

Git Repository Information:

GitHub Username:LizhengZhao1

URL:git@github.com:LizhengZhao1/COEN_241.git

Git commit ID: abff29ecf5818bf218d42c25276eb45fc965f880

Repository Name: COEN_241

1.Detailed configurations (CPU, Mem, etc...) of my experimental setup:

Memory: 16GB

CPU:Intel Core i5

Free Disk Space: 396.11GB

OS:Mac OS

Model Name:	MacBook Pro
Model Identifier:	MacBookPro15,2
Processor Name:	Quad-Core Intel Core i5
Processor Speed:	2.4 GHz
Number of Processors:	1
Total Number of Cores:	4
L2 Cache (per Core):	256 KB
L3 Cache:	6 MB
Hyper-Threading Technology:	Enabled
Memory:	16 GB
System Firmware Version:	1731.100.130.0.0 (iBridge: 19.16.14242.0.0,0)
OS Loader Version:	540.100.7~14
Serial Number (system):	C02YQ0YYLVDL
Hardware UUID:	70EA33ED-5A08-537D-A837-966C33E0C3B2
Provisioning UDID:	70EA33ED-5A08-537D-A837-966C33E0C3B2
Activation Lock Status:	Enabled

Hard disk:

Free:	396.11 GB (396,106,645,504 bytes)
Capacity:	499.96 GB (499,963,174,912 bytes)
Mount Point:	/System/Volumes/Update/mnt1
File System:	APFS
Writable:	Yes
Ignore Ownership:	No
BSD Name:	disk1s5
Volume UUID:	8B7CC257-4B89-474A-A9D2-B994CA382424
Physical Drive:	
Device Name:	APPLE SSD AP0512M
Media Name:	AppleAPFSMedia
Medium Type:	SSD
Protocol:	PCI-Express
Internal:	Yes
Partition Map Type:	Unknown
S.M.A.R.T. Status:	Verified

First download the ISO installation image(**ubuntu-20.04.4-live-server-amd64.iso**) from the link <https://releases.ubuntu.com/focal/ubuntu-20.04.4-live-server-amd64.iso> in HW1

a. Install QEMU:

HomeBrew is already installed in my Mac

Just execute: **\$ brew install qemu** in terminal

b. Prepare the Virtual Machine qcow2 image file

```
$ qemu-img create -f qcow2 ubuntu.img 20G
```

c. Install and Start the virtual machine

```
$ qemu-system-x86_64 \
  -machine type=q35,accel=hvf \
  -smp 4 \
  -cpu host \
  -hda ubuntu.img \
  -m 16G \
  -vga virtio \
  -usb \
  -device usb-tablet \
  -display default,show-cursor=on
-cdrom ./ubuntu-20.04.4-live-server-amd64.iso
```

For convenience, we could create three bash scripts(startUbuntu1.sh, startUbuntu2.sh, startUbuntu3.sh) which contains the above command lines to start different Ubuntu VM with different configurations(CPU4, memory 16G; CPU 1, memory 16G; CPU 4, memory 4G)

2. Main steps to enable a Docker container and some important operations :

a. Install a Docker Desktop Application from the link:

<https://docs.docker.com/desktop/mac/install/>

b. Get Image:

\$ docker pull zyclonite/sysbench

c. Start the container

\$ docker run -it zyclonite/sysbench

d. View all container commands as follows:

docker ps -a

e. Deletes all containers that have a status of exited:

\$ docker rm \$(docker ps -a -q -f status=exited)

f. Remove the specific container using the **docker rm** command:

\$ docker rm -f xxxxxxxxxxxx(container ID)

4. Screen snapshots of my Docker and QEMU running environments for each experiment:

CPU Test:

A.

CPU = 1, Memory = 16G, threads = 1, max-prime = 1000, time = 30s

a.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 20153.80

General statistics:
total time: 30.0002s
total number of events: 604652

Latency (ms):
min: 0.04
avg: 0.05
max: 8.10
95th percentile: 0.07
sum: 29844.10

Threads fairness:
events (avg/stddev): 604652.0000/0.00
execution time (avg/stddev): 29.8441/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 29760.84

General statistics:
total time: 30.0001s
total number of events: 892864

Latency (ms):
min: 0.03
avg: 0.03
max: 3.13
95th percentile: 0.05
sum: 29799.64

Threads fairness:
events (avg/stddev): 892864.0000/0.00
execution time (avg/stddev): 29.7996/0.00
```

b.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 20813.07

General statistics:
total time: 30.0002s
total number of events: 624431

Latency (ms):
min: 0.04
avg: 0.05
max: 2.88
95th percentile: 0.07
sum: 29865.34

Threads fairness:
events (avg/stddev): 624431.0000/0.00
execution time (avg/stddev): 29.8653/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 30160.61

General statistics:
total time: 30.0001s
total number of events: 904860

Latency (ms):
min: 0.03
avg: 0.03
max: 4.66
95th percentile: 0.05
sum: 29815.31

Threads fairness:
events (avg/stddev): 904860.0000/0.00
execution time (avg/stddev): 29.8153/0.00
```

C.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 20715.77

General statistics:
  total time:          30.0002s
  total number of events: 621507

Latency (ms):
  min:                 0.04
  avg:                 0.05
  max:                 1.38
  95th percentile:    0.07
  sum:                29844.45

Threads fairness:
  events (avg/stddev):   621507.0000/0.00
  execution time (avg/stddev): 29.8445/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 30453.07

General statistics:
  total time:          30.0001s
  total number of events: 913634

Latency (ms):
  min:                 0.03
  avg:                 0.03
  max:                 1.17
  95th percentile:    0.05
  sum:                29843.17

Threads fairness:
  events (avg/stddev):   913634.0000/0.00
  execution time (avg/stddev): 29.8432/0.00
```

d.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 20746.22

General statistics:
  total time:          30.0002s
  total number of events: 622422

Latency (ms):
  min:                 0.04
  avg:                 0.05
  max:                 1.51
  95th percentile:    0.07
  sum:                29852.67

Threads fairness:
  events (avg/stddev):   622422.0000/0.00
  execution time (avg/stddev): 29.8527/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 30428.48

General statistics:
  total time:          30.0001s
  total number of events: 912905

Latency (ms):
  min:                 0.03
  avg:                 0.03
  max:                 1.23
  95th percentile:    0.05
  sum:                29836.50

Threads fairness:
  events (avg/stddev):   912905.0000/0.00
  execution time (avg/stddev): 29.8365/0.00
```

e.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 19940.11

General statistics:
total time: 30.0001s
total number of events: 598244

Latency (ms):
min: 0.04
avg: 0.05
max: 8.01
95th percentile: 0.08
sum: 29795.95

Threads fairness:
events (avg/stddev): 598244.0000/0.00
execution time (avg/stddev): 29.7959/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 30426.12

General statistics:
total time: 30.0001s
total number of events: 912829

Latency (ms):
min: 0.03
avg: 0.03
max: 2.02
95th percentile: 0.05
sum: 29841.60

Threads fairness:
events (avg/stddev): 912829.0000/0.00
execution time (avg/stddev): 29.8416/0.00
```

B.

CPU = 4, Memory = 16G, threads = 1, max-prime = 1000, time = 30s

a.

Docker:

```
mac@MacBook-Pro-7 ~ % docker run -it --cpus=4 -m 16G zyclonite/sysbench --test=cpu --cpu-max-prime=1000 --time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 21017.78

General statistics:
total time: 30.0001s
total number of events: 630569

Latency (ms):
min: 0.04
avg: 0.05
max: 2.48
95th percentile: 0.07
sum: 29874.29

Threads fairness:
events (avg/stddev): 630569.0000/0.00
execution time (avg/stddev): 29.8743/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 30617.10

General statistics:
total time: 30.00002s
total number of events: 918562

Latency (ms):
min: 0.03
avg: 0.03
max: 1.07
95th percentile: 0.05
sum: 29851.16

Threads fairness:
events (avg/stddev): 918562.0000/0.00
execution time (avg/stddev): 29.8512/0.00
```

b.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 21049.62

General statistics:
total time: 30.0002s
total number of events: 631530

Latency (ms):
min: 0.04
avg: 0.05
max: 4.48
95th percentile: 0.07
sum: 29875.90

Threads fairness:
events (avg/stddev): 631530.0000/0.00
execution time (avg/stddev): 29.8759/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 30205.08

General statistics:
total time: 30.0001s
total number of events: 906202

Latency (ms):
min: 0.03
avg: 0.03
max: 2.12
95th percentile: 0.05
sum: 29829.48

Threads fairness:
events (avg/stddev): 906202.0000/0.00
execution time (avg/stddev): 29.8295/0.00
```

C.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 21028.27

General statistics:
    total time:          30.0002s
    total number of events: 630884

Latency (ms):
    min:                 0.04
    avg:                 0.05
    max:                 4.50
    95th percentile:     0.07
    sum:                29870.44

Threads fairness:
    events (avg/stddev): 630884.0000/0.00
    execution time (avg/stddev): 29.8704/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 30153.61

General statistics:
    total time:          30.0003s
    total number of events: 904668

Latency (ms):
    min:                 0.03
    avg:                 0.03
    max:                 1.72
    95th percentile:     0.05
    sum:                29824.31

Threads fairness:
    events (avg/stddev): 904668.0000/0.00
    execution time (avg/stddev): 29.8243/0.00
```

d.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 20769.65

General statistics:
total time: 30.0002s
total number of events: 623125

Latency (ms):
min: 0.04
avg: 0.05
max: 1.53
95th percentile: 0.07
sum: 29854.29

Threads fairness:
events (avg/stddev): 623125.0000/0.00
execution time (avg/stddev): 29.8543/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 30305.34

General statistics:
total time: 30.0002s
total number of events: 909206

Latency (ms):
min: 0.03
avg: 0.03
max: 9.76
95th percentile: 0.05
sum: 29833.28

Threads fairness:
events (avg/stddev): 909206.0000/0.00
execution time (avg/stddev): 29.8333/0.00
```

e.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 20418.23

General statistics:
  total time:          30.0002s
  total number of events: 612585

Latency (ms):
  min:                 0.04
  avg:                 0.05
  max:                 4.62
  95th percentile:    0.07
  sum:                29845.29

Threads fairness:
  events (avg/stddev): 612585.0000/0.00
  execution time (avg/stddev): 29.8453/0.00
```

QEMU:

```
Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 30542.91

General statistics:
  total time:          30.0001s
  total number of events: 916328

Latency (ms):
  min:                 0.03
  avg:                 0.03
  max:                 1.00
  95th percentile:    0.05
  sum:                29854.25

Threads fairness:
  events (avg/stddev): 916328.0000/0.00
  execution time (avg/stddev): 29.8543/0.00
```

C.

CPU = 4, Memory = 4G, threads = 1, max-prime = 1000, time = 30s

a.

Docker

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!
CPU speed:
  events per second: 20079.11

General statistics:
  total time:           30.0001s
  total number of events: 602412

Latency (ms):
  min:                 0.04
  avg:                 0.05
  max:                 3.17
  95th percentile:    0.08
  sum:                29826.05

Threads fairness:
  events (avg/stddev): 602412.0000/0.00
  execution time (avg/stddev): 29.8260/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
>

Prime numbers limit: 1000
> Initializing worker threads...
> Threads started!
> CPU speed:
>   events per second: 29401.49

General statistics:
  total time:           30.0001s
  total number of events: 882088

Latency (ms):
  min:                 0.03
  avg:                 0.03
  max:                 3.11
  95th percentile:    0.05
  sum:                29770.48

Threads fairness:
  events (avg/stddev): 882088.0000/0.00
  execution time (avg/stddev): 29.7705/0.00
```

b.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 19988.05

General statistics:
total time: 30.0002s
total number of events: 599684

Latency (ms):
min: 0.04
avg: 0.05
max: 4.36
95th percentile: 0.08
sum: 29811.66

Threads fairness:
events (avg/stddev): 599684.0000/0.00
execution time (avg/stddev): 29.8117/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 30383.38

General statistics:
total time: 30.0002s
total number of events: 911548

Latency (ms):
min: 0.03
avg: 0.03
max: 1.25
95th percentile: 0.05
sum: 29843.16

Threads fairness:
events (avg/stddev): 911548.0000/0.00
execution time (avg/stddev): 29.8432/0.00
```

C.

Docker:

```
| Running the test with following options:  
| Number of threads: 1  
| Initializing random number generator from current time  
  
Prime numbers limit: 1000  
  
Initializing worker threads...  
  
