Homework 1
COEN 241 System vs OS Virtualization

Host System Configuration

- CPU: Apple M1 Pro with 10 CPU cores
- Memory: 16GB
- OS: MacOS

QEMU Installation:

brew install qemu

qemu-img create -f raw /qemu/ubuntu-latest.raw 35G

Download UEFI driver files namely, QEMU_EFI.fd and QEMU_VARS.fd

https://gist.github.com/theboreddev/5f79f86a0f163e4a1f9df919da5eea20#:~:text=QEMU_EFI%2Dcb438b9%2Dedk2%2Dstable202011%2Dwith%2Dextra%2Dresolutions.tar.gz

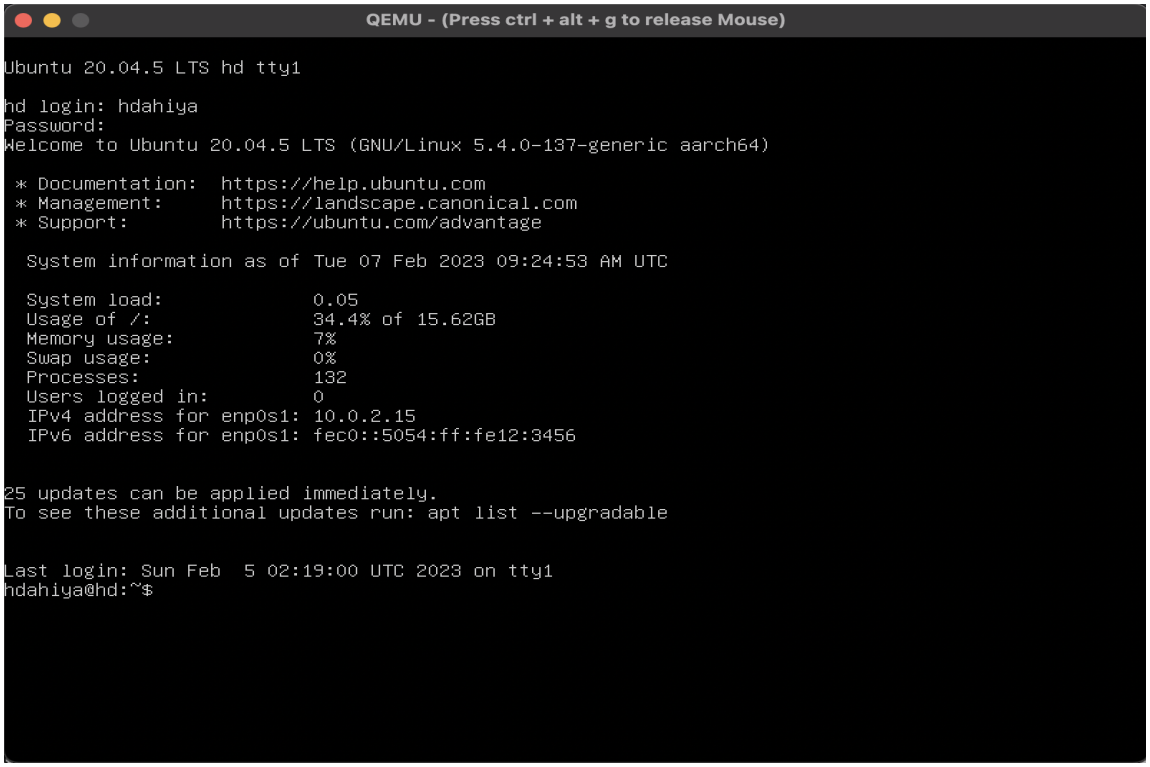
Install Ubuntu (CPUs set to 4 and memory 3000MB)

```
qemu-system-aarch64 \
  -monitor stdio \
  -M virt,highmem=off \
  -accel hvf \
  -cpu host \
  -smp 4 \
  -m 3000 \
  -bios QEMU_EFI.fd \
  -device virtio-gpu-pci \
  -display default,show-cursor=on \
  -device qemu-xhci \
  -device usb-kbd \
  -device usb-tablet \
  -device intel-hda \
  -device hda-duplex \
  -drive file=ubuntu-latest.raw,format=raw,if=virtio,cache=writethrough \
  -cdrom ubuntu-22.04.1-live-server-arm64.iso
```

Start Ubuntu with 4 cpus and 3000MB memory

```
himanshudahiya@Himanshus-MacBook-Pro coen241 % qemu-system-aarch64 \
    -monitor stdio \
    -M virt,highmem=off \
    -accel hvf \
    -cpu host \
    -smp 4 \
    -m 3000 \
    -bios QEMU_EFI.fd \
    -device virtio-gpu-pci \
    -display default,show-cursor=on \
    -device qemu-xhci \
    -device usb-kbd \
    -device usb-tablet \
    -device intel-hda \
    -device hda-duplex \
    -drive file=ubuntu.raw,format=raw,if=virtio,cache=writethrough
QEMU 7.2.0 monitor - type 'help' for more information
(qemu) audio: Failed to create voice `adc'
himanshudahiya@Himanshus-MacBook-Pro coen241 %
```

