COEN 241 HW1

Shivani Gopal Deosatwar

W1621005

1. QEMU

   Installation

   1) Download ISO
      For assignments, Ubuntu guest VM is needed which requires server ISO image.
      I have used following link:
      ARM (Apple Silicon): Ubuntu 20.04 Server for ARM
   2) As I am using MacBook Air M1 chip, following commands worked for installing QEMU on my laptop.
      brew install qemu
   3) To create QEMU image, following command is used:
      qemu-img create ubuntu.img 20G -f qcow2
   4) VM installation:
      qemu-system-aarch64 \
      -accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2G \
      -smp 2 \
      -drive
      file=/opt/homebrew/Cellar/qemu/6.2.0_1/share/qemu/edk2-aarch64-
      code.fd,if=pflash,format=raw,readonly=on \
      -drive if=none,file=ubuntu.img,format=qcow2,id=hd0 \
      -device virtio-blk-device,drive=hd0,serial="trial_2" \
      -device virtio-net-device,netdev=net0 \
      -netdev user,id=net0 \
      -vga none -device ramfb \
      -cdrom ubuntu-20.04.4-live-server-arm64.iso \

-device usb-ehci -device usb-kbd -device usb-mouse -usb
-nographic
Instructions will be displayed on the screen and which needs to be followed to install the VM successfully.

5) Run image:
qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2G \
-smp 2 \
-drive
file=/opt/homebrew/Cellar/qemu/6.2.0_1/share/qemu/edk2-aarch64-
code.fd,if=pflash,format=raw,readonly=on \
-drive if=none,file=ubuntu.img,format=qcow2,id=hd0 \
-device virtio-blk-device,drive=hd0,serial="trial_2" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-device usb-ehci -device usb-kbd -device usb-mouse -usb
-nographic

2. Experimental setup

System configurations are:

MacBook Air (M1, 2020)

M1 chip

8 GB memory

I am considering following 3 test cases for experimenting:

1) 2cores with 2GB memory allocation
2) 4cores with 4GB memory allocation
3) 6 cores with 6GB memory allocation

Same has been used for Docker as well.

3. Docker
   Docker is used to create and manage containers to run and maintain different images.

   Installation

   1) Rosetta 2 is required to be installed for docker.
      softwareupdate-install-rosetta
   2) Install docker engine
      Copy image from here.
   3) Check if installed successfully.
   4) Docker run hello-world

   Output

```
preetibhosale@Preetis-MacBook-Air ~ % docker run hello-world
zsh: command not found: docker
preetibhosale@Preetis-MacBook-Air ~ % docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
7050e35b49f5: Pull complete
Digest: sha256:18a657d0cc1c7d0678a3fbea8b7eb4918bba25968d3e1b0adebfa71caddbc346
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (arm64v8)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

4. Experiments- Reports and findings

Configuration 1: 2GB RAM and 2 cores

Tests:

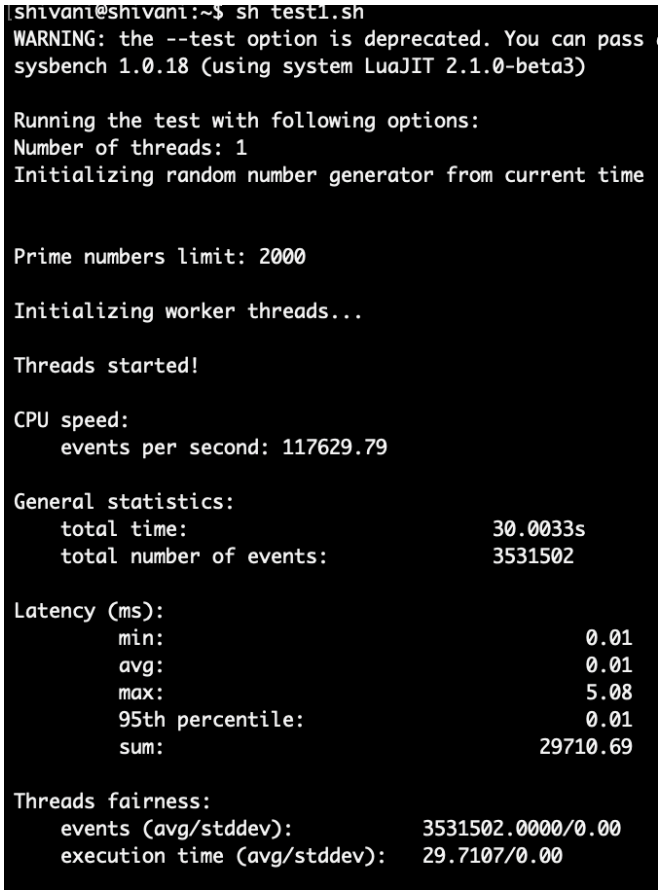
1) Max-prime=2000, time=30 seconds
2) Max-prime=20000, time=30 seconds
3) Max-prime=200000, time=30 seconds

Sysbench command:

sysbench cpu --cpu-max-prime={some_value} --num-threads={some_value} --time= {some_value} run

QEMU results:

Test 1:



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


Prime numbers limit: 2000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 115037.16

General statistics:
    total time:                          30.0113s
    total number of events:              3458529

Latency (ms):
         min:                                    0.01
         avg:                                    0.01
         max:                                  132.41
         95th percentile:                        0.01
         sum:                                29714.81

Threads fairness:
    events (avg/stddev):           3458529.0000/0.00
    execution time (avg/stddev):   29.7148/0.00
```

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


Prime numbers limit: 2000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 115160.19

General statistics:
    total time:                          30.0204s
    total number of events:              3459362

Latency (ms):
         min:                                    0.01
         avg:                                    0.01
         max:                                   16.17
         95th percentile:                        0.01
         sum:                                29501.63

Threads fairness:
    events (avg/stddev):           3459362.0000/0.00
    execution time (avg/stddev):   29.5016/0.00
```

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


Prime numbers limit: 2000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 110559.19

General statistics:
    total time:                          30.0076s
    total number of events:              3318943

Latency (ms):
         min:                                    0.01
         avg:                                    0.01
         max:                                   36.13
         95th percentile:                        0.01
         sum:                                29494.88

Threads fairness:
    events (avg/stddev):           3318943.0000/0.00
    execution time (avg/stddev):   29.4949/0.00
```

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


Prime numbers limit: 2000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 112912.98

General statistics:
    total time:                          30.0164s
    total number of events:              3390114

Latency (ms):
         min:                                    0.01
         avg:                                    0.01
         max:                                   24.70
         95th percentile:                        0.01
         sum:                                29660.42

Threads fairness:
    events (avg/stddev):           3390114.0000/0.00
    execution time (avg/stddev):   29.6604/0.00

shivani@shivani:~$
```

Iteration	events/sec
1	117629.79- max value
2	115037.16
3	115160.19
4	110559.19- min value
5	112912.98
average	114259.86

Test 2:

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


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  4301.90

General statistics:
    total time:                          30.0070s
    total number of events:              129154

Latency (ms):
         min:                                    0.23
         avg:                                    0.23
         max:                                    7.74
         95th percentile:                        0.24
         sum:                                29968.78

Threads fairness:
    events (avg/stddev):           129154.0000/0.00
    execution time (avg/stddev):   29.9688/0.00
```

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


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  4319.10

General statistics:
    total time:                          30.0002s
    total number of events:              129581

Latency (ms):
         min:                                    0.23
         avg:                                    0.23
         max:                                    2.00
         95th percentile:                        0.24
         sum:                                29949.83

Threads fairness:
    events (avg/stddev):           129581.0000/0.00
    execution time (avg/stddev):   29.9498/0.00
```

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


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  4326.48

General statistics:
    total time:                          30.0005s
    total number of events:              129821

Latency (ms):
         min:                                    0.23
         avg:                                    0.23
         max:                                    9.91
         95th percentile:                        0.24
         sum:                                29973.12

Threads fairness:
    events (avg/stddev):           129821.0000/0.00
    execution time (avg/stddev):   29.9731/0.00
```

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


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  4331.81

General statistics:
    total time:                          30.0013s
    total number of events:              129987

Latency (ms):
         min:                                    0.23
         avg:                                    0.23
         max:                                    0.84
         95th percentile:                        0.24
         sum:                                29970.90

Threads fairness:
    events (avg/stddev):           129987.0000/0.00
    execution time (avg/stddev):   29.9709/0.00
```

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


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  4322.40

General statistics:
    total time:                          30.0004s
    total number of events:              129683

Latency (ms):
         min:                                    0.23
         avg:                                    0.23
         max:                                    1.10
         95th percentile:                        0.24
         sum:                                29968.28

Threads fairness:
    events (avg/stddev):           129683.0000/0.00
    execution time (avg/stddev):   29.9683/0.00

shivani@shivani:~$
```

Iteration	events/sec
1	4301.90 -min value
2	4319.10
3	4326.48
4	4331.81- max value
5	4322.40
average	4320.33

Test 3:

```
shivani@shivani:~$ sh test3.sh
WARNING: the --test option is deprecated. You can pass a
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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


Prime numbers limit: 200000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   189.83

General statistics:
    total time:                          30.0041s
    total number of events:              5698

Latency (ms):
         min:                                    5.18
         avg:                                    5.26
         max:                                    7.48
         95th percentile:                        5.37
         sum:                                29957.30

Threads fairness:
    events (avg/stddev):           5698.0000/0.00
    execution time (avg/stddev):   29.9573/0.00
```

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


Prime numbers limit: 200000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   190.20

General statistics:
    total time:                          30.0087s
    total number of events:              5710

Latency (ms):
         min:                                    5.19
         avg:                                    5.25
         max:                                    7.94
         95th percentile:                        5.28
         sum:                                29968.41

Threads fairness:
    events (avg/stddev):           5710.0000/0.00
    execution time (avg/stddev):   29.9684/0.00
```

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


Prime numbers limit: 200000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   190.30

General statistics:
    total time:                          30.0040s
    total number of events:              5712

Latency (ms):
         min:                                    5.18
         avg:                                    5.25
         max:                                    8.63
         95th percentile:                        5.37
         sum:                                29973.66

Threads fairness:
    events (avg/stddev):           5712.0000/0.00
    execution time (avg/stddev):   29.9737/0.00
```

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


Prime numbers limit: 200000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   190.59

General statistics:
    total time:                          30.0005s
    total number of events:              5718

Latency (ms):
         min:                                    5.18
         avg:                                    5.24
         max:                                   16.02
         95th percentile:                        5.28
         sum:                                29972.36