Threads started!  
  
CPU speed:  
    events per second: 20796.42  
  
General statistics:  
    total time:          30.0002s  
    total number of events: 623928  
  
Latency (ms):  
    min:                 0.04  
    avg:                 0.05  
    max:                11.71  
    95th percentile:     0.07  
    sum:               29856.91  
  
Threads fairness:  
    events (avg/stddev): 623928.0000/0.00  
    execution time (avg/stddev): 29.8569/0.00
```

QEMU:

```
| Running the test with following options:  
| Number of threads: 1  
| Initializing random number generator from current time  
  
Prime numbers limit: 1000  
  
Initializing worker threads...  
  
Threads started!  
  
CPU speed:  
    events per second: 30452.66  
  
General statistics:  
    total time:          30.0003s  
    total number of events: 913628  
  
Latency (ms):  
    min:                 0.03  
    avg:                 0.03  
    max:                2.13  
    95th percentile:     0.05  
    sum:               29848.56  
  
Threads fairness:  
    events (avg/stddev): 913628.0000/0.00  
    execution time (avg/stddev): 29.8486/0.00
```

d.

Docker:

```
Running the test with following options:  
Number of threads: 1  
Initializing random number generator from current time  
  
Prime numbers limit: 1000  
  
Initializing worker threads...  
  
Threads started!  
  
CPU speed:  
    events per second: 20541.21  
  
General statistics:  
    total time: 30.0001s  
    total number of events: 616269  
  
Latency (ms):  
    min: 0.04  
    avg: 0.05  
    max: 1.77  
    95th percentile: 0.07  
    sum: 29866.25  
  
Threads fairness:  
    events (avg/stddev): 616269.0000/0.00  
    execution time (avg/stddev): 29.8663/0.00
```

QEMU:

```
Running the test with following options:  
Number of threads: 1  
Initializing random number generator from current time  
  
Prime numbers limit: 1000  
  
Initializing worker threads...  
  
Threads started!  
  
CPU speed:  
    events per second: 29885.95  
  
General statistics:  
    total time: 30.0002s  
    total number of events: 896621  
  
Latency (ms):  
    min: 0.03  
    avg: 0.03  
    max: 4.43  
    95th percentile: 0.05  
    sum: 29796.95  
  
Threads fairness:  
    events (avg/stddev): 896621.0000/0.00  
    execution time (avg/stddev): 29.7969/0.00
```

e.

Docker:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 20542.73

General statistics:
total time: 30.0002s
total number of events: 616315

Latency (ms):
min: 0.04
avg: 0.05
max: 0.56
95th percentile: 0.07
sum: 29864.97

Threads fairness:
events (avg/stddev): 616315.0000/0.00
execution time (avg/stddev): 29.8650/0.00
```

QEMU:

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 1000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 30237.21

General statistics:
total time: 30.0002s
total number of events: 907163

Latency (ms):
min: 0.03
avg: 0.03
max: 9.26
95th percentile: 0.05
sum: 29830.43

Threads fairness:
events (avg/stddev): 907163.0000/0.00
execution time (avg/stddev): 29.8304/0.00
```

File I/O Test:

A.a.CPU = 4, Memory = 16G, file-total-size=1G file-test-mode=rndrw, time = 10s(default)

Docker:

```
File operations:
  reads/s:          3492.15
  writes/s:         2328.10
  fsyncs/s:         7451.41

Throughput:
  read, MiB/s:      54.56
  written, MiB/s:   36.38

General statistics:
  total time:       10.0152s
  total number of events: 132811

Latency (ms):
  min:              0.00
  avg:              0.07
  max:              216.31
  95th percentile:  0.27
  sum:              9941.37

Threads fairness:
  events (avg/stddev): 132811.0000/0.00
  execution time (avg/stddev): 9.9414/0.00
```

QEMU:

```
File operations:
  reads/s:          1817.69
  writes/s:         1211.79
  fsyncs/s:         3877.84

Throughput:
  read, MiB/s:      28.40
  written, MiB/s:   18.93

General statistics:
  total time:       10.0321s
  total number of events: 69185

Latency (ms):
  min:              0.00
  avg:              0.14
  max:              27.69
  95th percentile:  0.52
  sum:              9955.98

Threads fairness:
  events (avg/stddev): 69185.0000/0.00
  execution time (avg/stddev): 9.9560/0.00
```

b.

Docker:

```
Threads started!

File operations:
    reads/s:          3450.96
    writes/s:         2300.64
    fsyncs/s:        7372.83

Throughput:
    read, MiB/s:      53.92
    written, MiB/s:   35.95

General statistics:
    total time:       10.0121s
    total number of events: 131308

Latency (ms):
    min:              0.00
    avg:              0.08
    max:              2.46
    95th percentile:  0.27
    sum:             9940.07

Threads fairness:
    events (avg/stddev): 131308.0000/0.00
    execution time (avg/stddev): 9.9401/0.00
```

QEMU:

```
File operations:
    reads/s:          1494.63
    writes/s:         996.42
    fsyncs/s:        3197.11

Throughput:
    read, MiB/s:      23.35
    written, MiB/s:   15.57

General statistics:
    total time:       10.0347s
    total number of events: 56958

Latency (ms):
    min:              0.00
    avg:              0.17
    max:              54.25
    95th percentile:  0.55
    sum:             9963.74

Threads fairness:
    events (avg/stddev): 56958.0000/0.00
    execution time (avg/stddev): 9.9637/0.00

sysbench 1.0.18 (using system LuAJIT 2.1.0-beta3)
```

C.

Docker:

```
Threads started!

File operations:
  reads/s:          3414.70
  writes/s:         2276.47
  fsyncs/s:         7289.79

Throughput:
  read, MiB/s:      53.35
  written, MiB/s:   35.57

General statistics:
  total time:       10.0139s
  total number of events: 129883

Latency (ms):
  min:              0.00
  avg:              0.08
  max:              43.67
  95th percentile:  0.28
  sum:              9936.62

Threads fairness:
  events (avg/stddev): 129883.0000/0.00
  execution time (avg/stddev): 9.9366/0.00
```

QEMU:

```
File operations:
  reads/s:          1597.09
  writes/s:         1064.73
  fsyncs/s:         3417.60

Throughput:
  read, MiB/s:      24.95
  written, MiB/s:   16.64

General statistics:
  total time:       10.0285s
  total number of events: 60853

Latency (ms):
  min:              0.00
  avg:              0.16
  max:              7.10
  95th percentile:  0.54
  sum:              9966.35

Threads fairness:
  events (avg/stddev): 60853.0000/0.00
  execution time (avg/stddev): 9.9663/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

d.

Docker:

```
File operations:
  reads/s:          3420.28
  writes/s:         2280.18
  fsyncs/s:         7299.59

Throughput:
  read, MiB/s:      53.44
  written, MiB/s:   35.63

General statistics:
  total time:       10.0153s
  total number of events: 130090

Latency (ms):
  min:              0.00
  avg:              0.08
  max:              7.90
  95th percentile:  0.27
  sum:              9943.55

Threads fairness:
  events (avg/stddev): 130090.0000/0.00
  execution time (avg/stddev): 9.9435/0.00
```

QEMU:

```
File operations:
  reads/s:          1549.31
  writes/s:         1032.87
  fsyncs/s:         3314.86

Throughput:
  read, MiB/s:      24.21
  written, MiB/s:   16.14

General statistics:
  total time:       10.0281s
  total number of events: 59021

Latency (ms):
  min:              0.00
  avg:              0.17
  max:              38.71
  95th percentile:  0.54
  sum:              9965.57

Threads fairness:
  events (avg/stddev): 59021.0000/0.00
  execution time (avg/stddev): 9.9656/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

e.

Docker:

```
Threads started!

File operations:
    reads/s:                3401.18
    writes/s:               2267.45
    fsyncs/s:                7257.94

Throughput:
    read, MiB/s:            53.14
    written, MiB/s:          35.43

General statistics:
    total time:              10.0180s
    total number of events: 129397

Latency (ms):
    min:                      0.00
    avg:                      0.08
    max:                     103.90
    95th percentile:          0.27
    sum:                     9943.39

Threads fairness:
    events (avg/stddev):   129397.0000/0.00
    execution time (avg/stddev): 9.9434/0.00

/ # sysbench fileio --file-total-size=1G --file-test-mode=rndrw cleanup
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)
```

QEMU:

```
File operations:
    reads/s:                1626.66
    writes/s:               1084.44
    fsyncs/s:                3474.98

Throughput:
    read, MiB/s:            25.42
    written, MiB/s:          16.94

General statistics:
    total time:              10.0314s
    total number of events: 61936

Latency (ms):
    min:                      0.00
    avg:                      0.16
    max:                     30.77
    95th percentile:          0.58
    sum:                     9959.91

Threads fairness:
    events (avg/stddev):   61936.0000/0.00
    execution time (avg/stddev): 9.9599/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

B.

CPU = 4, Memory = 4G, file-total-size=1G file-test-mode=rndrw, time = 10s(default)

a.

Docker:

```
File operations:
  reads/s:          3468.22
  writes/s:         2312.15
  fsyncs/s:         7401.76

Throughput:
  read, MiB/s:      54.19
  written, MiB/s:   36.13

General statistics:
  total time:       10.0152s
  total number of events: 131913

Latency (ms):
  min:              0.00
  avg:              0.08
  max:              88.10
  95th percentile:  0.27
  sum:              9935.56

Threads fairness:
  events (avg/stddev): 131913.0000/0.00
  execution time (avg/stddev): 9.9356/0.00
```

QEMU:

```
File operations:
  reads/s:          1879.87
  writes/s:         1253.25
  fsyncs/s:         4017.98

Throughput:
  read, MiB/s:      29.37
  written, MiB/s:   19.58

General statistics:
  total time:       10.0208s
  total number of events: 71540

Latency (ms):
  min:              0.00
  avg:              0.14
  max:              37.94
  95th percentile:  0.49
  sum:              9950.59

Threads fairness:
  events (avg/stddev): 71540.0000/0.00
  execution time (avg/stddev): 9.9506/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

b.

Docker:

```
File operations:
    reads/s:                2982.73
    writes/s:               1988.49
    fsyncs/s:                6364.36

Throughput:
    read, MiB/s:            46.61
    written, MiB/s:          31.07

General statistics:
    total time:              10.0162s
    total number of events:   113428

Latency (ms):
    min:                      0.00
    avg:                      0.09
    max:                     112.48
    95th percentile:           0.30
    sum:                     9941.08

Threads fairness:
    events (avg/stddev):    113428.0000/0.00
    execution time (avg/stddev): 9.9411/0.00
```

QEMU:

```
File operations:
    reads/s:                1386.88
    writes/s:               924.58
    fsyncs/s:                2963.85

Throughput:
    read, MiB/s:            21.67
    written, MiB/s:          14.45

General statistics:
    total time:              10.0357s
    total number of events:   52820

Latency (ms):
    min:                      0.00
    avg:                      0.19
    max:                     47.98
    95th percentile:           0.58
    sum:                     9965.47

Threads fairness:
    events (avg/stddev):    52820.0000/0.00
    execution time (avg/stddev): 9.9655/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

c.