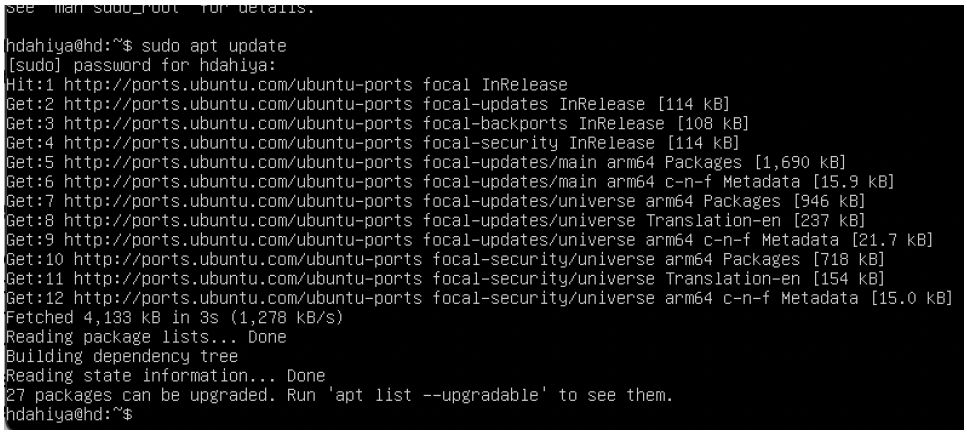
Login Ubuntu

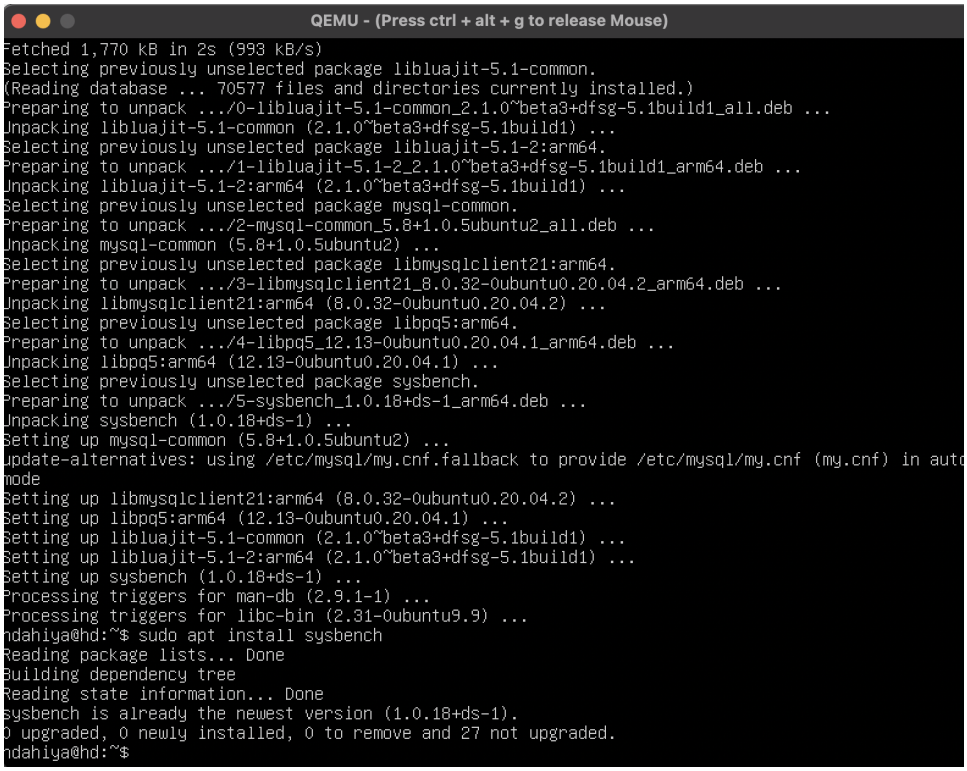


Install sysbench on Ubuntu

sudo apt update

sudo apt install sysbench





Docker Installation:

Download docker and install

Start docker

docker run hello-world

```
coen241 — -zsh — 96×29
himanshudahiya@Himanshus-MacBook-Pro coen241 % docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
7050e35b49f5: Pull complete
Digest: sha256:aa0cc8055b82dc2509bed2e19b275c8f463506616377219d9642221ab53cf9fe
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/

himanshudahiya@Himanshus-MacBook-Pro coen241 %
```

Download Ubuntu image

```
himanshudahiya@Himanshus-MacBook-Pro coen241 % docker pull ubuntu:latest
latest: Pulling from library/ubuntu
8b150fd943bc: Pull complete
Digest: sha256:9a0bdde4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
himanshudahiya@Himanshus-MacBook-Pro coen241 %
```