Threads fairness:
    events (avg/stddev):           5718.0000/0.00
    execution time (avg/stddev):   29.9724/0.00
```

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


Prime numbers limit: 200000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   190.00

General statistics:
    total time:                          30.0056s
    total number of events:              5703

Latency (ms):
         min:                                    5.19
         avg:                                    5.26
         max:                                    9.22
         95th percentile:                        5.37
         sum:                                29974.24

Threads fairness:
    events (avg/stddev):           5703.0000/0.00
    execution time (avg/stddev):   29.9742/0.00

shivani@shivani:~$
```

Iteration	events/sec
1	189.90 - min value
2	190.20
3	190.30
4	190.59 - max value
5	190.00
average	190.20

Docker results:

Test 1:

```
WARNING: the --test option is deprecated. You can pass a
sysbench 1.0.20-f6f6117dc4 (using bundled LuaJIT 2.1.0-b

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


Prime numbers limit: 2000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 55705.48

General statistics:
    total time:                          30.0061s
    total number of events:              1671610

Latency (ms):
         min:                                    0.02
         avg:                                    0.02
         max:                                    6.59
         95th percentile:                        0.02
         sum:                                29847.15

Threads fairness:
    events (avg/stddev):           1671610.0000/0.00
    execution time (avg/stddev):   29.8471/0.00
```

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


Prime numbers limit: 2000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 53339.24

General statistics:
    total time:                          30.0035s
    total number of events:              1600418

Latency (ms):
         min:                                    0.02
         avg:                                    0.02
         max:                                  647.49
         95th percentile:                        0.02
         sum:                                29799.64

Threads fairness:
    events (avg/stddev):           1600418.0000/0.00
    execution time (avg/stddev):   29.7996/0.00

sysbench 1.0.20-f6f6117dc4 (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: 2000

Initializing worker threads...

Threads started!

WARNING: the --test option is deprecated. You can pass
CPU speed:
    events per second: 53770.33

General statistics:
    total time:                          30.0019s
    total number of events:              1613486

Latency (ms):
         min:                                    0.02
         avg:                                    0.02
         max:                                   75.10
         95th percentile:                        0.02
         sum:                                29750.40

Threads fairness:
    events (avg/stddev):           1613486.0000/0.00
    execution time (avg/stddev):   29.7504/0.00
```

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


Prime numbers limit: 2000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 54454.38

General statistics:
    total time:                          30.0011s
    total number of events:              1633769

Latency (ms):
         min:                                    0.02
         avg:                                    0.02
         max:                                   19.96
         95th percentile:                        0.02
         sum:                                29839.23

Threads fairness:
    events (avg/stddev):           1633769.0000/0.00
    execution time (avg/stddev):   29.8392/0.00

sysbench 1.0.20-f6f6117dc4 (using bundled LuaJIT 2.1.0-b
```

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


Prime numbers limit: 2000

Initializing worker threads...

Threads started!

WARNING: the --test option is deprecated. You can pass
CPU speed:
    events per second: 55029.96

General statistics:
    total time:                          30.0011s
    total number of events:              1651037

Latency (ms):
         min:                                    0.02
         avg:                                    0.02
         max:                                    3.30
         95th percentile:                        0.02
         sum:                                29848.91

Threads fairness:
    events (avg/stddev):           1651037.0000/0.00
    execution time (avg/stddev):   29.8489/0.00
```

Iteration	events/sec
1	55705.48 - max value
2	53339.24 - min value
3	53770.33
4	54454.38
5	55029.96
average	54459.88

Test 2:

```
WARNING: the --test option is deprecated. You can pass a
sysbench 1.0.20-f6f6117dc4 (using bundled LuaJIT 2.1.0-b

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


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  3203.72

General statistics:
    total time:                          30.0012s
    total number of events:              96123

Latency (ms):
         min:                                    0.30
         avg:                                    0.31
         max:                                    9.90
         95th percentile:                        0.33
         sum:                                29927.79

Threads fairness:
    events (avg/stddev):           96123.0000/0.00
    execution time (avg/stddev):   29.9278/0.00
```

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


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  3190.15

General statistics:
    total time:                          30.0207s
    total number of events:              95901

Latency (ms):
         min:                                    0.30
         avg:                                    0.31
         max:                                    8.36
         95th percentile:                        0.33
         sum:                                29967.28

Threads fairness:
    events (avg/stddev):           95901.0000/0.00
    execution time (avg/stddev):   29.9673/0.00

WARNING: the --test option is deprecated. You can pass
sysbench 1.0.20-f6f6117dc4 (using bundled LuaJIT 2.1.0-
```

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


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  3207.99

General statistics:
    total time:                          30.0007s
    total number of events:              96245

Latency (ms):
         min:                                    0.30
         avg:                                    0.31
         max:                                    4.64
         95th percentile:                        0.32
         sum:                                29963.17

Threads fairness:
    events (avg/stddev):           96245.0000/0.00
    execution time (avg/stddev):   29.9632/0.00

WARNING: the --test option is deprecated. You can pass
sysbench 1.0.20-f6f6117dc4 (using bundled LuaJIT 2.1.0-
```

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


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  3197.21

General statistics:
    total time:                          30.0010s
    total number of events:              95924

Latency (ms):
         min:                                    0.30
         avg:                                    0.31
         max:                                    1.89
         95th percentile:                        0.32
         sum:                                29963.23

Threads fairness:
    events (avg/stddev):           95924.0000/0.00
    execution time (avg/stddev):   29.9632/0.00

sysbench 1.0.20-f6f6117dc4 (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: 20000

Initializing worker threads...

WARNING: the --test option is deprecated. You can pass
Threads started!

CPU speed:
    events per second:  3194.72

General statistics:
    total time:                          30.0029s
    total number of events:              95854

Latency (ms):
         min:                                    0.30
         avg:                                    0.31
         max:                                    3.43
         95th percentile:                        0.32
         sum:                                29956.37

Threads fairness:
    events (avg/stddev):           95854.0000/0.00
    execution time (avg/stddev):   29.9564/0.00
```

Iteration	events/sec
1	3203.72
2	3190.15 - min value
3	3207.99 - max value
4	3197.21
5	3194.72
average	3198.75

Test 3:

```
sysbench 1.0.20-f6f6117dc4 (using bundled LuaJIT 2.1.0-

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


Prime numbers limit: 200000

Initializing worker threads...

Threads started!

WARNING: the --test option is deprecated. You can pass
CPU speed:
    events per second:   171.87

General statistics:
    total time:                          30.0043s
    total number of events:              5157

Latency (ms):
         min:                                    5.72
         avg:                                    5.81
         max:                                   15.25
         95th percentile:                        5.99
         sum:                                29967.81

Threads fairness:
    events (avg/stddev):           5157.0000/0.00
    execution time (avg/stddev):   29.9678/0.00
```

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


Prime numbers limit: 200000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   170.92

General statistics:
    total time:                          30.0060s
    total number of events:              5129

Latency (ms):
         min:                                    5.72
         avg:                                    5.85
         max:                                   13.94
         95th percentile:                        6.09
         sum:                                29982.56

Threads fairness:
    events (avg/stddev):           5129.0000/0.00
    execution time (avg/stddev):   29.9826/0.00

sysbench 1.0.20-f6f6117dc4 (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: 200000

Initializing worker threads...

WARNING: the --test option is deprecated. You can pass a s
Threads started!

CPU speed:
    events per second:   170.08

General statistics:
    total time:                          30.0068s
    total number of events:              5104

Latency (ms):
         min:                                    5.72
         avg:                                    5.87
         max:                                   16.41
         95th percentile:                        6.09
         sum:                                29981.24

Threads fairness:
    events (avg/stddev):           5104.0000/0.00
    execution time (avg/stddev):   29.9812/0.00

sysbench 1.0.20-f6f6117dc4 (using bundled LuaJIT 2.1.0-bet
```

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


Prime numbers limit: 200000

Initializing worker threads...

WARNING: the --test option is deprecated. You can pass a scrip
Threads started!

CPU speed:
    events per second:   168.58

General statistics:
    total time:                          30.0064s
    total number of events:              5059

Latency (ms):
         min:                                    5.72
         avg:                                    5.93
         max:                                   18.57
         95th percentile:                        6.21
         sum:                                29978.28

Threads fairness:
    events (avg/stddev):           5059.0000/0.00
    execution time (avg/stddev):   29.9783/0.00

WARNING: the --test option is deprecated. You can pass a scrip
sysbench 1.0.20-f6f6117dc4 (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: 200000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   167.92

General statistics:
    total time:                          30.0058s
    total number of events:              5039

Latency (ms):
         min:                                    5.73
         avg:                                    5.95
         max:                                   13.19
         95th percentile:                        6.09
         sum:                                29963.09

Threads fairness:
    events (avg/stddev):           5039.0000/0.00
    execution time (avg/stddev):   29.9631/0.00
```

Iteration	events/sec
1	171.87-max value
2	170.92
3	170.08
4	168.58
5	167.92-min value
average	169.87

4. Conclusion

There is very slight deviation in the performance even after using more resources for both QEMU and Docker. Also when we increase max-prime value, events/sec decreases.

5. File I/O testing QEMU vs Docker

For File I/O testing, 2 mods are supported by sysbench:

i. Sequential Rewrite (seqrewr)

ii. Combined random read/write (rndrw)

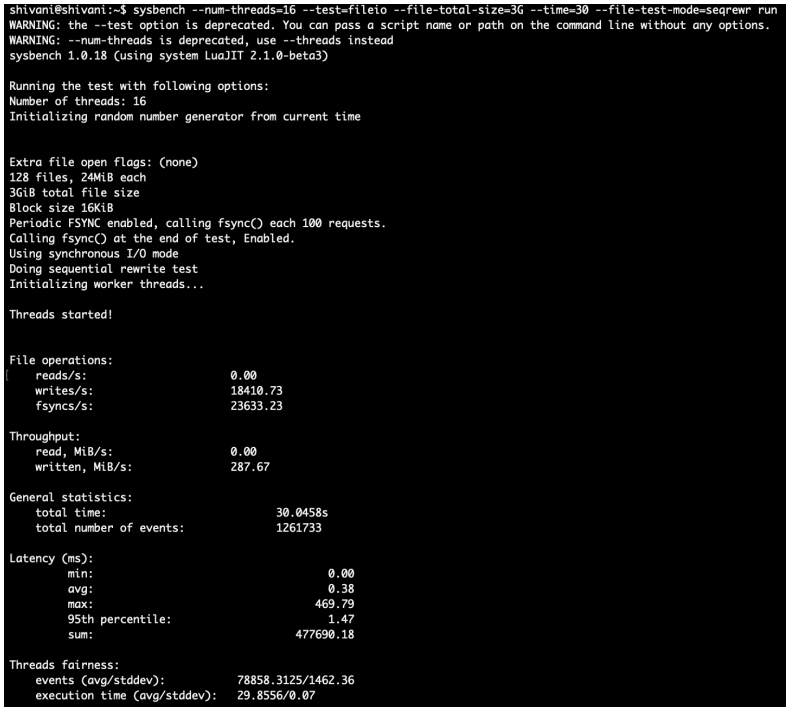
file size=3GB

Performance is tested using experimental setups.