Docker:

```
File operations:
  reads/s:          3102.49
  writes/s:         2068.33
  fsyncs/s:         6618.95

Throughput:
  read, MiB/s:      48.48
  written, MiB/s:   32.32

General statistics:
  total time:       10.0163s
  total number of events: 117979

Latency (ms):
  min:              0.00
  avg:              0.08
  max:              116.07
  95th percentile:  0.30
  sum:              9942.15

Threads fairness:
  events (avg/stddev): 117979.0000/0.00
  execution time (avg/stddev): 9.9421/0.00

/ # sysbench fileio --file-total-size=1G --file-test-mode=rndrw cleanup
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)
```

QEMU:

```
File operations:
  reads/s:          1422.20
  writes/s:         948.13
  fsyncs/s:         3089.90

Throughput:
  read, MiB/s:      22.22
  written, MiB/s:   14.81

General statistics:
  total time:       10.0396s
  total number of events: 54195

Latency (ms):
  min:              0.00
  avg:              0.18
  max:              6.13
  95th percentile:  0.59
  sum:              9955.41

Threads fairness:
  events (avg/stddev): 54195.0000/0.00
  execution time (avg/stddev): 9.9554/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

d.

Docker:

```
File operations:
  reads/s:          3126.74
  writes/s:         2084.49
  fsyncs/s:         6673.77

Throughput:
  read, MiB/s:      48.86
  written, MiB/s:   32.57

General statistics:
  total time:       10.0150s
  total number of events: 118922

Latency (ms):
  min:              0.00
  avg:              0.08
  max:              160.32
  95th percentile:  0.29
  sum:              9941.33

Threads fairness:
  events (avg/stddev): 118922.0000/0.00
  execution time (avg/stddev): 9.9413/0.00
```

QEMU:

```
File operations:
  reads/s:          1488.40
  writes/s:         992.26
  fsyncs/s:         3178.83

Throughput:
  read, MiB/s:      23.26
  written, MiB/s:   15.50

General statistics:
  total time:       10.0356s
  total number of events: 56680

Latency (ms):
  min:              0.00
  avg:              0.18
  max:              7.15
  95th percentile:  0.56
  sum:              9959.48

Threads fairness:
  events (avg/stddev): 56680.0000/0.00
  execution time (avg/stddev): 9.9595/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

e.

Docker:

```
File operations:
  reads/s:          3317.62
  writes/s:         2211.75
  fsyncs/s:         7077.70

Throughput:
  read, MiB/s:      51.84
  written, MiB/s:   34.56

General statistics:
  total time:       10.0168s
  total number of events: 126185

Latency (ms):
  min:              0.00
  avg:              0.08
  max:              7.09
  95th percentile:  0.29
  sum:              9938.52

Threads fairness:
  events (avg/stddev): 126185.0000/0.00
  execution time (avg/stddev): 9.9385/0.00
```

QEMU:

```
File operations:
  reads/s:          1387.95
  writes/s:         925.30
  fsyncs/s:         2967.04

Throughput:
  read, MiB/s:      21.69
  written, MiB/s:   14.46

General statistics:
  total time:       10.0276s
  total number of events: 52829

Latency (ms):
  min:              0.00
  avg:              0.19
  max:              7.59
  95th percentile:  0.61
  sum:              9962.26

Threads fairness:
  events (avg/stddev): 52829.0000/0.00
  execution time (avg/stddev): 9.9623/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

C.

CPU = 1, Memory = 16G, file-total-size=1G file-test-mode=rndrw, time = 10s(default)

a.

Docker:

```
File operations:
  reads/s:                3228.67
  writes/s:               2152.44
  fsyncs/s:                6891.32

Throughput:
  read, MiB/s:             50.45
  written, MiB/s:           33.63

General statistics:
  total time:              10.0149s
  total number of events:   122799

Latency (ms):
  min:                      0.00
  avg:                      0.08
  max:                     375.34
  95th percentile:          0.27
  sum:                     9936.92

Threads fairness:
  events (avg/stddev):    122799.0000/0.00
  execution time (avg/stddev): 9.9369/0.00
```

QEMU:

```
File operations:
  reads/s:                2179.00
  writes/s:               1452.67
  fsyncs/s:                4656.62

Throughput:
  read, MiB/s:             34.05
  written, MiB/s:           22.70

General statistics:
  total time:              10.0207s
  total number of events:   82945

Latency (ms):
  min:                      0.00
  avg:                      0.12
  max:                     31.06
  95th percentile:          0.44
  sum:                     9951.93

Threads fairness:
  events (avg/stddev):    82945.0000/0.00
  execution time (avg/stddev): 9.9519/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

b.

Docker:

```
File operations:
  reads/s:          3537.78
  writes/s:         2358.52
  fsyncs/s:         7555.25

Throughput:
  read, MiB/s:      55.28
  written, MiB/s:   36.85

General statistics:
  total time:       10.0215s
  total number of events: 134700

Latency (ms):
  min:              0.00
  avg:              0.07
  max:              2.18
  95th percentile:  0.27
  sum:              9935.30

Threads fairness:
  events (avg/stddev): 134700.0000/0.00
  execution time (avg/stddev): 9.9353/0.00
```

QEMU:

```
File operations:
  reads/s:          1896.87
  writes/s:         1264.58
  fsyncs/s:         4059.11

Throughput:
  read, MiB/s:      29.64
  written, MiB/s:   19.76

General statistics:
  total time:       10.0246s
  total number of events: 72273

Latency (ms):
  min:              0.00
  avg:              0.14
  max:              37.39
  95th percentile:  0.47
  sum:              9953.25

Threads fairness:
  events (avg/stddev): 72273.0000/0.00
  execution time (avg/stddev): 9.9533/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

c.

Docker:

```
File operations:
  reads/s:          3283.71
  writes/s:         2189.14
  fsyncs/s:         7016.53

Throughput:
  read, MiB/s:      51.31
  written, MiB/s:   34.21

General statistics:
  total time:       10.0106s
  total number of events: 124929

Latency (ms):
  min:              0.00
  avg:              0.08
  max:              23.25
  95th percentile:  0.29
  sum:              9937.89

Threads fairness:
  events (avg/stddev): 124929.0000/0.00
  execution time (avg/stddev): 9.9379/0.00
```

QEMU:

```
File operations:
  reads/s:          2128.24
  writes/s:         1418.83
  fsyncs/s:         4547.23

Throughput:
  read, MiB/s:      33.25
  written, MiB/s:   22.17

General statistics:
  total time:       10.0339s
  total number of events: 81110

Latency (ms):
  min:              0.00
  avg:              0.12
  max:              7.11
  95th percentile:  0.44
  sum:              9954.09

Threads fairness:
  events (avg/stddev): 81110.0000/0.00
  execution time (avg/stddev): 9.9541/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

d.

Docker:

```
File operations:
  reads/s:          3372.26
  writes/s:         2248.17
  fsyncs/s:         7198.94

Throughput:
  read, MiB/s:      52.69
  written, MiB/s:   35.13

General statistics:
  total time:       10.0156s
  total number of events: 128284

Latency (ms):
  min:              0.00
  avg:              0.08
  max:              191.67
  95th percentile:  0.27
  sum:              9939.64

Threads fairness:
  events (avg/stddev): 128284.0000/0.00
  execution time (avg/stddev): 9.9396/0.00
```

QEMU:

```
File operations:
  reads/s:          2076.17
  writes/s:         1384.12
  fsyncs/s:         4431.06

Throughput:
  read, MiB/s:      32.44
  written, MiB/s:   21.63

General statistics:
  total time:       10.0268s
  total number of events: 79007

Latency (ms):
  min:              0.00
  avg:              0.13
  max:              7.07
  95th percentile:  0.46
  sum:              9952.72

Threads fairness:
  events (avg/stddev): 79007.0000/0.00
  execution time (avg/stddev): 9.9527/0.00

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

e.

Docker:

```
File operations:
  reads/s:          3272.15
  writes/s:         2181.43
  fsyncs/s:         6992.27

Throughput:
  read, MiB/s:      51.13
  written, MiB/s:   34.08

General statistics:
  total time:       10.0103s
  total number of events: 124477

Latency (ms):
  min:              0.00
  avg:              0.08
  max:              6.22
  95th percentile:  0.28
  sum:              9936.37

Threads fairness:
  events (avg/stddev): 124477.0000/0.00
  execution time (avg/stddev): 9.9364/0.00
```

QEMU:

```
File operations:
  reads/s:          2059.03
  writes/s:         1372.68
  fsyncs/s:         4398.08

Throughput:
  read, MiB/s:      32.17
  written, MiB/s:   21.45

General statistics:
  total time:       10.0218s
  total number of events: 78359

Latency (ms):
  min:              0.00
  avg:              0.13
  max:              14.77
  95th percentile:  0.46
  sum:              9949.09

Threads fairness:
  events (avg/stddev): 78359.0000/0.00
  execution time (avg/stddev): 9.9491/0.00
```

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

5. How I conduct my measurements in three different scenarios for each virtualization technology:

CPU Test:

Docker:

- a. Just execute the docker ***command line*** in the home path of the terminal

In the command line, we should also specify the number of cores of CPU, amount of RAM and etc.

command line:

```
$ docker run -it --cpus=4 -m 16G zyclonite/sysbench --test(cpu
```

```
--cpu-max-prime=1000 --time=30 run
```

--cpus=? (this command could help us specify the number of cores of CPU)

-m ? (this command could help us specify the amount of RAM)

At this homework, I used 3 combinations. 1.--cpus=4 -m 16G, 2.--cpus=4 -m 4G

3. --cpus=1 -m 16G

After executing this command line, we should also clear containers which have expired by executing this command line:

```
$ docker rm $(docker ps -a -q -f status=exited)
```

This step can prevent the impact of expired containers on experiments

After these steps, we could get the result of CPU Test of Docker.

QEMU(Ubuntu):

- a. Firstly, we need to use the **cd** command to help us switch to the directory where Ubuntu is installed:

On my mac, I need to execute the below command:

```
$ cd ~/desktop/scu/cloud_computing/hw1
```

- b. Then, at the current directory, we need to execute the below command lines to start our Ubuntu VM

```
$ qemu-system-x86_64 \
  -machine type=q35,accel=hvf \
  -smp 4 \
  -cpu host \
  -hda ubuntu.img \
  -m 16G \
  -vga virtio \
  -usb \
  -device usb-tablet \
  -display default,show-cursor=on
-cdrom ./ubuntu-20.04.4-live-server-amd64.iso
```

In these command lines, we could specify the number of cores of CPU and the amount of RAM of Ubuntu VM.

-smp ? (specify the number of cores of CPU)

-m ? (specify the amount of RAM)

At this homework, I used 3 combinations. 1.-smp 4 -m 16G, 2.-smp=4 -m 4G

3. -smp 1 -m 16G

c. After starting Ubuntu VM, we could execute sysbench command lines to test the CPU in Ubuntu terminal.

For convenience, we could create a bash script to automate the experiments:

```
$ nano cpuTest_ubuntu.sh
```

I created a bash script called **cpuTest.sh** which contains the below command line

```
$ sysbench --test=cpu --cpu-max-prime=1000 --time=30 run
```

Then we can just execute the below command line to test CPU in Ubuntu terminal

```
$ bash cpuTest.sh
```

After these steps, we could get the result of CPU Test of QEMU(Ubuntu).

File I/O Test:

Docker:

a. First we need to execute the below command line to start our sysbench container at the home path in terminal

```
$ docker run --rm -it --cpus=4 -m 16G --entrypoint /bin/sh zyclonite/sysbench
```

--entrypoint /bin/sh command could prevent the expiration of sysbench container

--cpus=? (this command could help us specify the number of cores of CPU)

-m ? (this command could help us specify the amount of RAM)

At this homework, I used 3 combinations. 1.--cpus=4 -m 16G, 2.--cpus=4 -m 4G

3. --cpus=1 -m 16G

b. Then we need to execute all of these below command lines at once

```
$ sysbench fileio --file-total-size=1G --file-test-mode=rndrw prepare
```

```
$ sysbench fileio --file-total-size=1G --file-test-mode=rndrw run
```

```
$ sysbench fileio --file-total-size=1G --file-test-mode=rndrw cleanup
```

After these steps, we could get the result of File I/O Test of Docker.

QEMU(Ubuntu):

a. Firstly, we need to use the **cd** command to help us switch to the directory where Ubuntu is installed:

On my mac, I need to execute the below command:

```
$ cd ~/desktop/scu/cloud_computing/hw1
```

b. Then, at the current directory, we need to execute the below command lines to start our Ubuntu VM