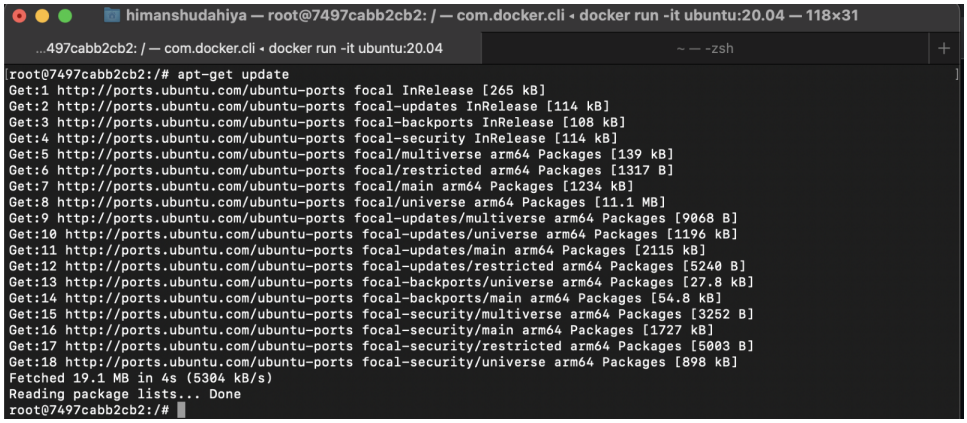
Run Ubuntu container

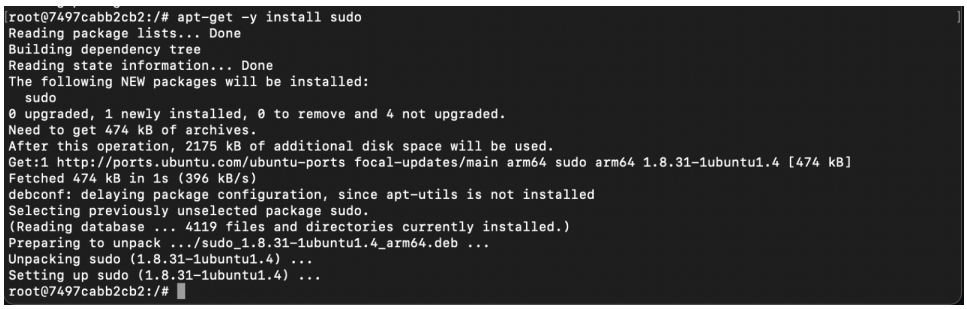
```
himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 118×31
himanshudahiya@Himanshus-MacBook-Pro ~ % docker run -it ubuntu:20.04
root@7497cabb2cb2:/#
```

Check if the container is running

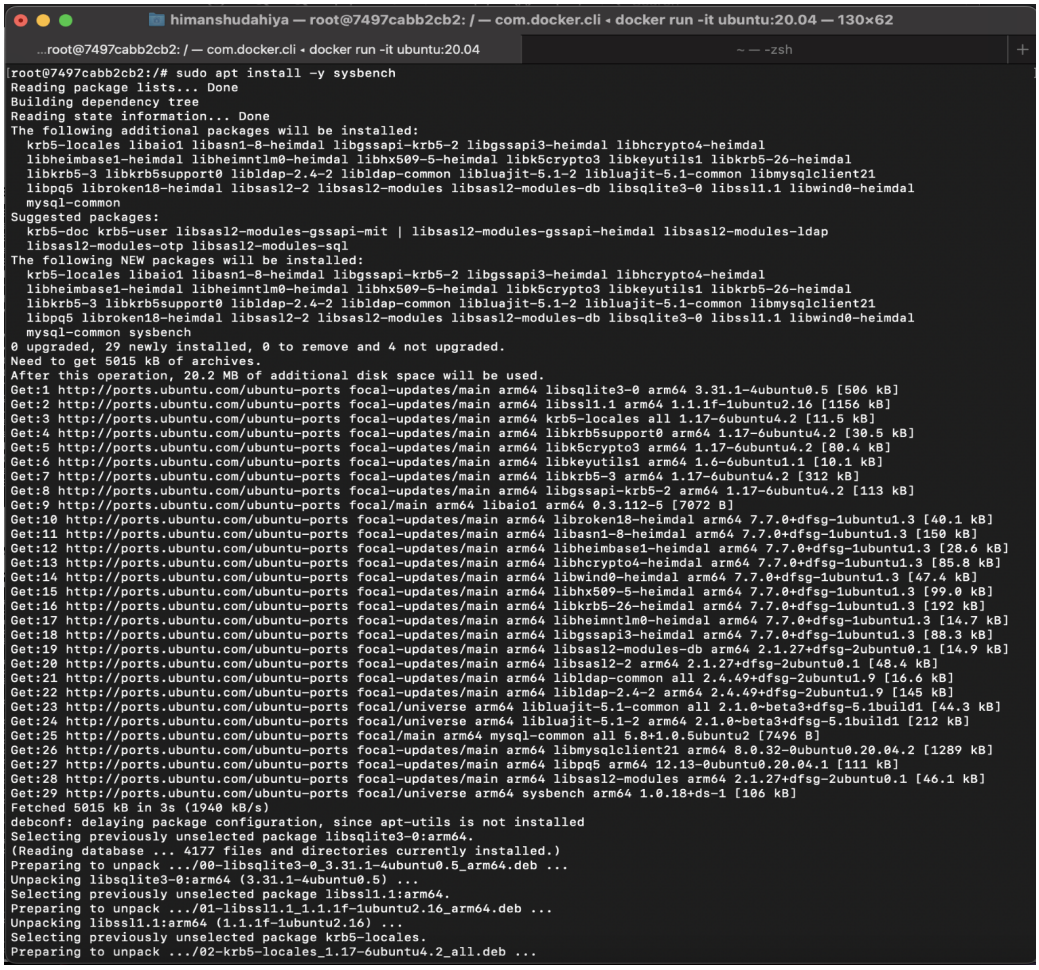
```
himanshudahiya — -zsh — 118×31
...497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 ...   ~ — -zsh
himanshudahiya@Himanshus-MacBook-Pro ~ % docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED          STATUS          PORTS     NAMES
7497cabb2cb2   ubuntu:20.04   "/bin/bash"   51 seconds ago   Up 50 seconds             gifted_satoshi
himanshudahiya@Himanshus-MacBook-Pro ~ %
```