I will use two modes which sysbench supports, which are: We will keep the file size constant at 3GB and will test the performance against our experimental setups accordingly.

QEMU execution:

1) Sequential Rewrite



```
shivani@shivani:~$ sysbench --num-threads=16 --test=fileio --file-total-size=3G --time=30 --file-test-mode=seqrewr run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 24MiB each
3GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      0.00
    writes/s:                     18410.73
    fsyncs/s:                     23633.23

Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               287.67

General statistics:
    total time:                          30.0458s
    total number of events:              1261733

Latency (ms):
         min:                                    0.00
         avg:                                    0.38
         max:                                  469.79
         95th percentile:                        1.47
         sum:                               477690.18

Threads fairness:
    events (avg/stddev):           78858.3125/1462.36
    execution time (avg/stddev):   29.8556/0.07
```

Iteration	output
1	reads/s=0 writes/s=18410.73 fsyncs/s=23633.23 events/s=21028.88
2	reads/s = 0 writes/sec = 44361.77 fsyncs/sec = 56848.83 events/second = 101252.66
3	reads/s = 0 writes/s=44132.92 fsyncs/sec = 56554.27 events/second = 100665.5
4	reads/s = 0 writes/s = 42941.03 fsyncs/sec = 55031.80 events/second = 97985.86
5	reads/s = 0 writes/s = 40869.42 fsyncs/s = 52379.79 events/second = 93235.5

Random Read Write(rndrw)

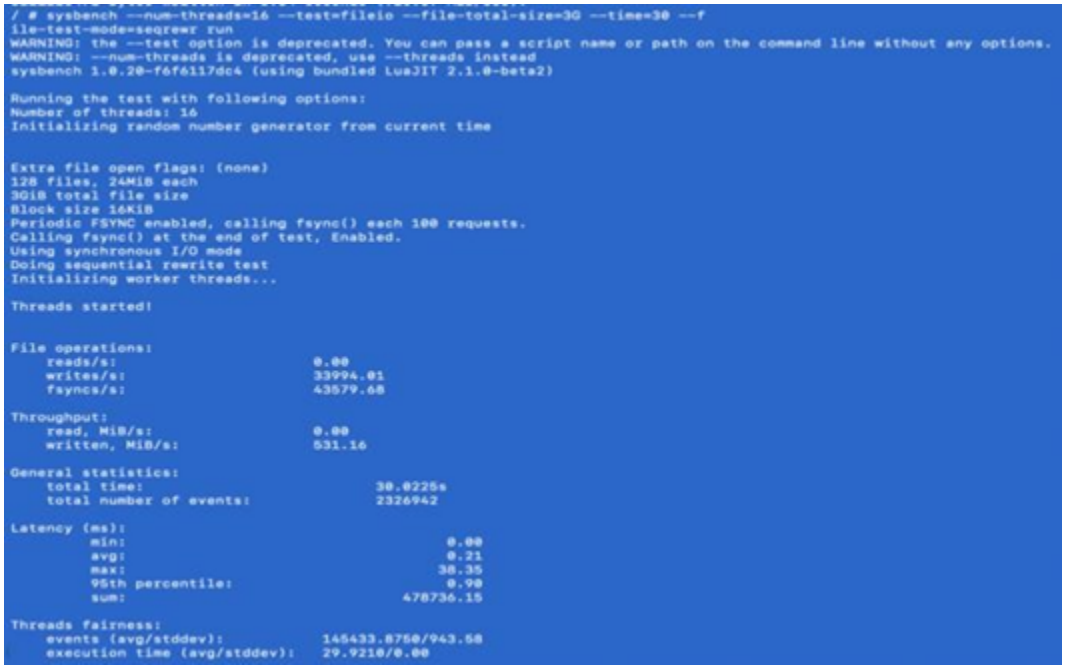
```
shivani@shivani:~$ sysbench --num-threads=16 --test=fileio --file-total-size=3G --time=30 --file-test-mode=rndrw run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 24MiB each
3GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      11594.62
    writes/s:                     7729.41
    fsyncs/s:                     24799.34

Throughput:
    read, MiB/s:                  181.17
    written, MiB/s:               120.77

General statistics:
    total time:                          30.0235s
    total number of events:              1323432

Latency (ms):
         min:                                    0.00
         avg:                                    0.36
         max:                                   84.02
         95th percentile:                        1.42
         sum:                               478877.93

Threads fairness:
    events (avg/stddev):           82714.5000/459.07
    execution time (avg/stddev):   29.9299/0.00
```

Iteration	output
1	reads/s = 11594.62 writes/s = 7729.41 fsyncs/s = 24799.34 events/second = 22057.2
2	reads/s = 12925.26 writes/s = 8616.51 fsyncs/s = 27637.79 events/second = 49145.3
3	reads/s = 10645.87 writes/s = 7096.89 fsyncs/s =22777.00 events/second = 40479.13
4	reads/s = 11293.72

	writes/s = 7529.37 fsyncs/s = 24157.81 events/seconds = 42936.03
5	reads/s = 12491.06 writes/s = 8327.07 fsyncs/s = 26713.69 events/seconds = 47486.16

Docker execution:

1) Sequential Read Write



Iteration	output
1	reads/s = 0 writes/sec = 33944.01 fsyncs/sec = 43579.68 events/second = 77564.73
2	reads/s = 0

	writes/s=35435.34 fsyncs/sec = 42342.32 events/second = 75343.24
3	reads/s = 0 writes/s = 33958.73 fsyncs/sec = 43684.54 events/second = 77846.34
4	reads/s = 0 writes/s = 32091.93 fsyncs/s = 41321.54 events/second = 73654.18
5	reads/s = 0 writes/s = 34546.54 fsyncs/s = 44323.20 events/second = 78234.32

2) Random read write

Iteration	output
1	reads/s = 13950.71 writes/sec = 9300.24 fsyncs/sec = 29826.54 events/second = 53042.66
2	reads/s = 13155.49 writes/s=8769.99 fsyncs/sec = 28129.15 events/second = 50019.5
3	reads/s = 14353.95 writes/s = 9569.02 fsyncs/sec = 30688.31 events/second = 54574.26

4	reads/s = 14674.66 writes/s = 9782.83 fsyncs/s = 31370.93 events/second = 55796.33
5	reads/s = 14488.90 writes/s = 9659.14 fsyncs/s = 30975.53 events/second = 55094.1

Conclusion:

Sequential rewrite operations is faster on QEMU than Docker. While for combined read write, docker is faster than QEMU.

CONFIGURATION 2: 4GB RAM 4 CORES

Now, we change our system configuration and run the same tests again, to verify whether we

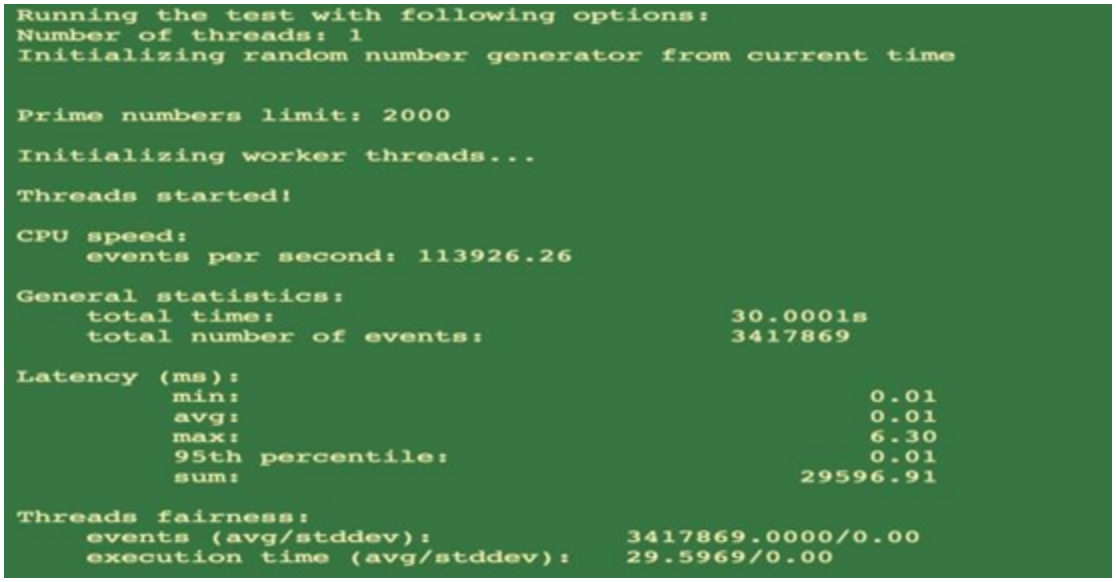
see any performance changes. Our next configuration is 4GB ram and 4 CPU cores for both

our QEMU VM and Docker.