```
$ qemu-system-x86_64 \
    -machine type=q35,accel=hvf \
    -smp 4 \
    -cpu host \
    -hda ubuntu.img \
    -m 16G \
    -vga virtio \
    -usb \
    -device usb-tablet \
    -display default,show-cursor=on
-cdrom ./ubuntu-20.04.4-live-server-amd64.iso
```

In these command lines, we could specify the number of cores of CPU and the amount of RAM of Ubuntu VM.

-smp ? (specify the number of cores of CPU)

-m ? (specify the amount of RAM)

At this homework, I used 3 combinations. 1.-smp 4 -m 16G, 2.-smp=4 -m 4G

3. -smp 1 -m 16G

c. After starting Ubuntu VM, we could execute sysbench command lines to test the File I/O in Ubuntu terminal.

For convenience, we could create a bash script to automate the experiments:

\$ nano fileIO_ubuntu.sh

I created a bash script called **fileIO_ubuntu.sh** which contains the below command lines

\$ sysbench fileio --file-total-size=1G --file-test-mode=rndrw prepare

\$ sysbench fileio --file-total-size=1G --file-test-mode=rndrw run

\$ sysbench fileio --file-total-size=1G --file-test-mode=rndrw cleanup

Then we can just execute the below command line to test CPU in Ubuntu terminal

\$ bash fileIO_ubuntu.sh

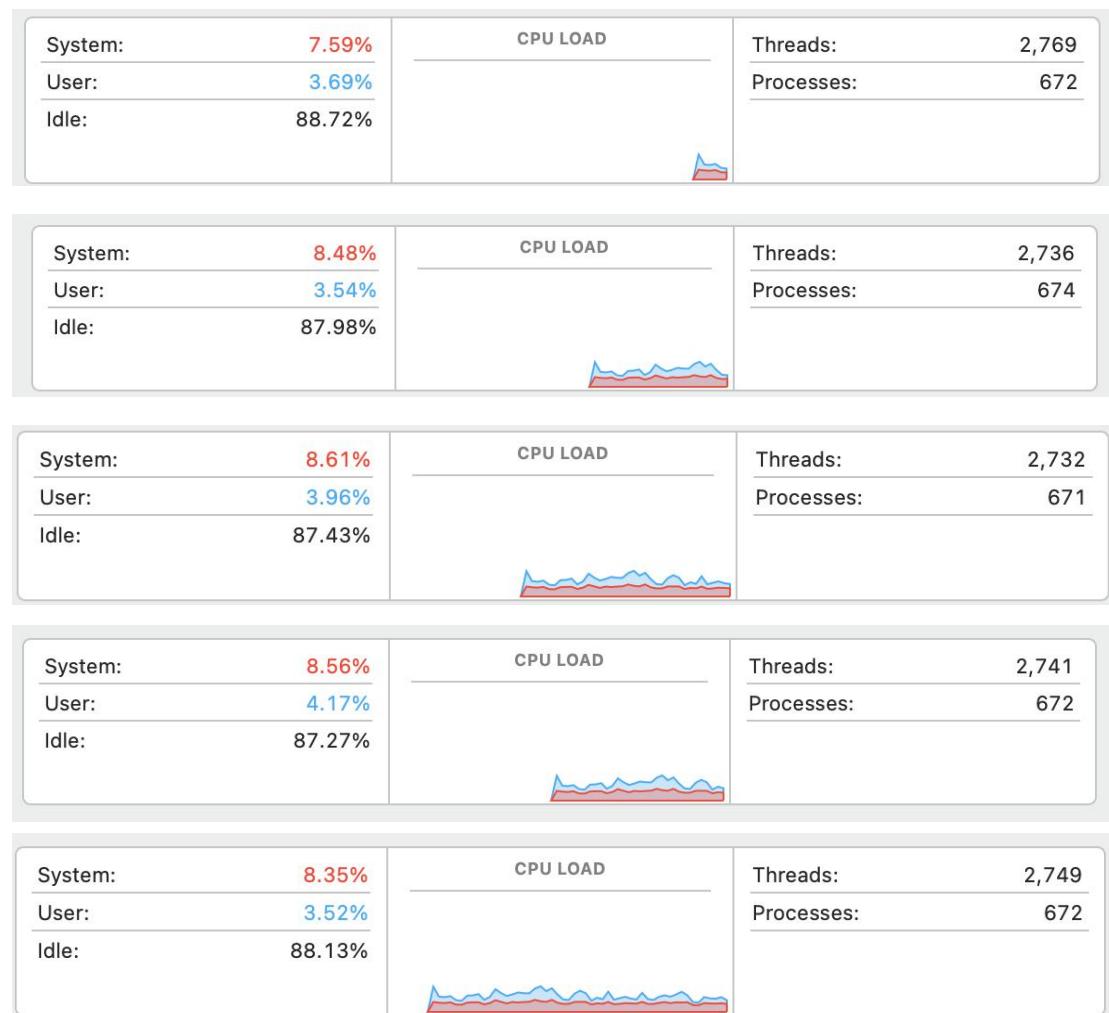
After these steps, we could get the result of CPU Test of QEMU(Ubuntu).

6. Shell scripts for running the experiment: All shell scripts are in my GitHub repository(COEN_241)

7.How I use performance tools to collect performance data:

CPU Test: Using Activity Monitor to observe CPU utilization while testing CPU

Neither Docker nor QEMU is running:

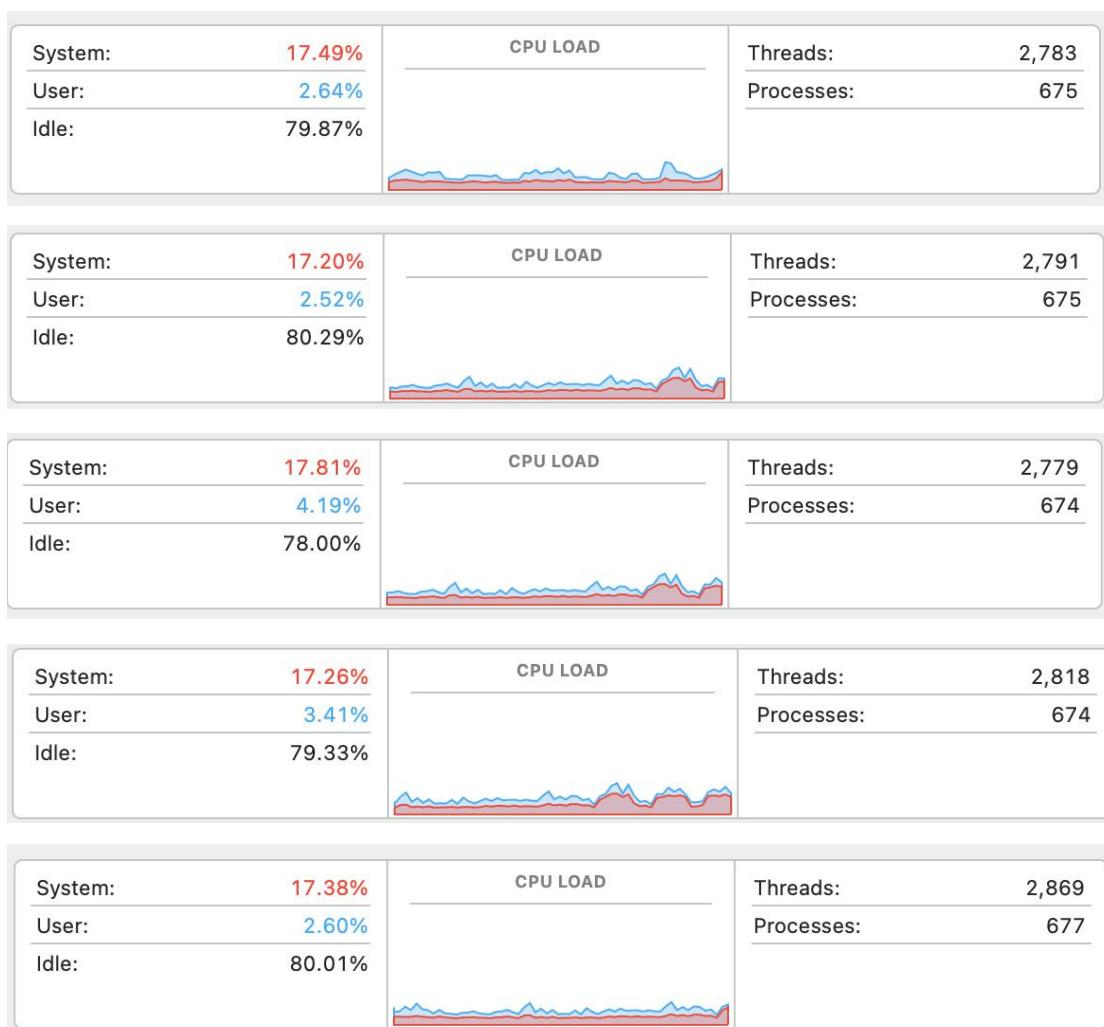


CPU utilization : Avg System :8.32% Avg User:3.78%

When sysbench is testing CPU for Docker:

A.

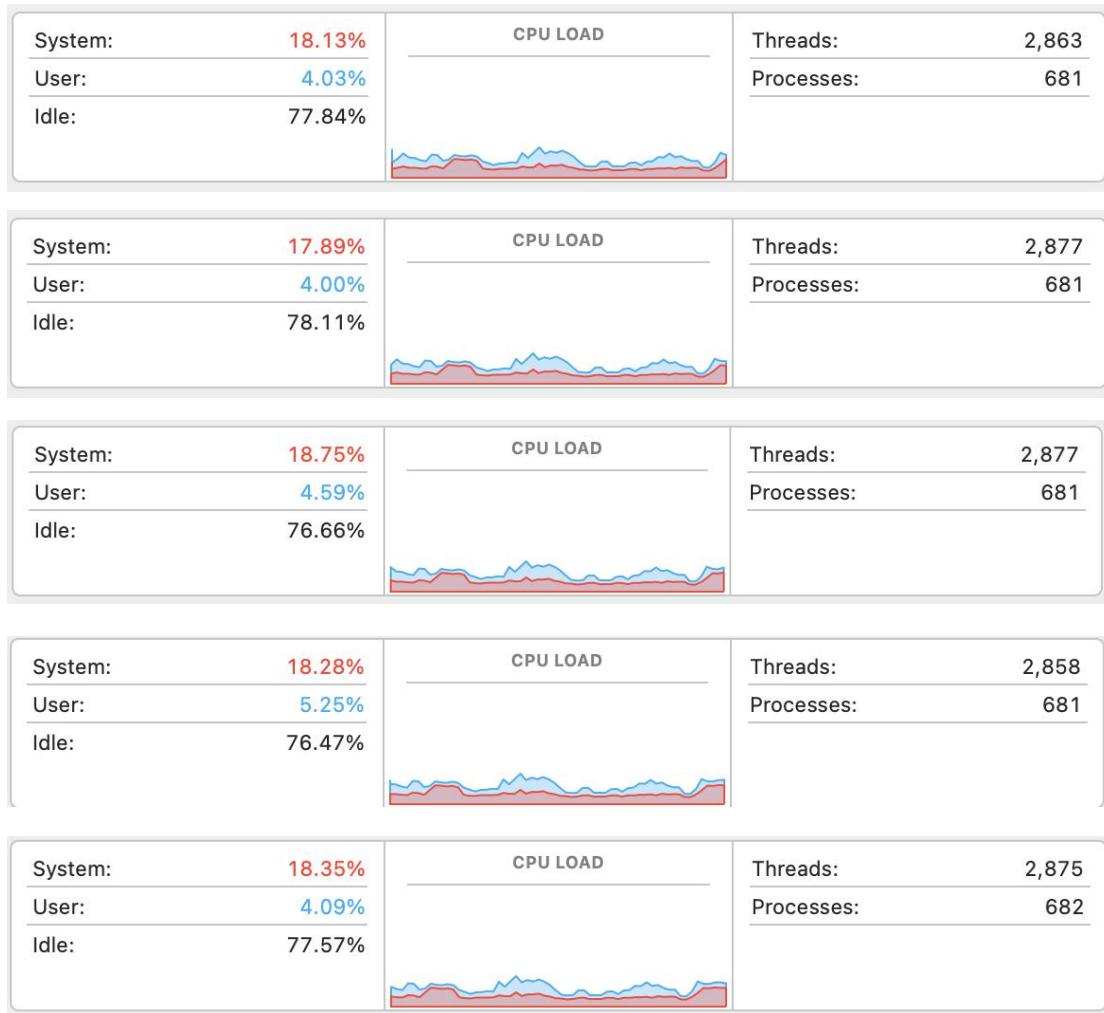
The number of cores of CPU: 4; Memory: 16G



CPU utilization : Avg System :17.43% Avg User:3.07%

B.

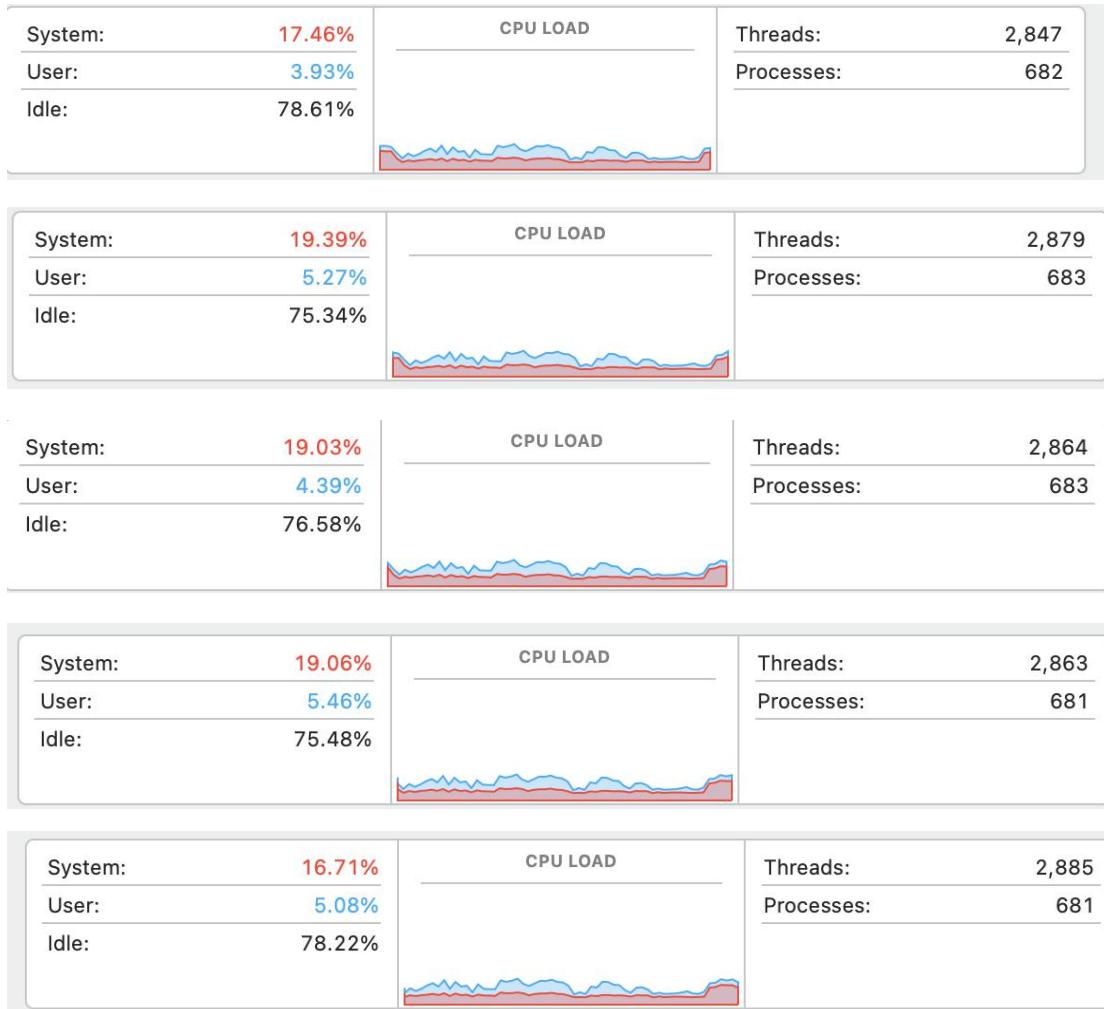
The number of cores of CPU: 4; Memory: 4G



CPU utilization : Avg System :18.28% Avg User:4.45%

C.

The number of cores of CPU: 1; Memory: 16G

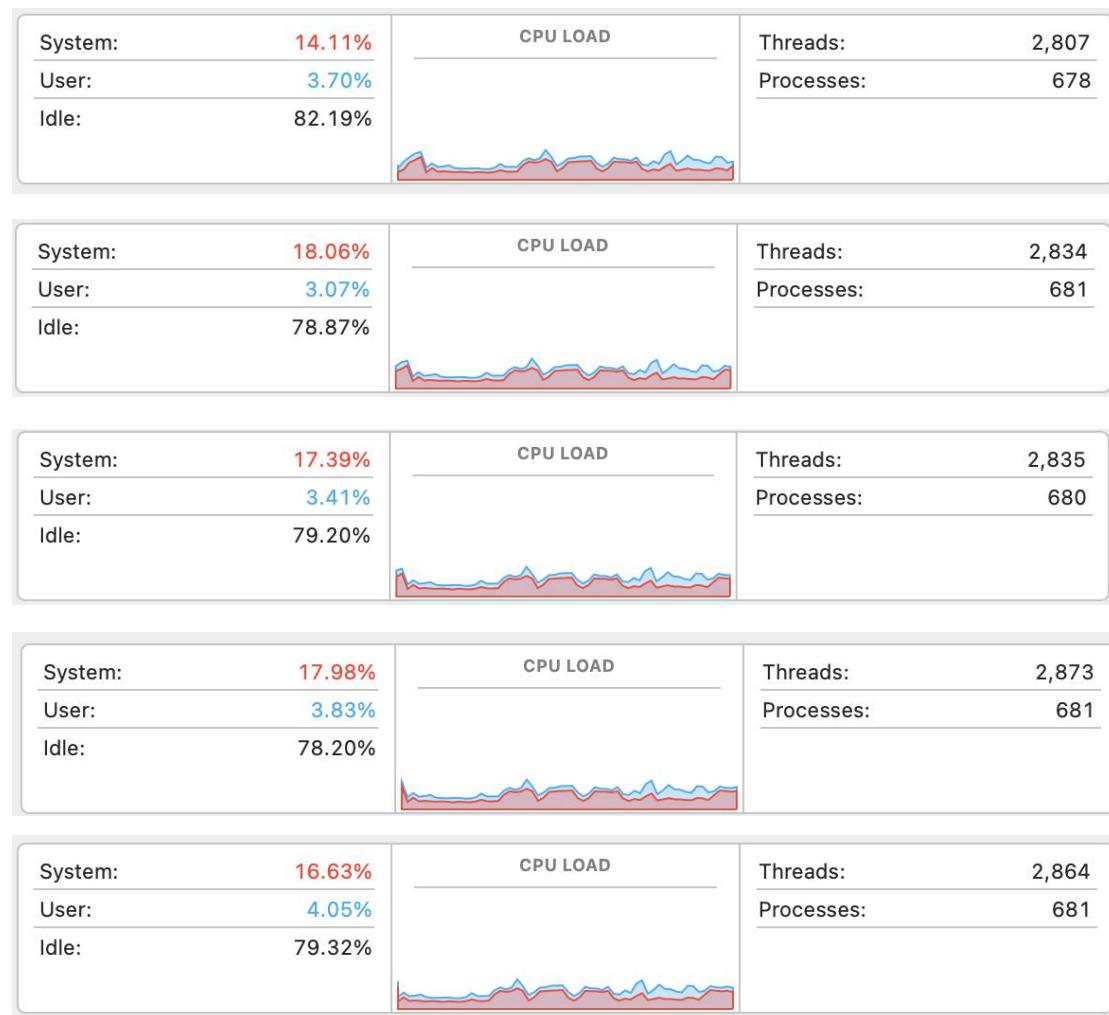


CPU utilization : Avg System :18.33% Avg User:4.83%

When sysbench is testing CPU for QEMU(Ubuntu):

A.

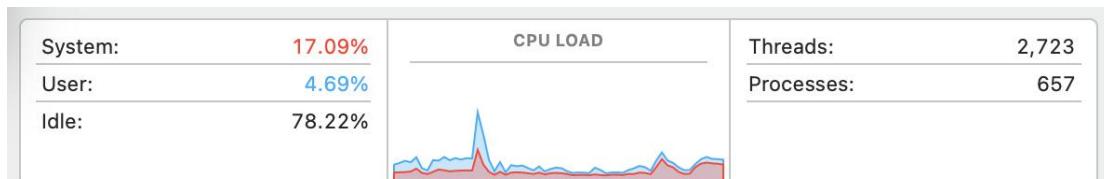
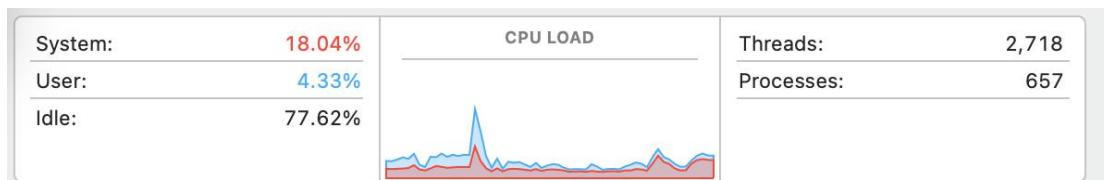
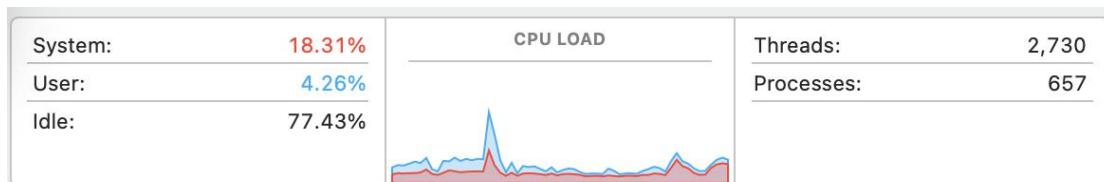
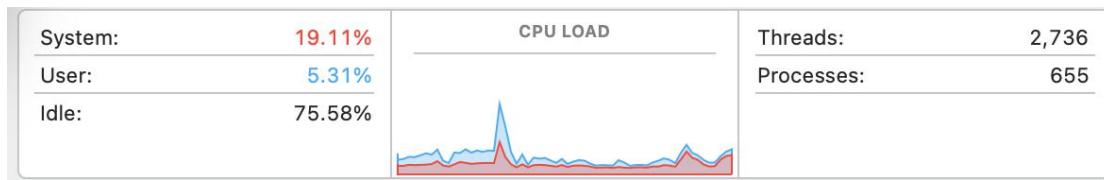
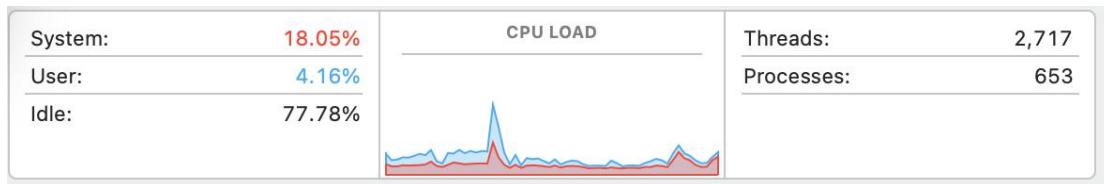