Update and install sudo command





Install sysbench

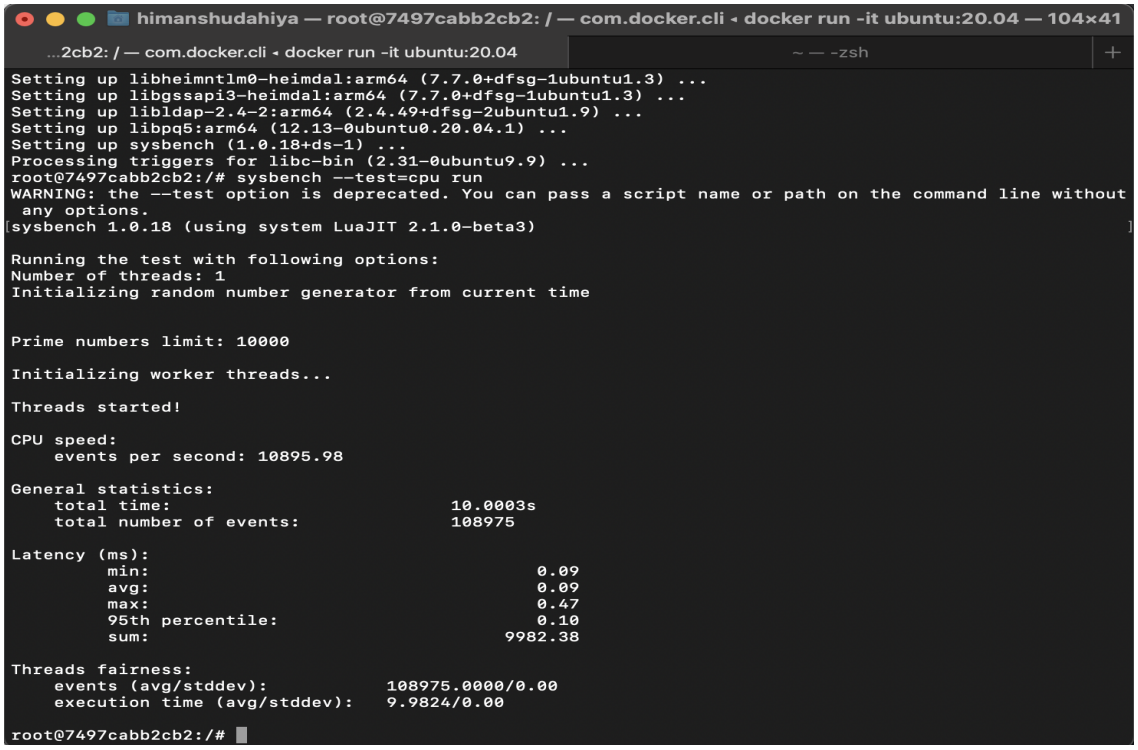


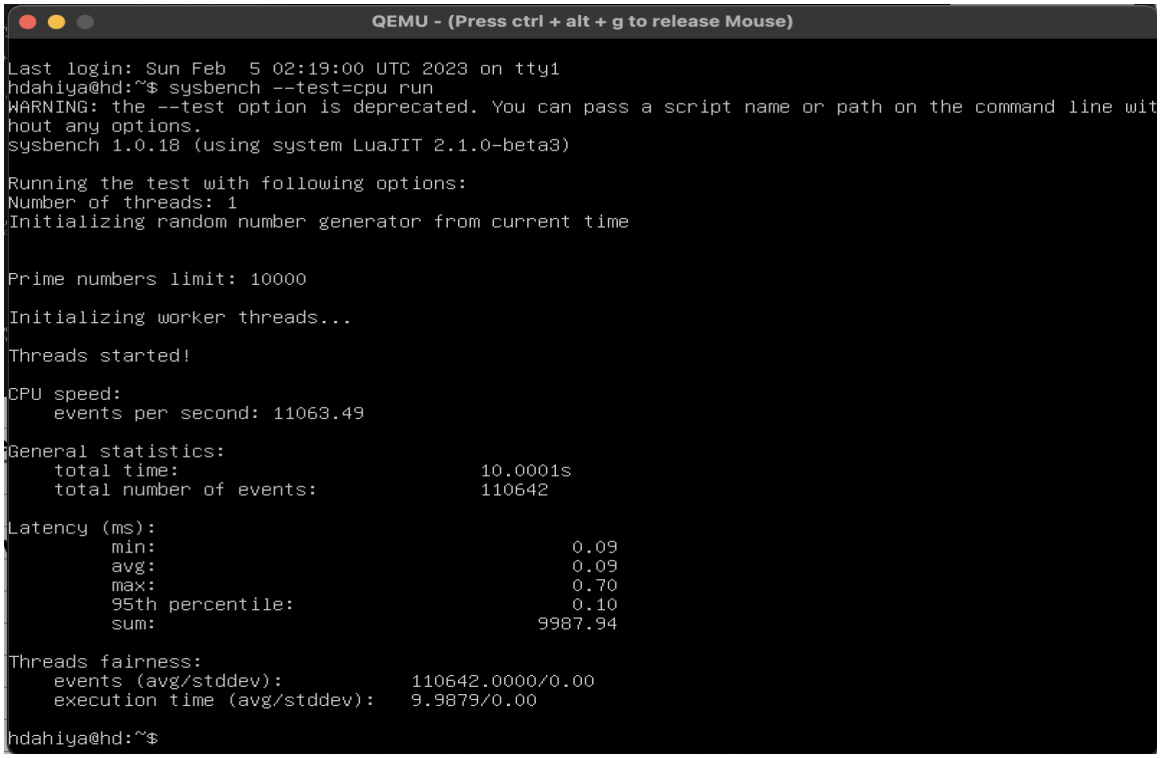
Running Tests on Docker and QEMU

Testcase 1

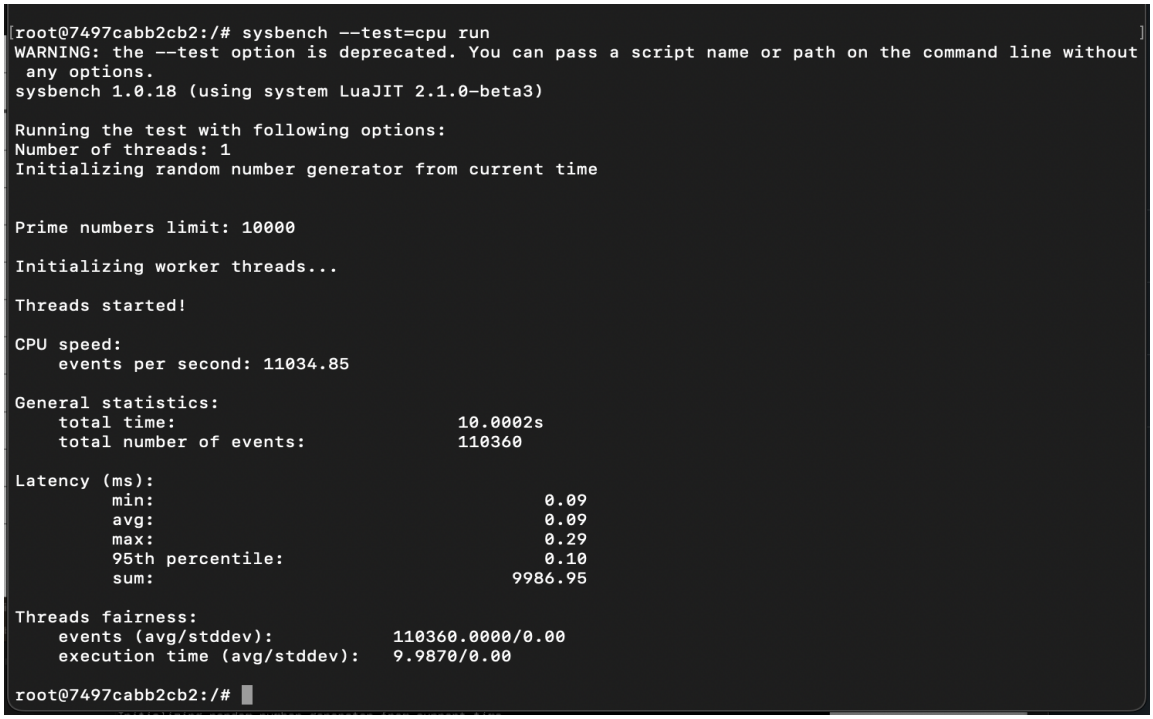
sysbench --test=cpu run

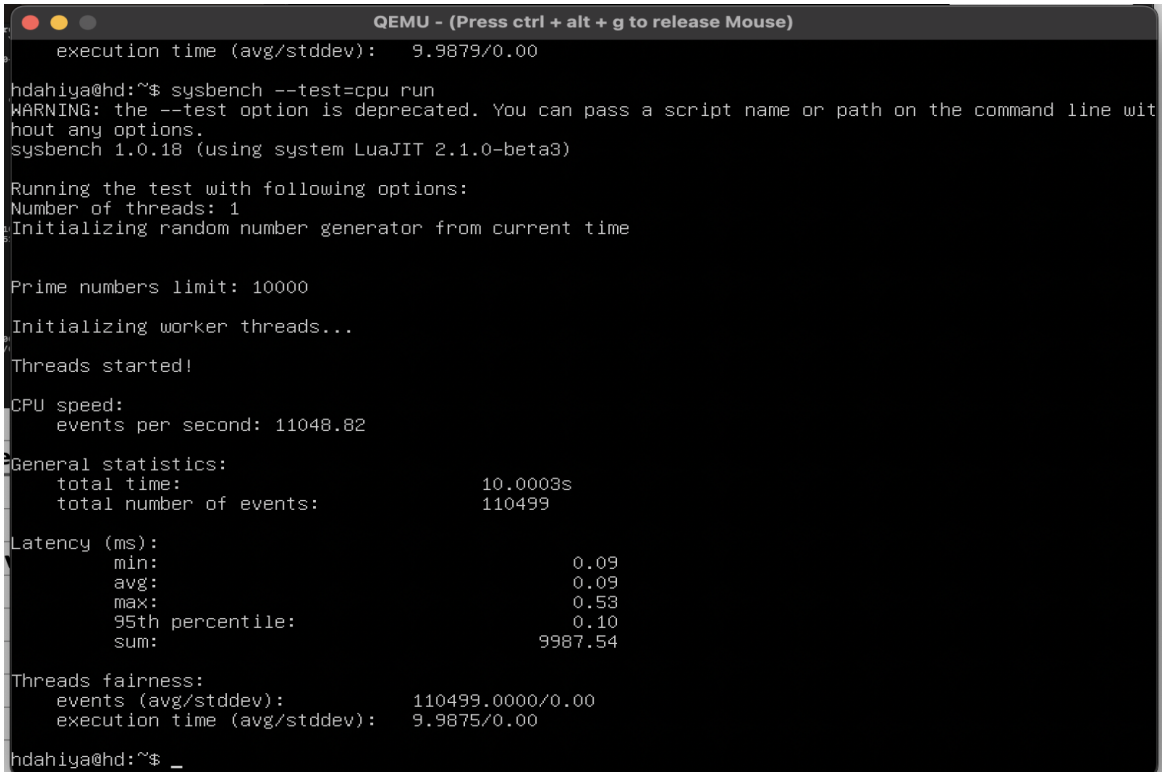