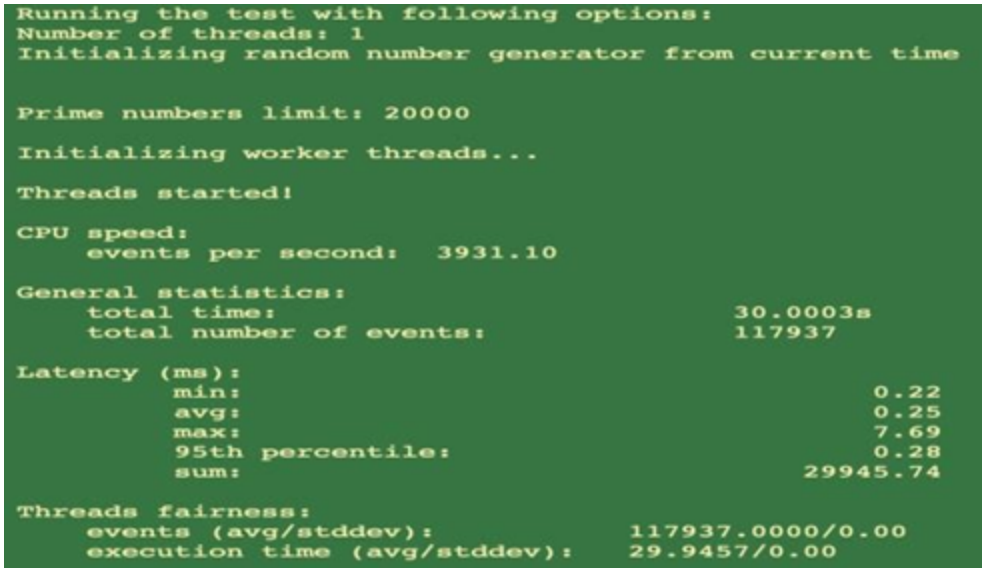
a. CPU TESTING

i. QEMU CPU TESTING

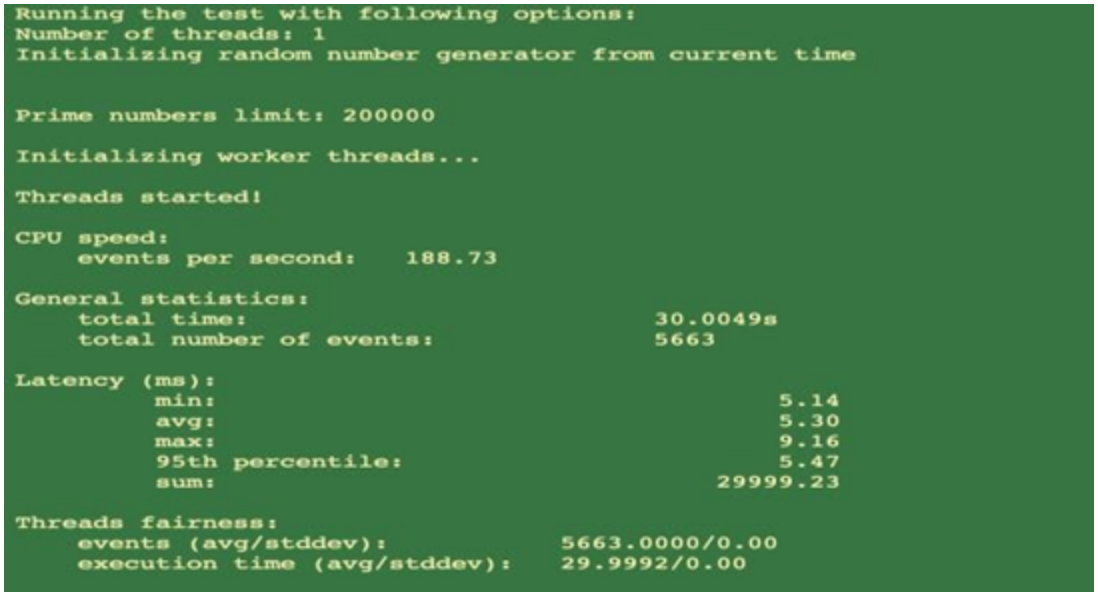
TEST CASE 1:



Iteration	events/sec
1	113926.26
2	113201.06
3	113018.63-min
4	114854.19-max
5	113129.79
average	113625.98

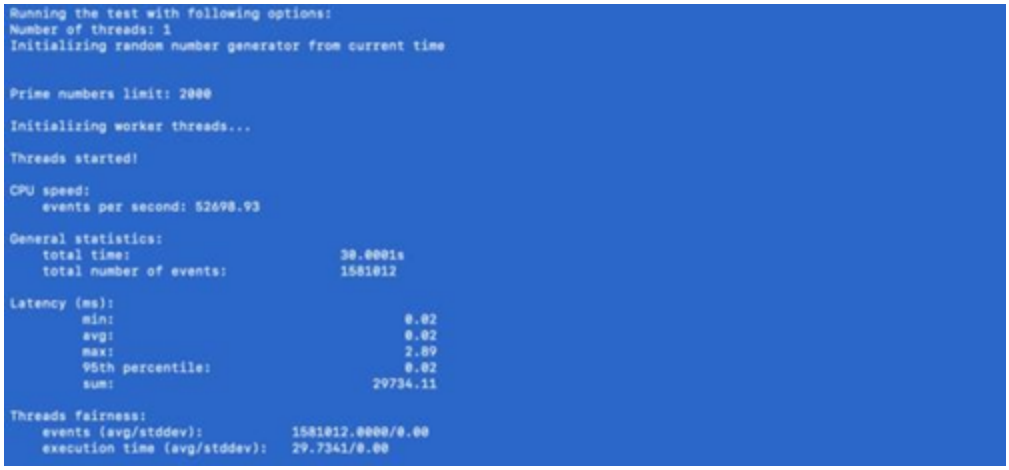


Iteration	events/sec
1	3931.10
2	3933.90
3	4281.01-max
4	4051.55-min
5	3972.59
average	4034.03

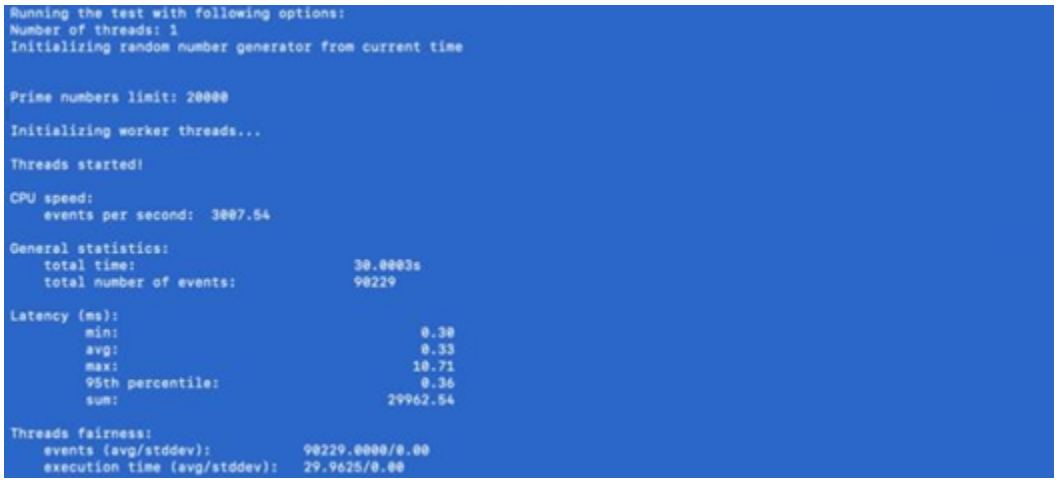


Iteration	events/sec
1	188.73-max
2	188.33
3	185.49-min
4	189.86
5	189.83
average	188.44

Docker

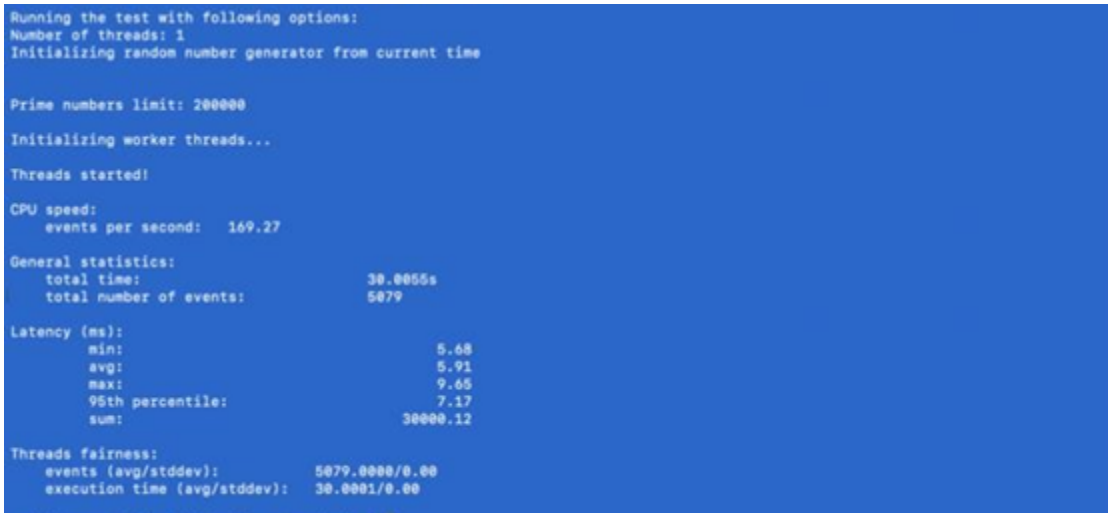


Iteration	events/sec
1	52698.93
2	52132.09
3	51893.90-min
4	56403.96-max
5	56134.25
average	53852.62



Iteration	events/sec

1	3007.54
2	3247.12-max
3	3181.35
4	3164.53
5	2917.72-min
average	3103.65



Iteration	events/sec
1	169.27
2	172.51
3	172.34
4	171.67
5	173.24
average	171.80

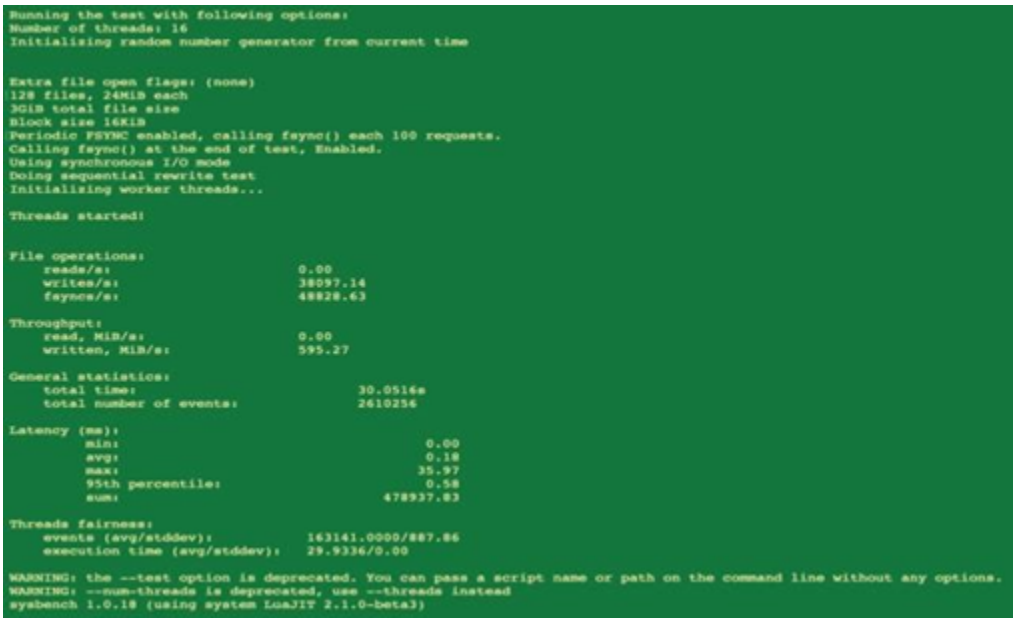
Conclusion:

QEMU is faster than docker. But change in configuration doesnt have significant effect on performance.