The number of cores of CPU: 4; Memory: 16G



CPU utilization : Avg System :16.83% Avg User:3.61%

B.

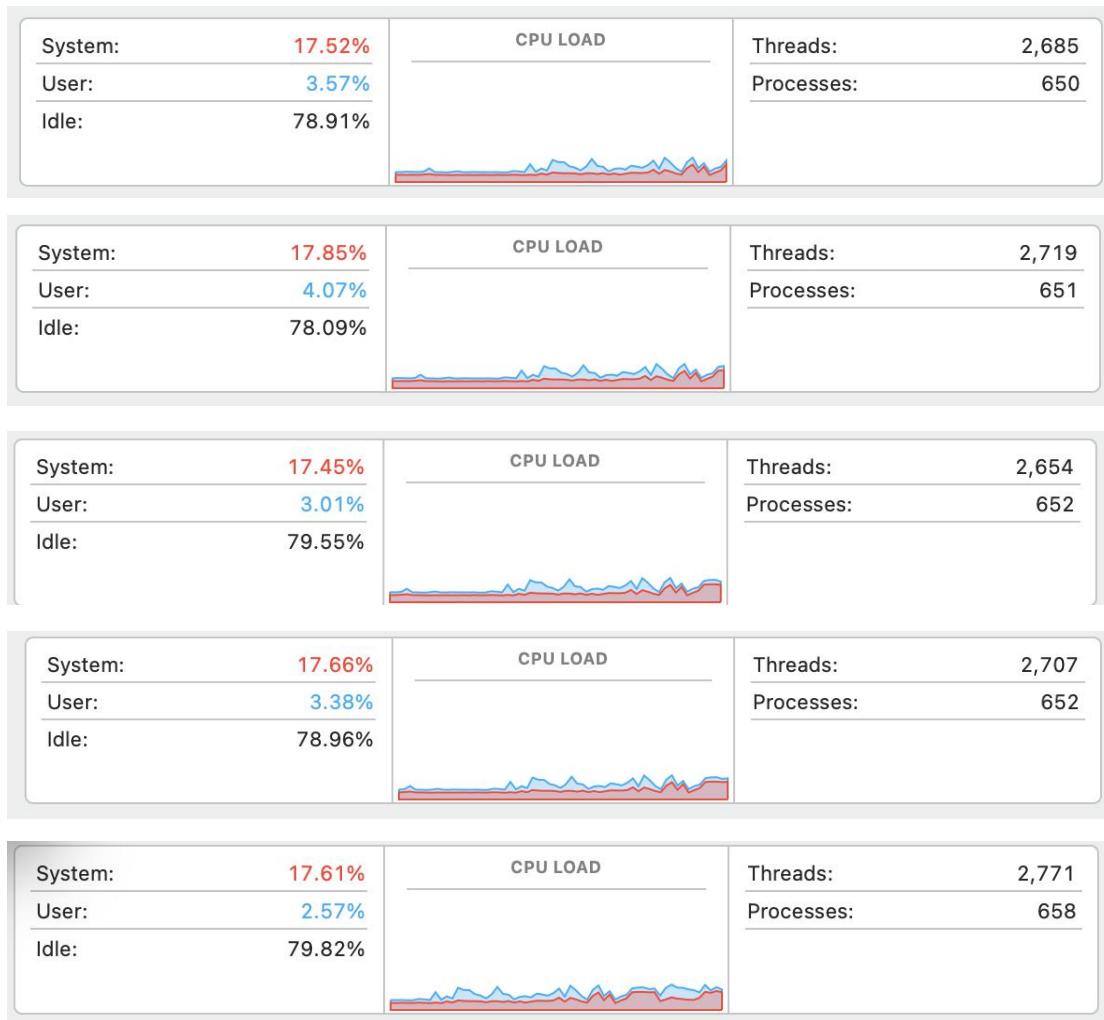
The number of cores of CPU: 4; Memory: 4G



CPU utilization : Avg System :18.12% Avg User:4.55%

C.

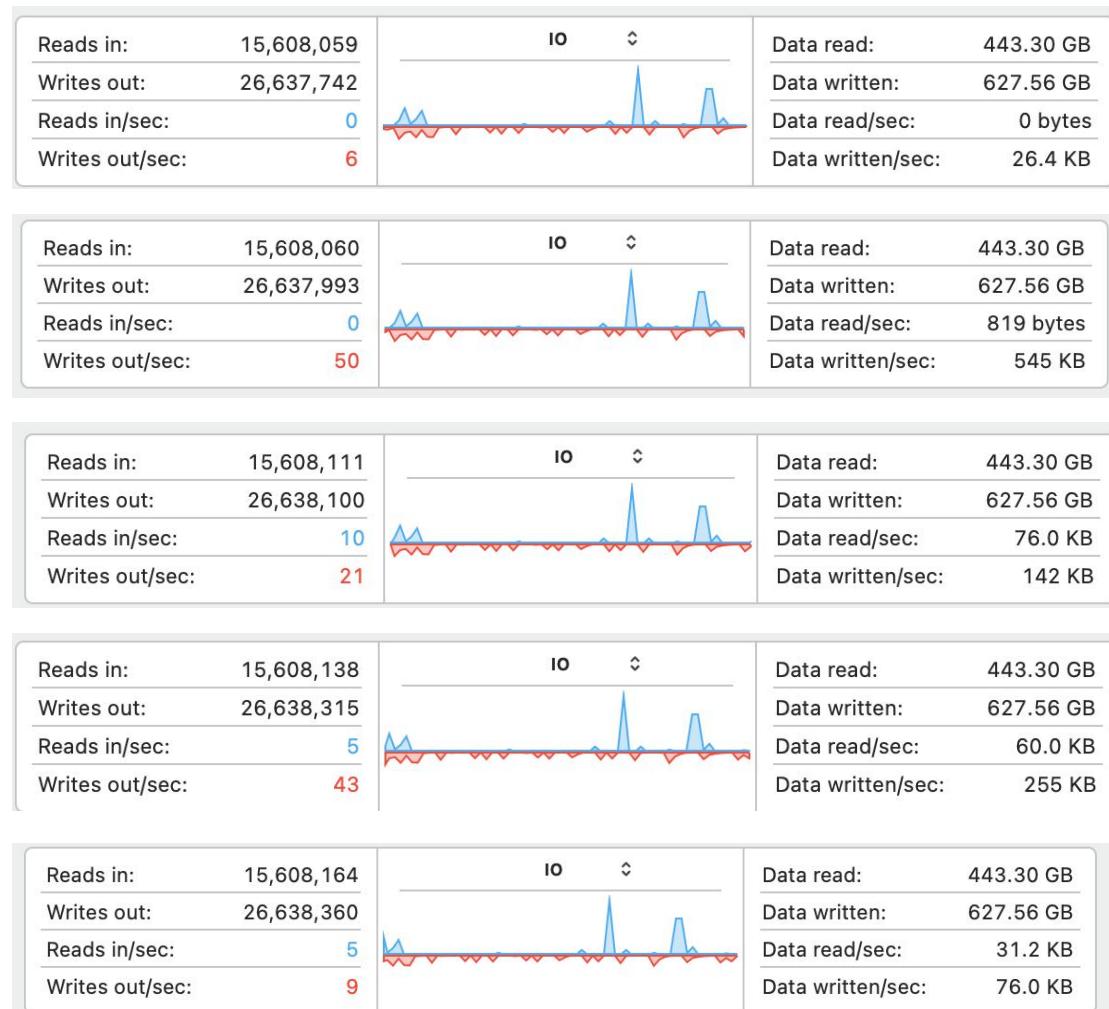
The number of cores of CPU: 1; Memory: 16G



CPU utilization : Avg System :17.62% Avg User:3.32%

File I/O Test: Using Activity Monitor to observe disk I/O while testing File I/O

Neither Docker nor QEMU is running:



Avg Reads in/sec: 4

Avg Writes out/sec: 25.8

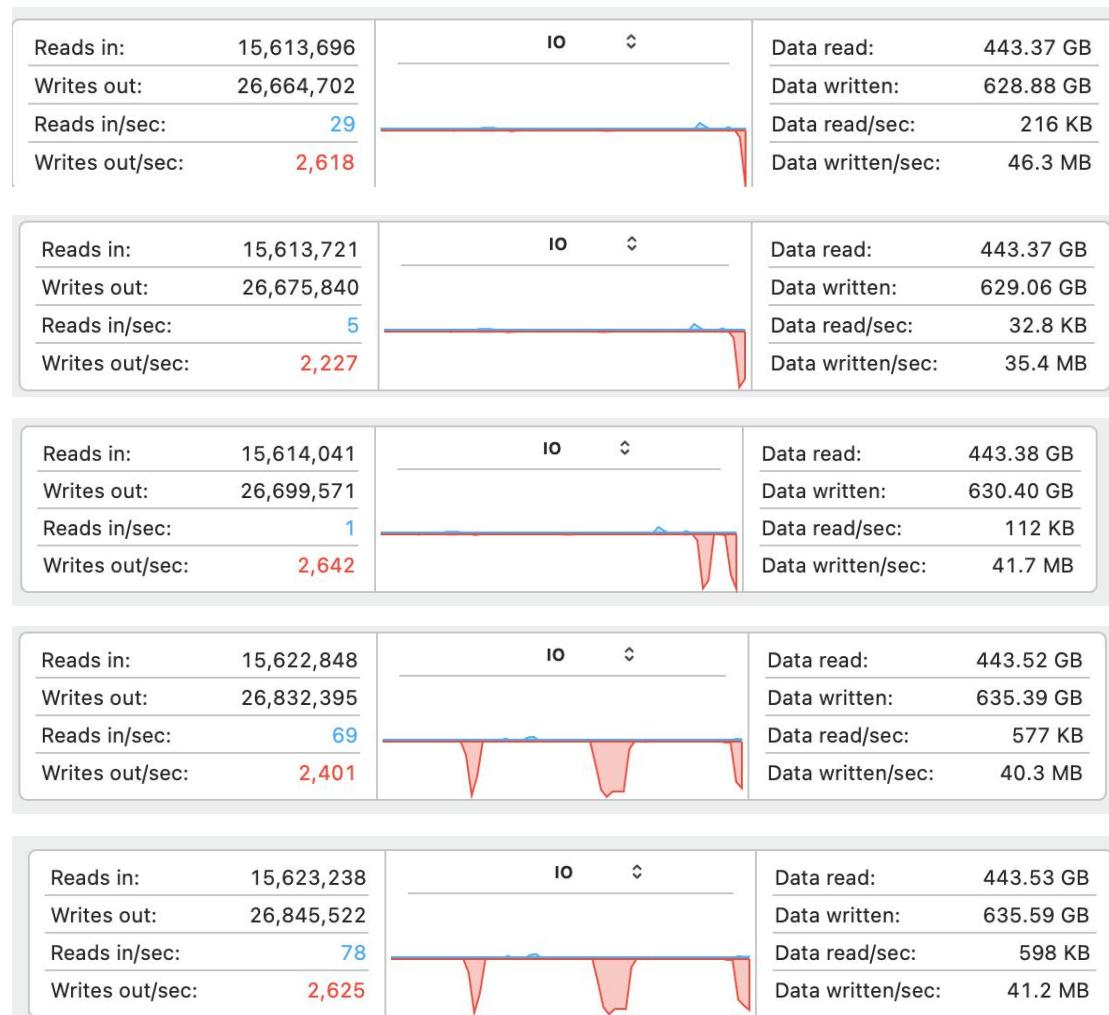
Avg Data read/sec: 21KB

Avg Data written/sec: 208.9KB

When sysbench is testing File I/O for Docker:

A.

The number of cores of CPU: 4; Memory: 16G



Avg Reads in/sec: 36

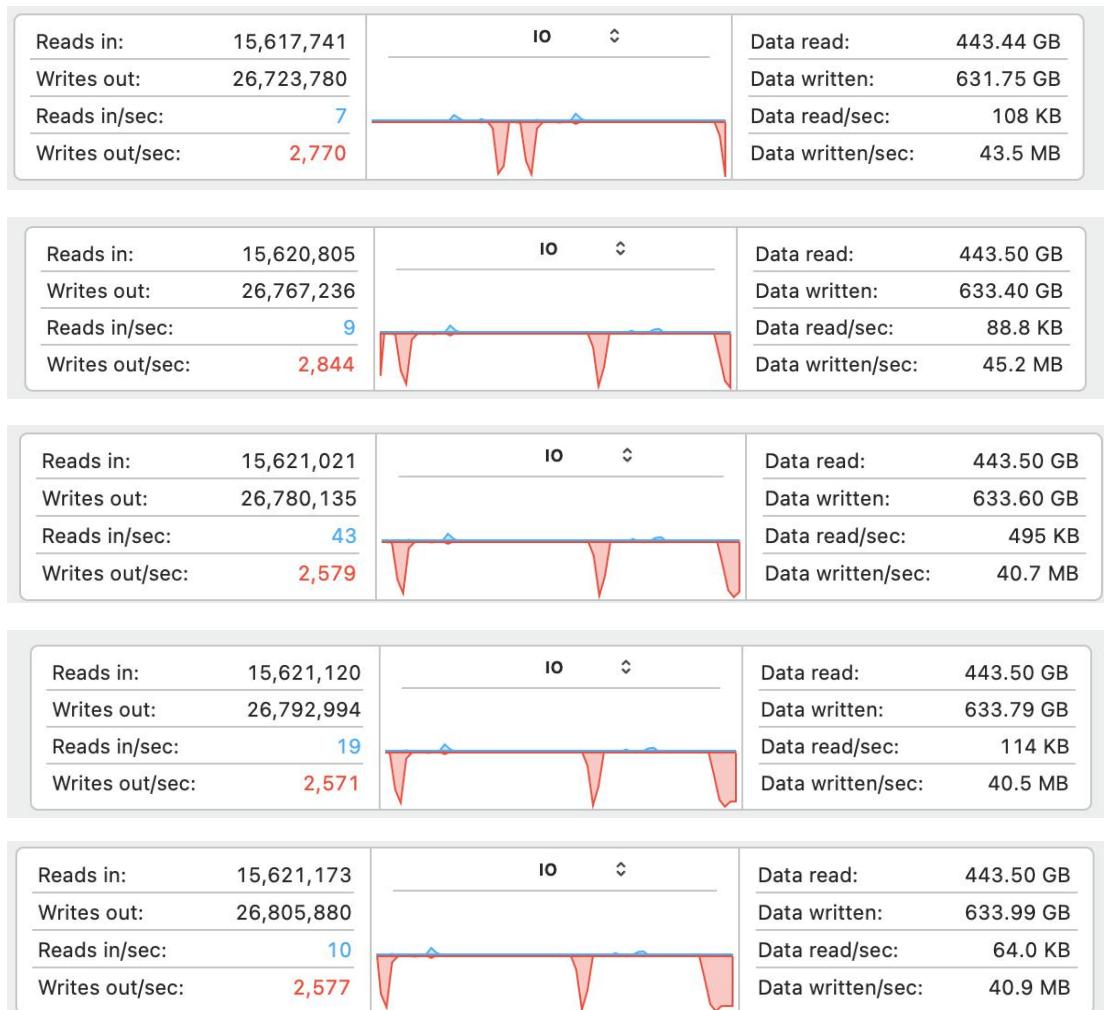
Avg Write Out/sec: 2502

Avg Data read/sec: 307KB

Avg Data written/sec: 41MB

B.

The number of cores of CPU: 4; Memory: 4G



Avg Reads in/sec: 18

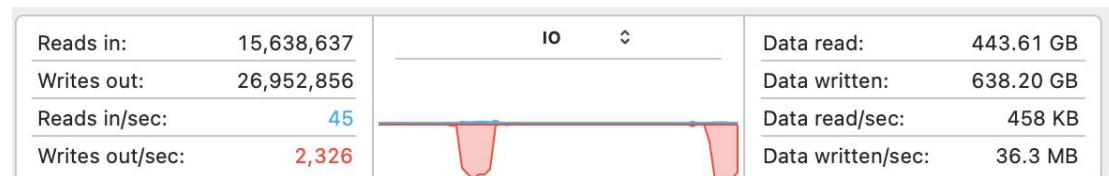
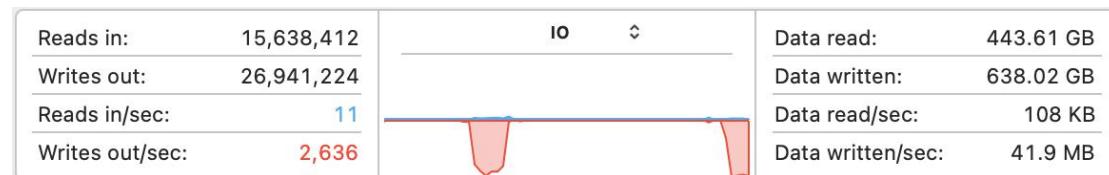
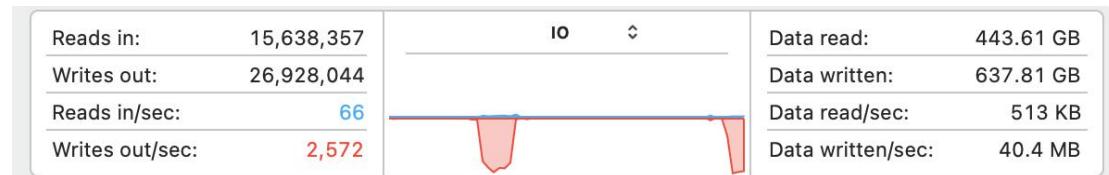
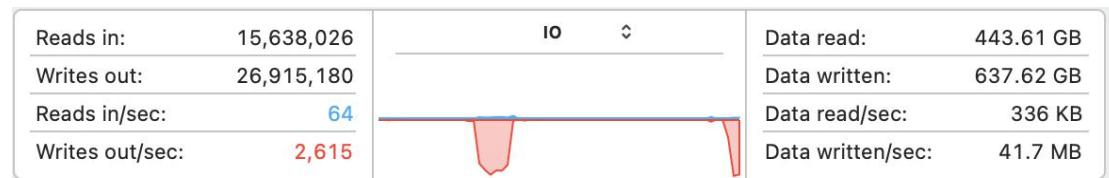
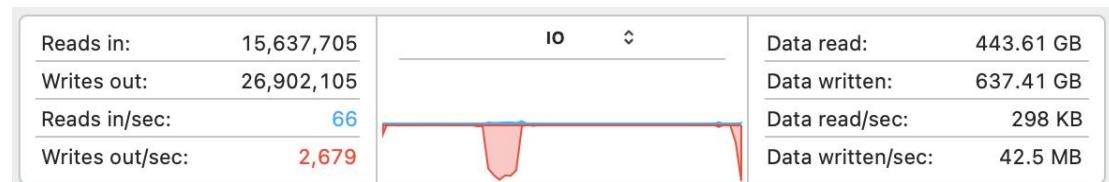
Avg Write out/sec: 2668

Avg Data read/sec: 173.9KB

Avg Data written/sec: 42.2MB

C.

The number of cores of CPU: 1; Memory: 16G



Avg Reads in/sec: 50

Avg Write out/sec: 2565

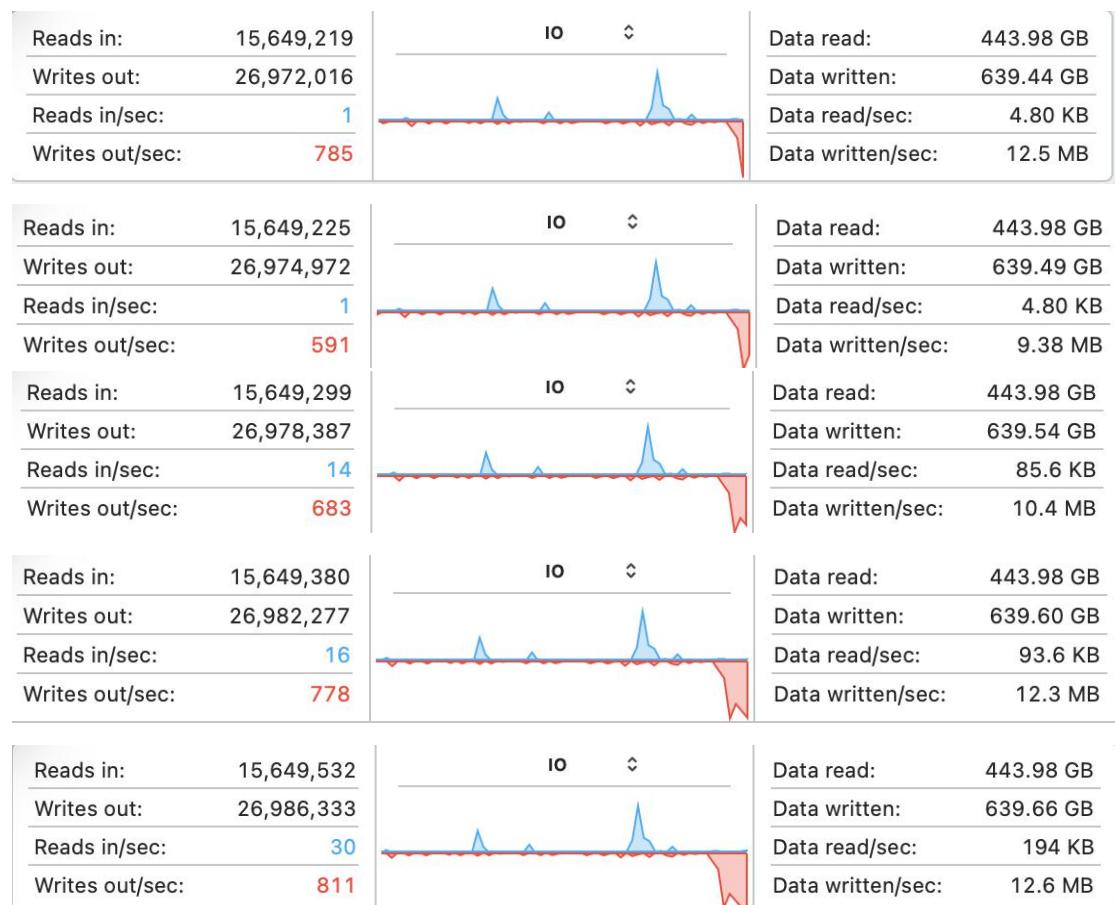
Avg Data read/sec: 343KB

Avg Data written/sec: 40.6MB

When sysbench is testing File I/O for QEMU:

A.

The number of cores of CPU: 4; Memory: 16G



Avg Reads in/sec: 12

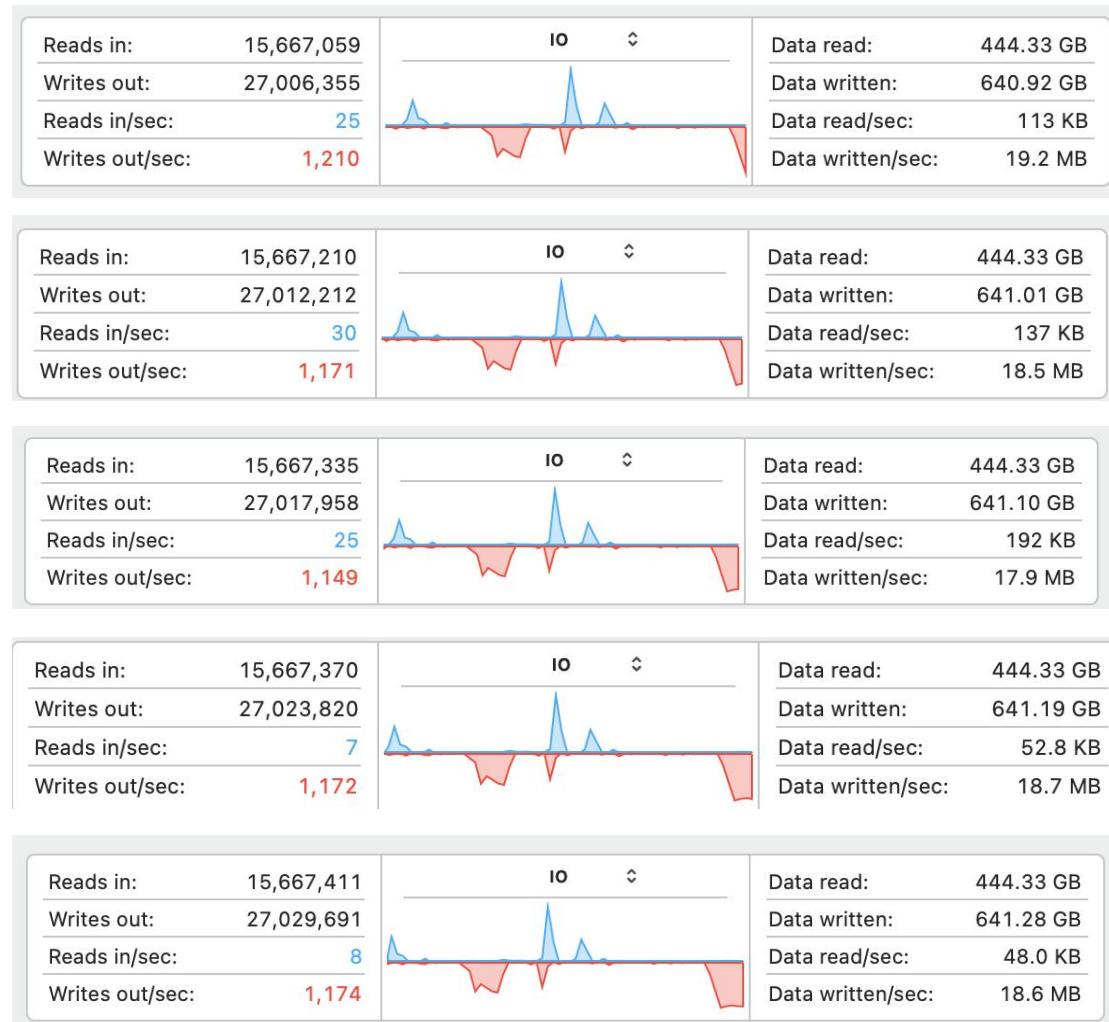
Avg Write out/sec: 886

Avg Data read/sec: 76.6KB

Avg Data written/sec: 11.4MB

B.

The number of cores of CPU: 4; Memory: 4G



Avg Reads in/sec: 19

Avg Write out/sec: 1175

Avg Data read/sec: 108.6KB

Avg Data written/sec: 18.6MB

C.

The number of cores of CPU: 1; Memory: 16G



Avg Reads in/sec: 8

Avg Write out/sec: 2565

Avg Data read/sec: 72.3KB

Avg Data written/sec: 26.2MB

8. Presentation and analysis of the performance data:

CPU Test:

Monitor CPU Utilization(%)	Avg System		Avg User	
	Docker	VM	Docker	VM
Test A	17.43	16.83	3.07	3.61
Test B	18.28	18.12	4.45	4.55
Test C	18.33	17.62	4.83	3.32

A.

CPU = 1, Memory = 16G, threads = 1, max-prime = 1000, time = 30s

CPU Speed(events/s)	Docker		VM	
	Docker	VM	Docker	VM
1	20153.80		29760.84	