Run 1



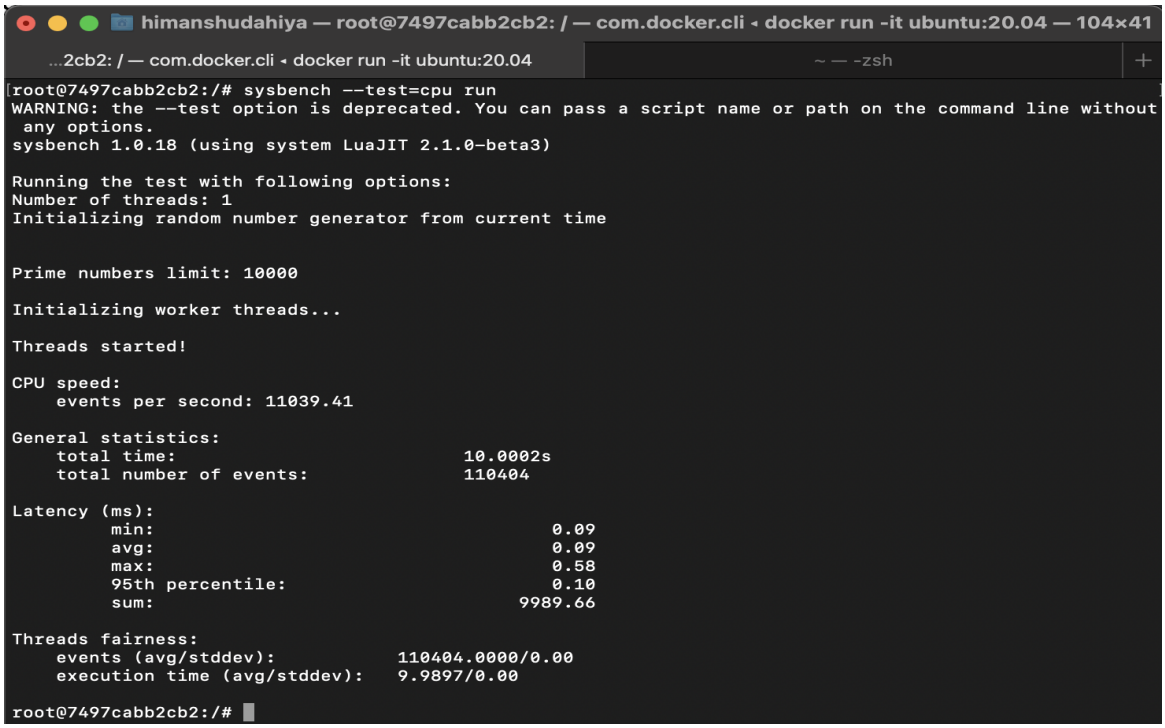

QEMU - (Press ctrl + alt + g to release Mouse)
Last login: Sun Feb 5 02:19:00 UTC 2023 on tty1
hdahiya@hd:~$ sysbench --test=cpu run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line wit
hout any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 11063.49
General statistics:
total time: 10.0001s
total number of events: 110642
Latency (ms):
min: 0.09
avg: 0.09
max: 0.70
95th percentile: 0.10
sum: 9987.94
Threads fairness:
events (avg/stddev): 110642.0000/0.00
execution time (avg/stddev): 9.9879/0.00
hdahiya@hd:~$