File IO testing

QEMU

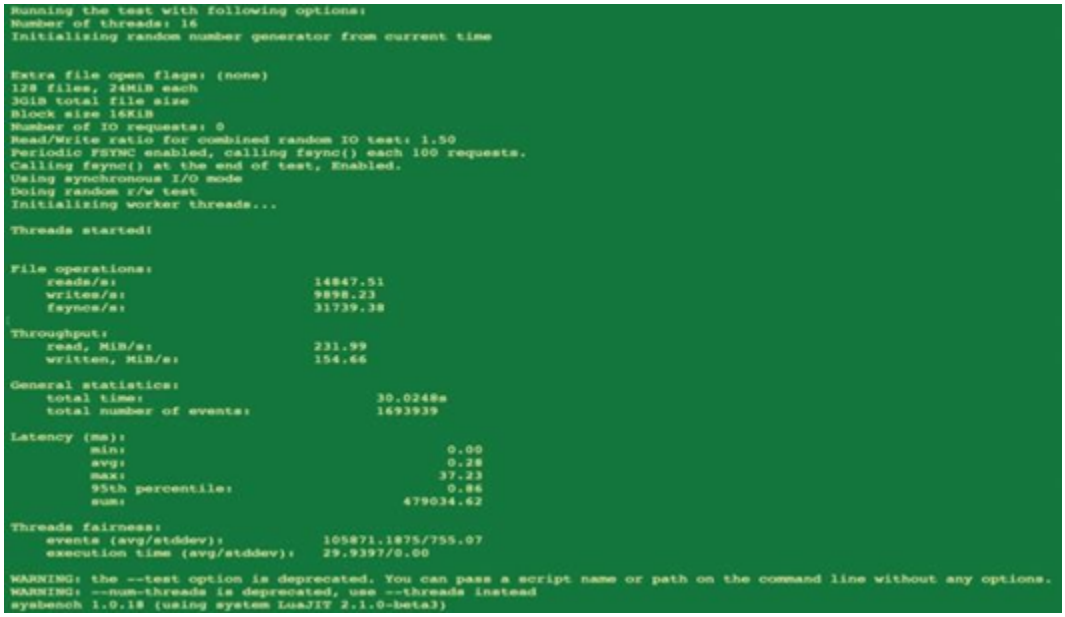
1) Sequential Rewrite



Iteration	results
1	reads/s = 0 writes/sec = 38097.14 fsyncs/sec = 48828.63 events/second = 87008.56
2	reads/s = 0

	writes/s=39530.62 fsyncs/sec = 50664.21 events/second = 90199.3
3	reads/s = 0 writes/s = 35886.20 fsyncs/sec = 46000.81 events/second = 81888.26
4	reads/s = 0 writes/s = 33329.60 fsyncs/s = 42727.80 events/second = 76088.93
5	reads/s = 0 writes/s = 33704.58 fsyncs/s = 43206.91 events/second = 76908.83

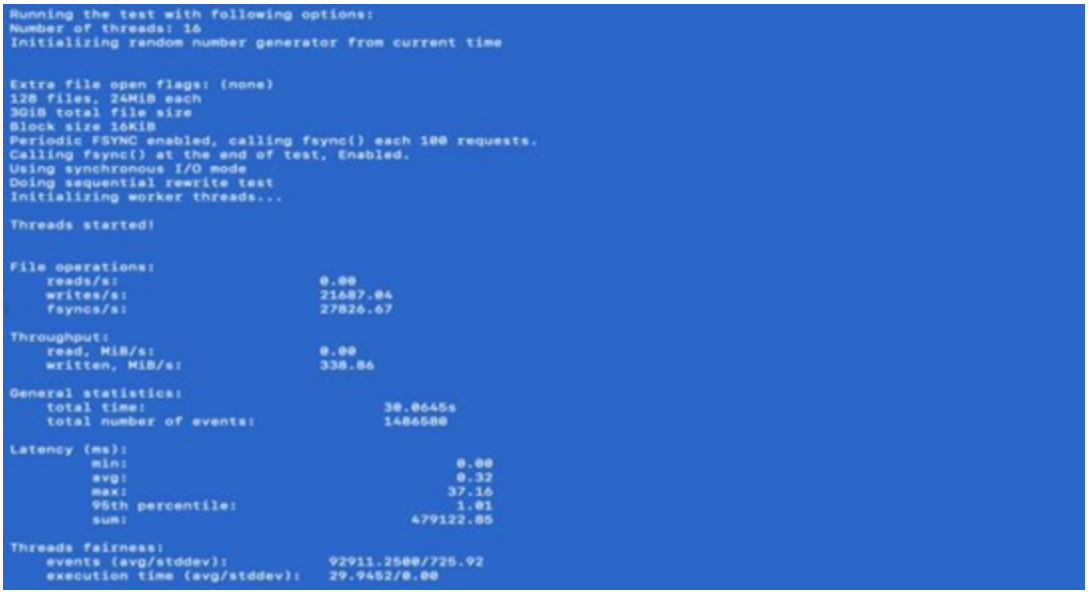
Random Read Write



Iteration	results
1	reads/s = 14487.51 writes/s = 9898.23 fsyncs/s = 31739.38 events/second = 56464.43
2	reads/s = 13910 writes/s = 9273 fsyncs/s = 29740 events/second = 52915.76
3	reads/s = 14165.54 writes/s = 9443.58 fsyncs/s =30284.24 events/second = 53880.36
4	reads/s = 12349.31 writes/s = 8232.87 fsyncs/s = 26409.53 events/seconds = 46994.56
5	reads/s = 12001.06 writes/s = 8000.63 fsyncs/s = 25669.69 events/seconds =45638.56

Docker

1) Sequential Rewrite



Iteration	results
1	reads/s = 0 writes/sec = 21687.04 fsyncs/sec = 27826.67 events/second = 49552.66
2	reads/s = 0 writes/s= 22690.64 fsyncs/sec = 29019.49 events/second = 51806.5
3	reads/s = 0 writes/s = 25688.43 fsyncs/sec = 32946.24 events/second = 58659.23
4	reads/s = 0 writes/s = 22382.72 fsyncs/s = 28716.33 events/second = 51146.33

5	reads/s = 0 writes/s = 21880.69 fsyncs/s =28071.27 events/second = 49996.26

Random Read Write

Iteration	results
1	reads/s = 13850.71 writes/sec = 9320.24 fsyncs/sec = 28826.54 events/second = 53032.66
2	reads/s = 14155.49 writes/s=8754.94 fsyncs/sec = 27129.15 events/second = 50045.5

3	reads/s = 14251.83 writes/s = 9545.02 fsyncs/sec = 30457.31 events/second = 54375.26
4	reads/s = 14324.66 writes/s = 9231.83 fsyncs/s = 31432.93 events/second = 55987.33
5	reads/s = 14231.90 writes/s = 9658.14 fsyncs/s = 30324.53 events/second = 55432.1

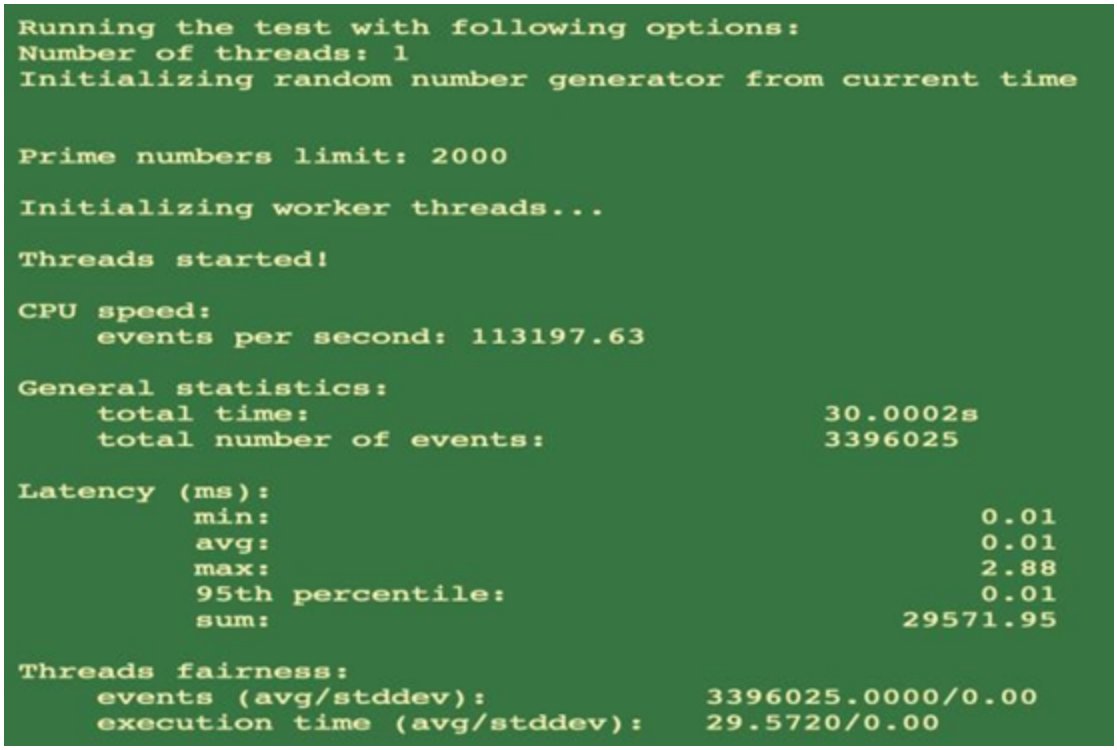
Conclusion:

As compared with 2GB 2 core config, there is decrease in file i/o for QEMU and Docker.

