/sim'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/workspaces/Workspace/sim'
-------Build Test-------
make[1]: Entering directory '/workspaces/Workspace/test'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/workspaces/Workspace/test'
-------Start Simulation-------
[INFO] (src/memory.c:98) Physical Memory Range:[0000000008000000,
000000000fffffff].
[INFO] (src/memory.c:104) The image is test/build/huge-matrix-vec-mul.bin, size =
67472.
HIT GOOD TRAP!
Performance Profiler: Pipeline
Dynamic instructions: 1436212
Dynamic cycles: 4047479
CPI: 2.82
Control Hazard Stall Cycles: 774
Data Hazard Stall Cycles: 215046
L1 miss rate: 0.43%
```

```
L2 miss rate: 33.71%
LLC miss rate: 100.00%
[INFO] (src/cpu.c:60) Program ended at pc 0800032c, with exit code 0.
```

测试发现在要求cache规格下此前的样例工作集大小不超过L1，因此L2和LLC的miss rate都为100%. 因此使用自定义的工作集较大的`huge-matrix-vec-mul`测试样例,结果如上.