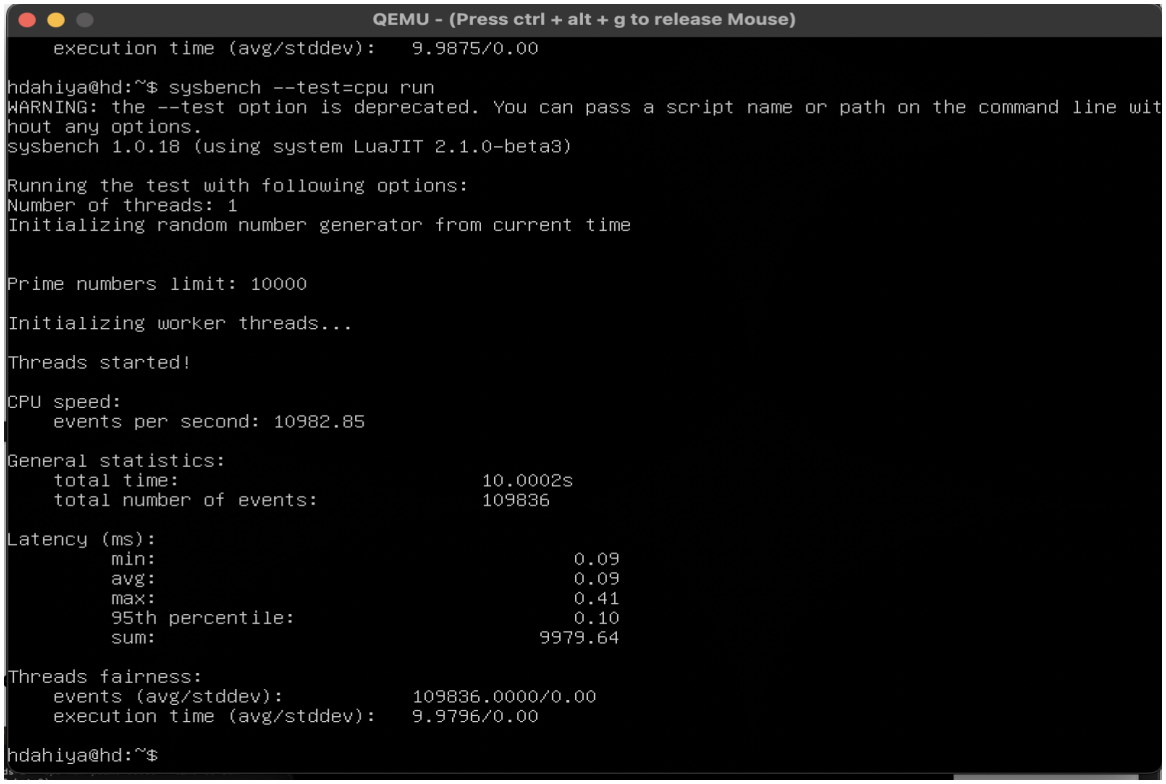
Run 2


root@7497cabb2cb2:/# sysbench --test=cpu run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without
any options.
sysbench 1.0.18 (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: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 11034.85
General statistics:
total time: 10.0002s
total number of events: 110360
Latency (ms):
min: 0.09
avg: 0.09
max: 0.29
95th percentile: 0.10
sum: 9986.95
Threads fairness:
events (avg/stddev): 110360.0000/0.00
execution time (avg/stddev): 9.9870/0.00
root@7497cabb2cb2:/#