Configuration 3: 6GB 6 cores

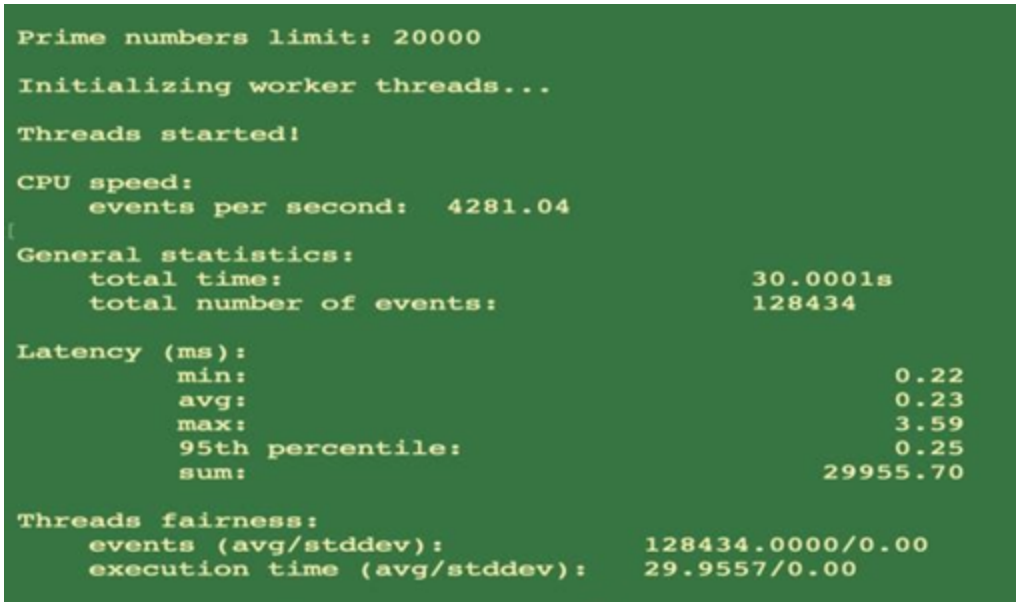
QEMU

Test Case 1



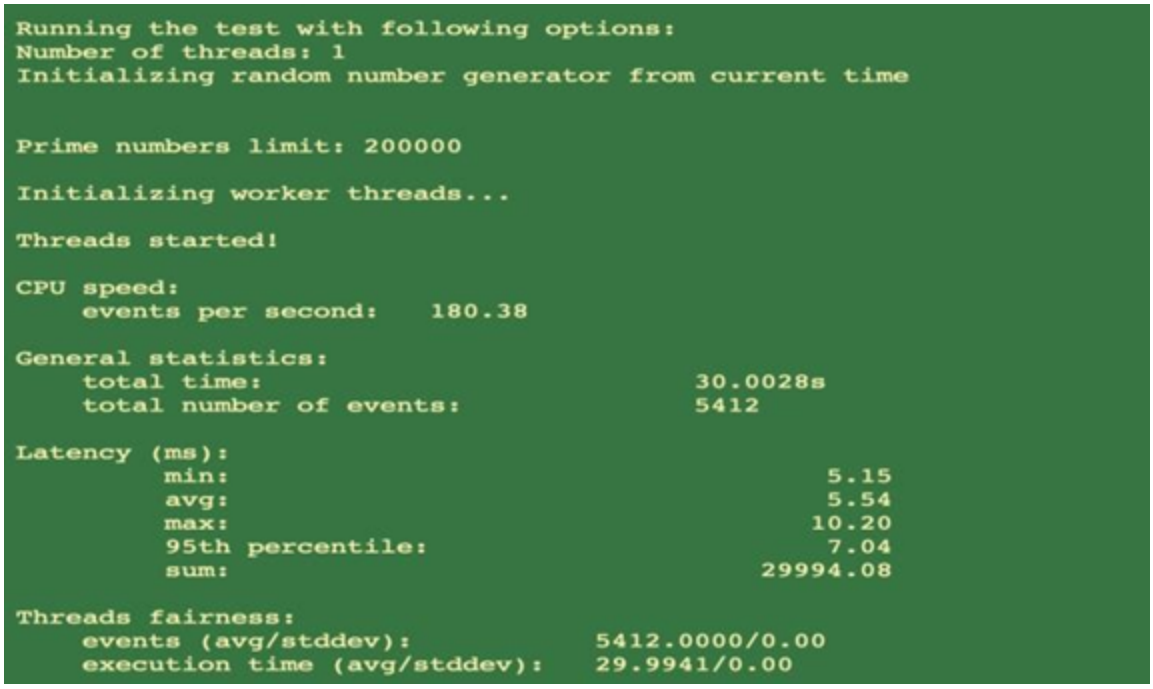
Iteration	events/sec
1	113197.63-min
2	114251.46
3	113318.53
4	114254.59-max
5	113529.89
average	113710.42

test case 2



Iteration	events/sec
1	4281.04-max
2	3945.40
3	4284.01
4	4054.45
5	3942.69-min
average	4101.51

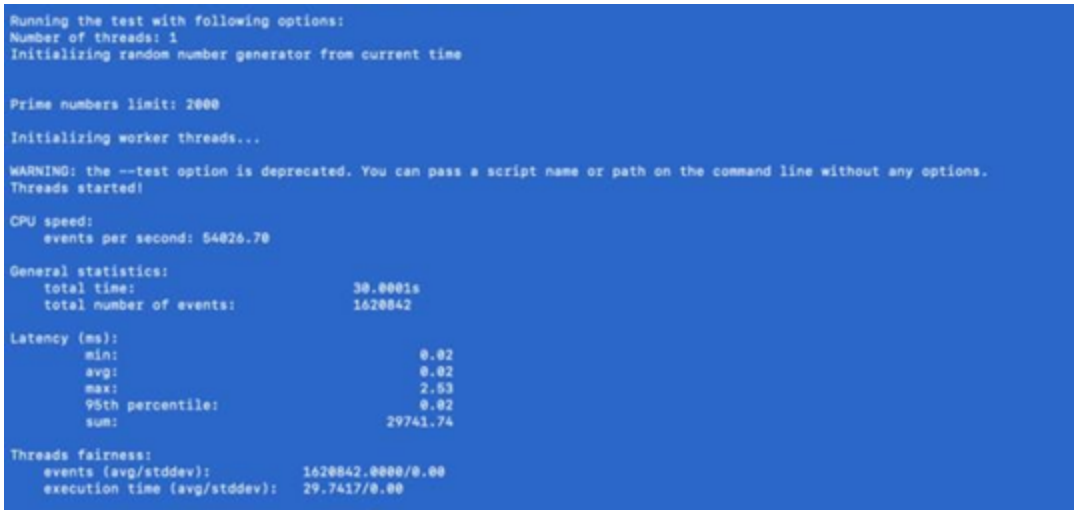
test case 3



Iteration	events/sec
1	180.38-min
2	182.33
3	184.49-max
4	180.86
5	181.83
average	181.97

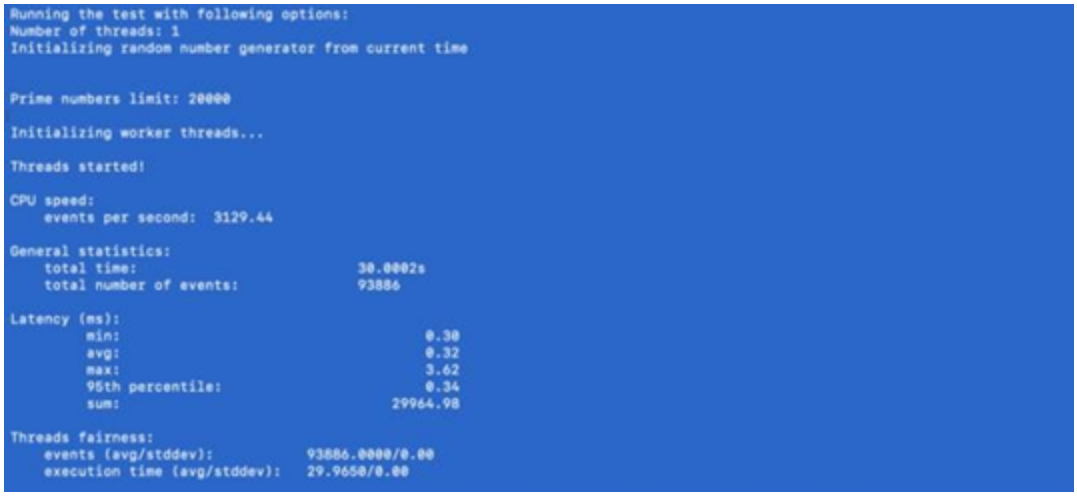
Docker

Test case 1



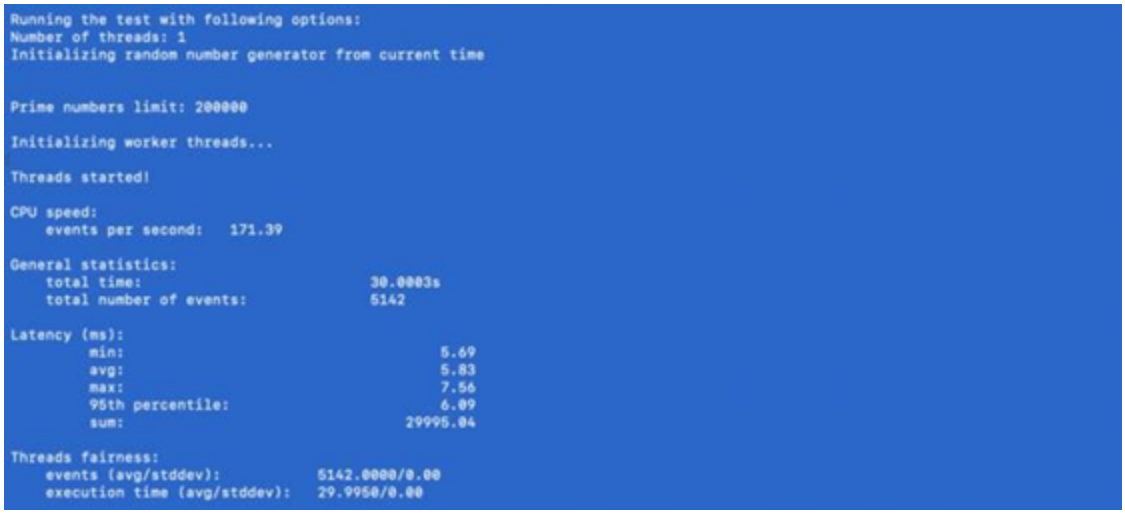
Iteration	events/sec
1	54026.70
2	52534.09
3	52843.95-min
4	53403.88
5	56479.38-max
average	53857.6

test case 2



Iteration	events/sec
1	3129.44
2	3147.42
3	3141.35
4	3264.53-max
5	3024.72-min
average	3141.49

test case 3



Iteration	events/sec
1	171.39
2	172.47
3	171.34-min
4	172.67
5	174.24-max
average	172.42