2	20813.07		30160.61	
3	20715.77		30453.07	
4	20746.22		29447.38	
5	19940.11		30426.12	

Latency(ms)	Min		Avg		Max		95th percentile	
	Docker	VM	Docker	VM	Docker	VM	Docker	VM
1	0.04	0.03	0.05	0.03	8.10	3.13	0.07	0.05
2	0.04	0.03	0.05	0.03	2.88	4.66	0.07	0.05
3	0.04	0.03	0.05	0.03	1.38	1.17	0.07	0.05
4	0.04	0.03	0.05	0.03	1.51	1.23	0.07	0.05
5	0.04	0.03	0.05	0.03	8.01	2.02	0.08	0.05

B.

CPU = 4, Memory = 16G, threads = 1, max-prime = 1000, time = 30s

CPU Speed(Events/s)	Docker		VM	
	Docker	VM	Docker	VM
1	21017.18		30617.10	
2	21049.62		30205.08	
3	21028.27		30153.61	
4	20749.65		30305.34	
5	20418.23		30542.91	

Latency(ms)	Min		Avg		Max		95th percentile	
	Docker	VM	Docker	VM	Docker	VM	Docker	VM
1	0.04	0.03	0.05	0.03	2.48	1.07	0.07	0.05
2	0.04	0.03	0.05	0.03	4.48	2.12	0.07	0.05
3	0.04	0.03	0.05	0.03	4.50	1.72	0.07	0.05
4	0.04	0.03	0.05	0.03	1.53	9.76	0.07	0.05
5	0.04	0.03	0.05	0.03	4.62	1.00	0.07	0.05

C.

CPU = 4, Memory = 4G, threads = 1, max-prime = 1000, time = 30s

CPU Speed(Events/s)	Docker		VM	
1	20079.11		29401.49	
2	19988.05		30383.38	
3	20796.42		30452.66	
4	20541.21		29885.95	
5	20542.73		30237.21	

Latency(ms)	Min		Avg		Max		95th percentile	
	Docker	VM	Docker	VM	Docker	VM	Docker	VM
1	0.04	0.03	0.05	0.03	3.17	3.11	0.08	0.05
2	0.04	0.03	0.05	0.03	4.36	1.25	0.08	0.05
3	0.04	0.03	0.05	0.03	11.71	2.13	0.07	0.05
4	0.04	0.03	0.05	0.03	1.77	4.43	0.07	0.05
5	0.04	0.03	0.05	0.03	0.56	9.26	0.07	0.05

Analysis of CPU Test:

According to the degree of virtualization, we know that KVM virtual machine uses Hypervisor to virtualize hardware resources. It simulates a virtual CPU, while Docker has less abstraction layers than virtual machines. Docker does not require Hypervisor to virtualize hardware resources. It directly uses the host physical The kernel resources of the machine do not need to be installed as a Guest os. Therefore, when we create a Docker container, we do not need to reload an operating system kernel like a virtual machine. The program on the Docker container directly utilizes the hardware resources of the host physical machine, so Docker has a relatively large advantage in the efficiency of CPU resource usage.

However, according to the table of these three experiments, we found that the CPU performance of the desktop docker is worse than that of QEMU.