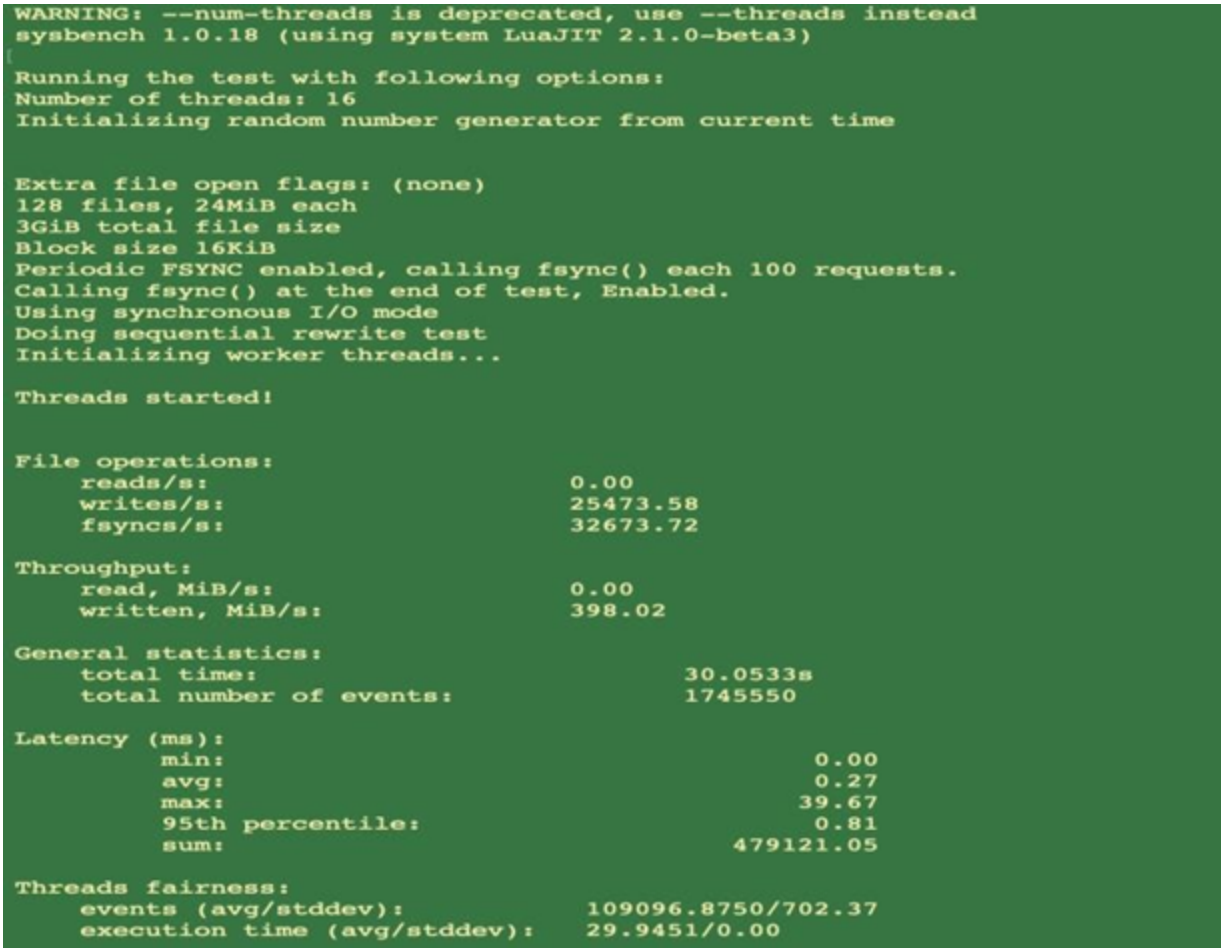
Conclusion:

There is no significant change in CPU performance after adding resources.

File IO testing

QEMU

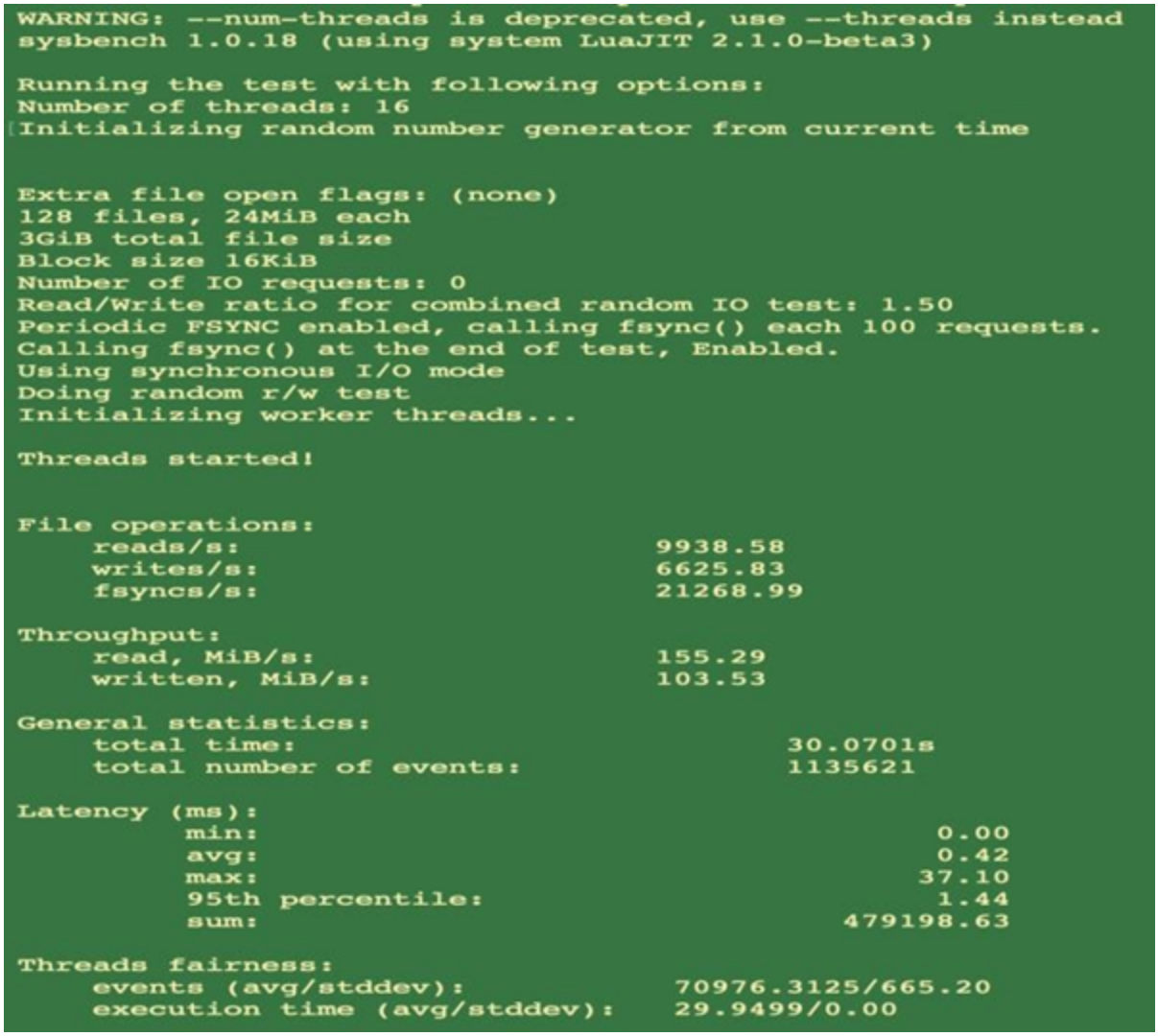
1) Sequential Rewrite



Iteration	results
1	reads/s = 0 writes/sec = 25473.58 fsyncs/sec = 32673.72 events/second = 58185
2	reads/s = 0 writes/s=27687.54 fsyncs/sec = 33764.65 events/second = 59124.32

3	reads/s = 0 writes/s = 24534.32 fsyncs/sec = 31986.87 events/second = 56126.76
4	reads/s = 0 writes/s = 25765.32 fsyncs/s = 32675.83 events/second = 58234.21
5	reads/s = 0 writes/s = 28675.77 fsyncs/s = 35687.32 events/second = 59543.65

Random Read Write



Iteration	results
1	reads/s = 9938.58 writes/s = 6625.83 fsyncs/s = 21268.98 events/second = 37854.03
2	reads/s = 9845.43 writes/s = 6541.32 fsyncs/s = 20267.32

	events/second = 35643.76
3	reads/s = 9876.73 writes/s = 6543.21 fsyncs/s =21124.43 events/second = 35541.28
4	reads/s = 9765.53 writes/s = 7021.32 fsyncs/s = 30832.11 events/seconds = 39878.32
5	reads/s = 9763.23 writes/s = 6312.63 fsyncs/s = 24569.69 events/seconds =34532.56

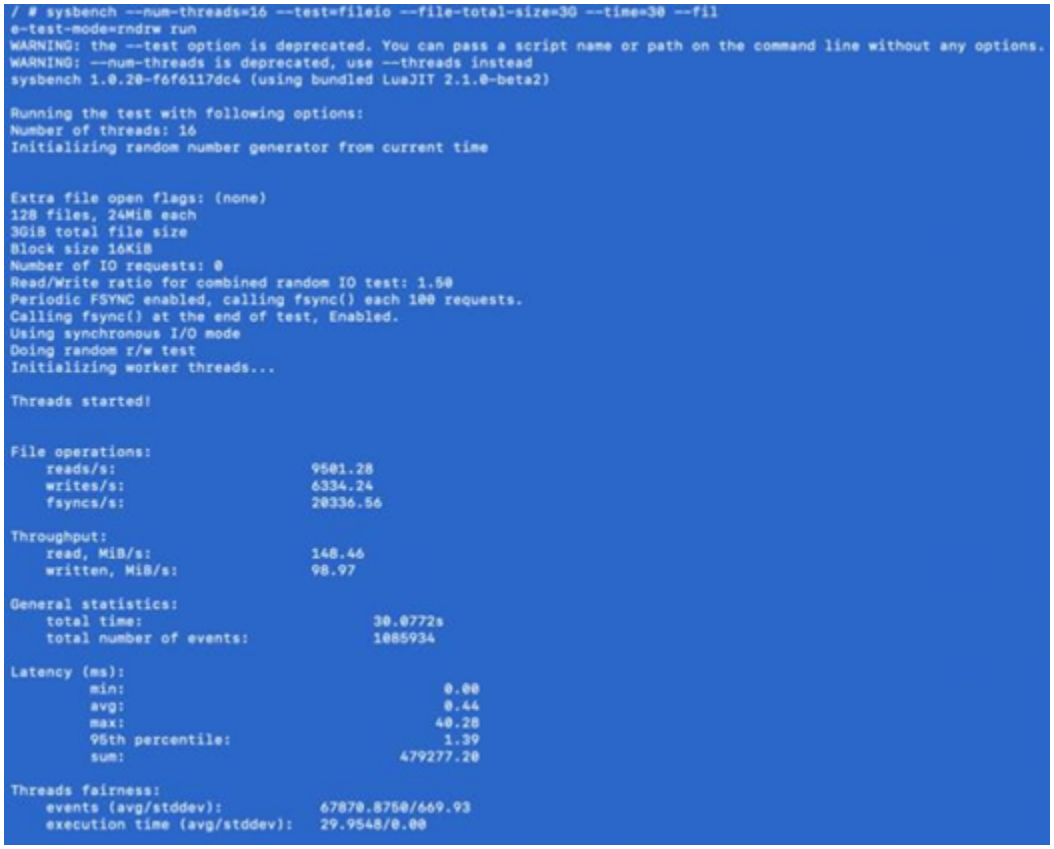
Docker

1) Sequential Rewrite

Iteration	results
1	reads/s = 0 writes/sec = 13778.77 fsyncs/sec = 17701.47 events/second = 31506.16
2	reads/s = 0 writes/s= 13543.64 fsyncs/sec = 17634.49 events/second = 30234.5
3	reads/s = 0 writes/s = 12456.32 fsyncs/sec = 16323.87 events/second = 29876.36
4	reads/s = 0

	writes/s = 14328.87 fsyncs/s = 18785.62 events/second = 32324.32
5	reads/s = 0 writes/s = 13534.11 fsyncs/s =17431.89 events/second = 31232.21

Random Read Write



Iteration	results
1	reads/s = 9501.28

	writes/sec = 6334.24 fsyncs/sec = 20336.56 events/second = 36197.8
2	reads/s = 9323.32 writes/s=6213.41 fsyncs/sec = 20223.12 events/second = 36097.87
3	reads/s = 9674.98 writes/s = 6984.73 fsyncs/sec = 20989.89 events/second = 37019.21
4	reads/s = 9594.32 writes/s = 6434.43 fsyncs/s = 20214.43 events/second = 35788.33
5	reads/s = 9287.90 writes/s = 6021.14 fsyncs/s = 18673.53 events/second = 32764.12

Conclusion:

Performance of file io decreases if resources are increased. Sometimes QEMU is better than Dccker.

Performance Analysis:

a) QEMU

Disk

1) 2GB 2cores

   i) Sequential rewrite

   read,MiB/s=0

   written,MiB/s=715.68

   ii) Random Read Write

   read, MiB/s=188.23

   written, MiB/s=122.43

2) 4GB 4 cores

   i) Sequential rewrite

   read,MiB/s=0

   written,MiB/s=595.27

   ii) Random Read Write

   read, MiB/s=231.99

   written, MiB/s=154.66

3) 6GB 6 cores

   i) Sequential rewrite

   read,MiB/s=0

   written,MiB/s=398.02

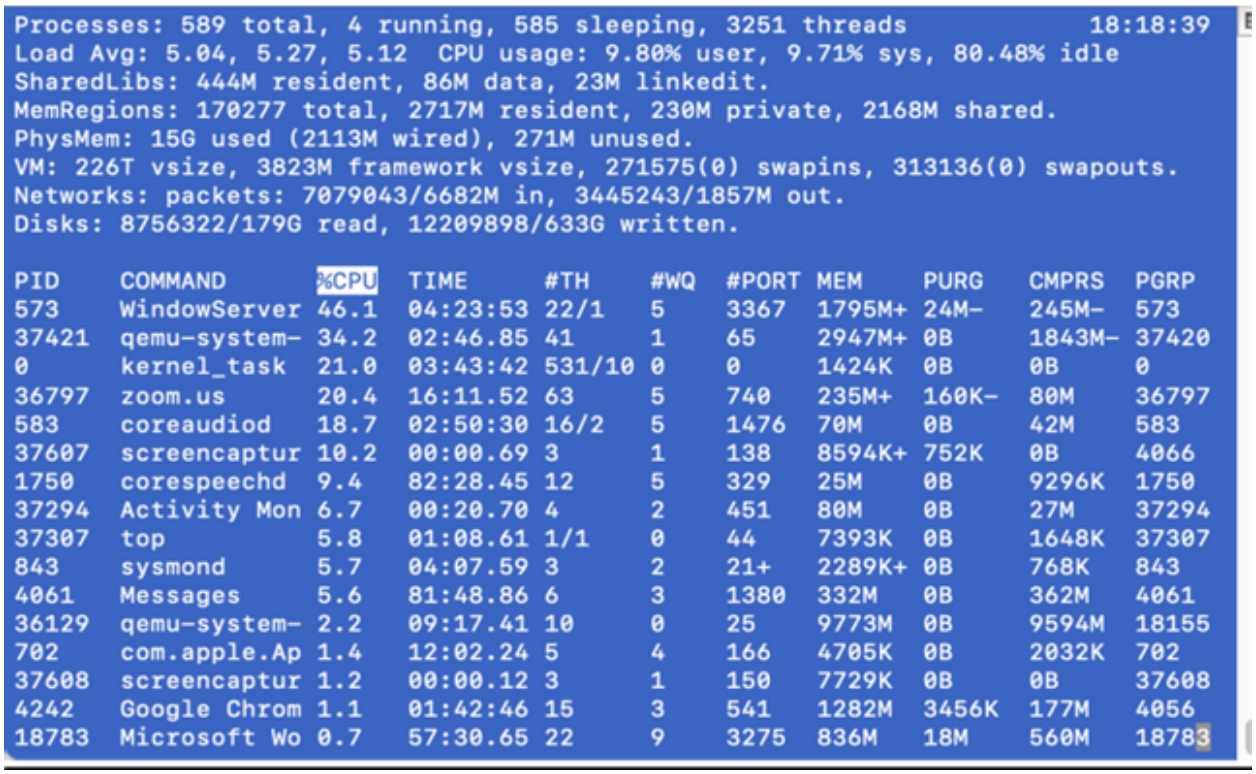
   ii) Random Read Write

   read, MiB/s=155.29

   written, MiB/s=103.53

CPU

Google C...	11.8	19:19.38	24	115	Apple	0.0	0.00	18913	yashbhargava
Messages	6.4	1:25:36.87	6	132	Apple	0.0	0.00	4061	yashbhargava
qemu-sy...	3.2	11:58.50	14	96	Apple	0.0	0.00	36129	yashbhargava



```
Processes: 589 total, 4 running, 585 sleeping, 3251 threads                 18:18:39
Load Avg: 5.04, 5.27, 5.12  CPU usage: 9.80% user, 9.71% sys, 80.48% idle
SharedLibs: 444M resident, 86M data, 23M linkedit.
MemRegions: 170277 total, 2717M resident, 230M private, 2168M shared.
PhysMem: 15G used (2113M wired), 271M unused.
VM: 226T vsize, 3823M framework vsize, 271575(0) swapins, 313136(0) swapouts.
Networks: packets: 7079043/6682M in, 3445243/1857M out.
Disks: 8756322/179G read, 12209898/633G written.

PID    COMMAND      %CPU  TIME     #TH    #WQ  #PORT MEM    PURG   CMPRS  PGRP
573    WindowServer 46.1  04:23:53 22/1   5    3367  1795M+ 24M-   245M-  573
37421  qemu-system- 34.2  02:46.85 41     1    65    2947M+ 0B     1843M- 37420
0      kernel_task  21.0  03:43:42 531/10 0    0     1424K  0B     0B     0
36797  zoom.us      20.4  16:11.52 63     5    740   235M+  160K-  80M    36797
583    coreaudiod   18.7  02:50:30 16/2   5    1476  70M    0B     42M    583
37607  screencaptur 10.2  00:00.69 3      1    138   8594K+ 752K   0B     4066
1750   corespeechd  9.4   82:28.45 12     5    329   25M    0B     9296K  1750
37294  Activity Mon 6.7   00:20.70 4      2    451   80M    0B     27M    37294
37307  top          5.8   01:08.61 1/1    0    44    7393K  0B     1648K  37307
843    sysmond      5.7   04:07.59 3      2    21+   2289K+ 0B     768K   843
4061   Messages     5.6   81:48.86 6      3    1380  332M   0B     362M   4061
36129  qemu-system- 2.2   09:17.41 10     0    25    9773M  0B     9594M  18155
702    com.apple.Ap 1.4   12:02.24 5      4    166   4705K  0B     2032K  702
37608  screencaptur 1.2   00:00.12 3      1    150   7729K  0B     0B     37608
4242   Google Chrom 1.1   01:42:46 15     3    541   1282M  3456K  177M   4056
18783  Microsoft Wo 0.7   57:30.65 22     9    3275  836M   18M    560M   18783
```

Percentage of CPU used – 34.2%

Kernel Usage – user = 9.71% , System – 9.71%, Idle – 80.48%

b) Docker

Disk

1) 2GB 2 cores

i) Sequential rewrite

read,MiB/s=0

written,MiB/s=531.16

ii) Random Read Write

read, MiB/s=217.98

written, MiB/s=145.32

2) 4GB 4 cores

i) Sequential rewrite

read,MiB/s=0

written,MiB/s=338.86

ii) Random Read Write

read, MiB/s=213.43

written, MiB/s=142.28

3) 6GB 6 cores

i) Sequential rewrite

read,MiB/s=0

written,MiB/s=215.29

ii) Random Read Write

read, MiB/s=148.46

written, MiB/s=98.97

CPU

```
Mem: 921604K used, 5159976K free, 326472K shrd, 86512K buff, 544596K cached
CPU:   0% usr   0% sys   0% nic  99% idle   0% io   0% irq   0% sirq
Load average: 0.16 0.08 0.03 3/506 10
  PID  PPID USER     STAT   VSZ %VSZ CPU %CPU COMMAND
    1     0 root     S     1728   0%   1   0% /bin/sh
   10     1 root     R     1660   0%   5   0% top
```

CPU Time used – 4hrs 45mins
Kernel used – 921604Kb => 115.2 MB

Git Repository

Account Name:SDeosatwar

link to repo: https://github.com/SDeosatwar/COEN241.git

commit id: 747ee30857ef478af226f78c6ec345b099e243a1