I learned from online information that native docker cannot be installed directly on mac OS. native docker can be installed directly on Linux OS. So I think there is a Linux virtual machine built into the desktop docker, and then the native docker is installed and run on the Linux virtual machine. At this condition, the desktop Docker also require Hypervisor to virtualize hardware resources and docker has more abstraction layers than the virtual machine we installed directly on local machine. So the CPU performance tested on docker with sysbench is worse than the performance of the virtual machine we installed directly on the machine.

File I/O Test:

Monitor File I/O	Avg Read(Reads/s)		Avg Write(Writes/s)		Avg Data Read(KB/s)		Avg Written(MB/s)	
	Docker	VM	Docker	VM	Docker	VM	Docker	VM
Test A	36	12	2502	886	307	76.6	41	11.4
Test B	18	19	2668	1175	173.9	108.6	42.2	18.6
Test C	50	8	2565	2565	343	72.3	40.6	26.2

A. CPU = 4, Memory = 16G, file-total-size=1G file-test-mode=rndrw, time = 10s(default)

File Operations	Reads/S		Writes/S		Fsyncs/S	
	Docker	VM	Docker	VM	Docker	VM
1	3492.15	1817.69	2328.10	1211.79	7451.41	3877.84
2	3450.96	1494.63	2300.64	996.42	7372.83	3197.11
3	3414.70	1597.09	2276.47	1064.73	7289.79	3417.60
4	3420.28	1519.31	2280.18	1032.87	7299.59	3314.86
5	3401.18	1626.66	2267.45	1084.44	7257.94	3474.98

Throughput	Read(MiB/s)		Written(MiB/s)	
	Docker	VM	Docker	VM
1	54.56	28.40	36.38	18.93
2	53.92	23.35	35.95	15.57
3	53.35	24.95	35.57	16.64
4	53.44	24.21	35.63	16.14
5	53.14	25.42	35.43	16.94

Latency(ms)	Min		Avg		Max		95th percentile	
	Docker	VM	Docker	VM	Docker	VM	Docker	VM
1	0.00	0.00	0.07	0.14	216.31	27.69	0.27	0.52
2	0.00	0.00	0.08	0.17	2.46	54.25	0.27	0.55
3	0.00	0.00	0.08	0.16	43.67	7.10	0.28	0.54
4	0.00	0.00	0.08	0.17	7.90	38.71	0.27	0.54
5	0.00	0.00	0.08	0.16	103.90	30.77	0.27	0.53

B. CPU = 4, Memory = 4G, file-total-size=1G file-test-mode=rndrw, time = 10s(default)

File Operations	Reads/S		Writes/S		Fsyncs/S	
	Docker	VM	Docker	VM	Docker	VM
1	3468.22	1879.87	2312.15	1253.25	7401.76	4017.98
2	2982.73	1386.88	1988.49	924.58	6364.46	2963.85
3	3102.49	1422.20	2068.33	948.13	6618.95	3039.90
4	3126.74	1488.40	2084.49	992.26	6673.77	3178.83
5	3317.62	1387.95	2211.75	925.30	7077.70	2967.04

Throughput	Read(MiB/s)		Written(MiB/s)	
	Docker	VM	Docker	VM
1	54.19	29.37	36.13	19.58
2	46.61	21.67	31.07	14.45
3	48.48	22.22	32.32	14.81
4	48.86	23.26	32.57	15.50
5	51.84	21.69	34.56	14.46

Latency(ms)	Min		Avg		Max		95th percentile	
	Docker	VM	Docker	VM	Docker	VM	Docker	VM
1	0.00	0.00	0.08	0.14	88.10	37.94	0.27	0.49
2	0.00	0.00	0.09	0.19	112.48	47.98	0.30	0.58
3	0.00	0.00	0.08	0.18	116.07	6.13	0.30	0.59
4	0.00	0.00	0.08	0.18	160.32	7.15	0.29	0.56
5	0.00	0.00	0.08	0.19	7.09	7.59	0.29	0.61

C. CPU = 1, Memory = 16G, file-total-size=1G file-test-mode=rndrw, time = 10s(default)

File Operations	Reads/S		Writes/S		Fsyncs/S	
	Docker	VM	Docker	VM	Docker	VM
1	3228.67	2179.00	2152.44	1452.67	6891.32	4656.62
2	3537.78	1896.87	2358.52	1264.58	7555.25	4059.11
3	3283.71	2128.24	2189.14	1418.83	7016.53	4547.23
4	3372.26	2076.17	2248.17	1384.12	7198.94	4431.06
5	3272.15	2059.03	2181.43	1372.68	6992.27	4398.08

Throughput	Read(MiB/s)		Written(MiB/s)	
	Docker	VM	Docker	VM
1	50.45	34.05	33.63	22.70
2	55.28	29.64	36.85	19.76
3	51.31	33.25	34.21	22.17
4	52.69	32.44	35.31	21.63
5	51.13	32.17	34.08	21.45

Latency(ms)	Min		Avg		Max		95th percentile	
	Docker	VM	Docker	VM	Docker	VM	Docker	VM
1	0.00	0.00	0.08	0.12	375.34	31.06	0.27	0.44
2	0.00	0.00	0.07	0.14	2.18	37.39	0.27	0.47
3	0.00	0.00	0.08	0.12	23.25	7.11	0.29	0.44
4	0.00	0.00	0.08	0.13	191.67	7.07	0.27	0.46
5	0.00	0.00	0.08	0.13	6.22	14.77	0.28	0.46

Analysis of File I/O Test:

According to the data obtained from these three experiments, the disk I/O in the KVM virtualization environment is significantly lower than the Docker container disk I/O.

According to the analysis of the test results, in fact, the disk of the virtual machine is essentially just a "large file" of the hard disk of the physical host. Currently, there are generally two images in raw format and qcow2 format. Under the condition of full virtualization, when reading and writing disk data, the driver of the I/O device needs to switch between the two different operating modes of the physical machine CPU and the virtual machine CPU, resulting in a large degree of resource consumption. Therefore, in a virtualized environment, the disk I/O efficiency of virtual machines is low. However, the Docker container reads and writes files through Volume. Volume itself is a directory or file. It can bypass the default joint file system and exist on the host in the form of a normal file or directory. Therefore, the Docker container disk performance is better than virtual Machine performance is much higher.