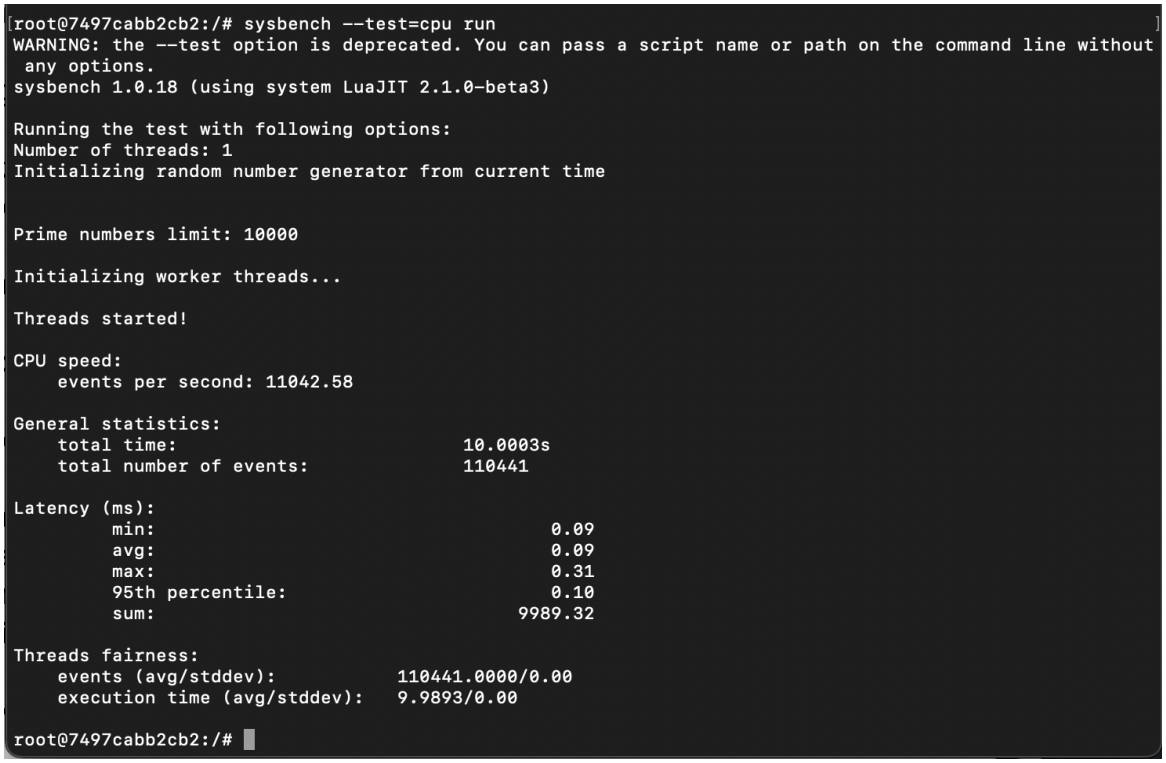

QEMU - (Press ctrl + alt + g to release Mouse)
execution time (avg/stddev): 9.9879/0.00
hdahiya@hd:~$ sysbench --test=cpu run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line wit
hout any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 11048.82
General statistics:
total time: 10.0003s
total number of events: 110499
Latency (ms):
min: 0.09
avg: 0.09
max: 0.53
95th percentile: 0.10
sum: 9987.54
Threads fairness:
events (avg/stddev): 110499.0000/0.00
execution time (avg/stddev): 9.9875/0.00
hdahiya@hd:~$ _

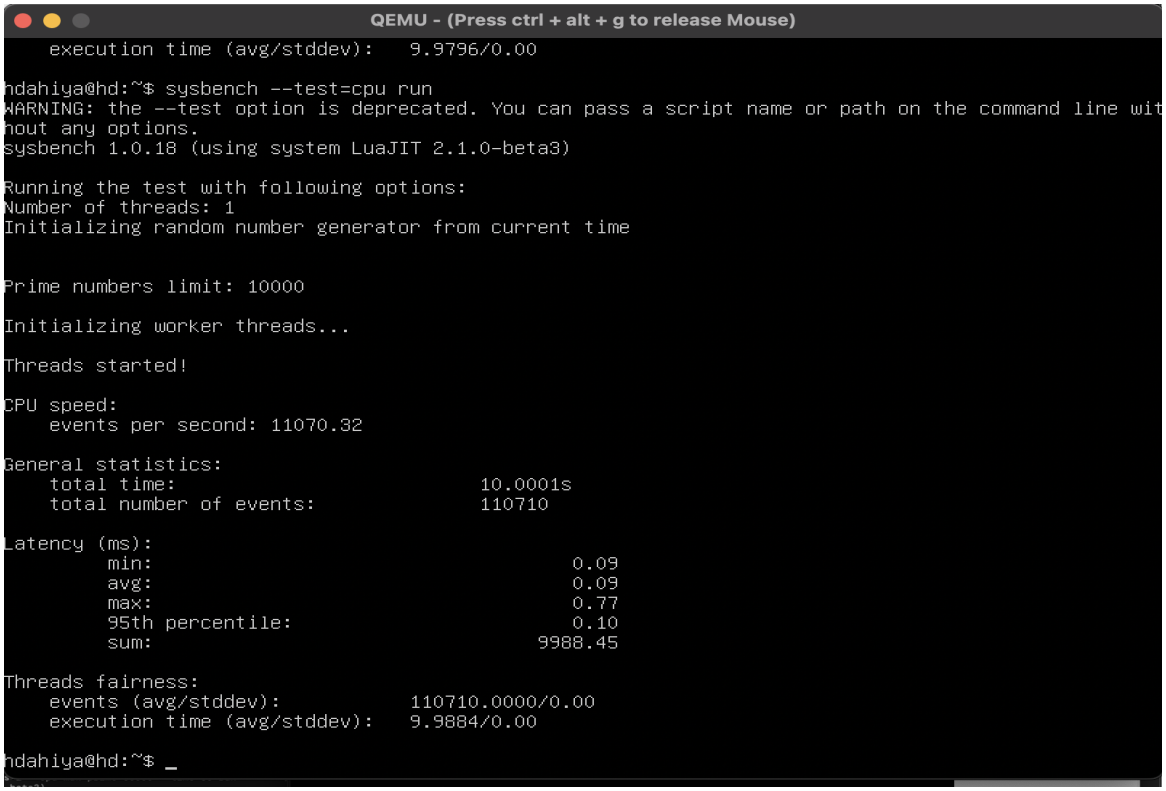
Run 3


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 104×41
...2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench --test=cpu run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without
any options.
sysbench 1.0.18 (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: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 11039.41
General statistics:
total time: 10.0002s
total number of events: 110404
Latency (ms):
min: 0.09
avg: 0.09
max: 0.58
95th percentile: 0.10
sum: 9989.66
Threads fairness:
events (avg/stddev): 110404.0000/0.00
execution time (avg/stddev): 9.9897/0.00
root@7497cabb2cb2:/#

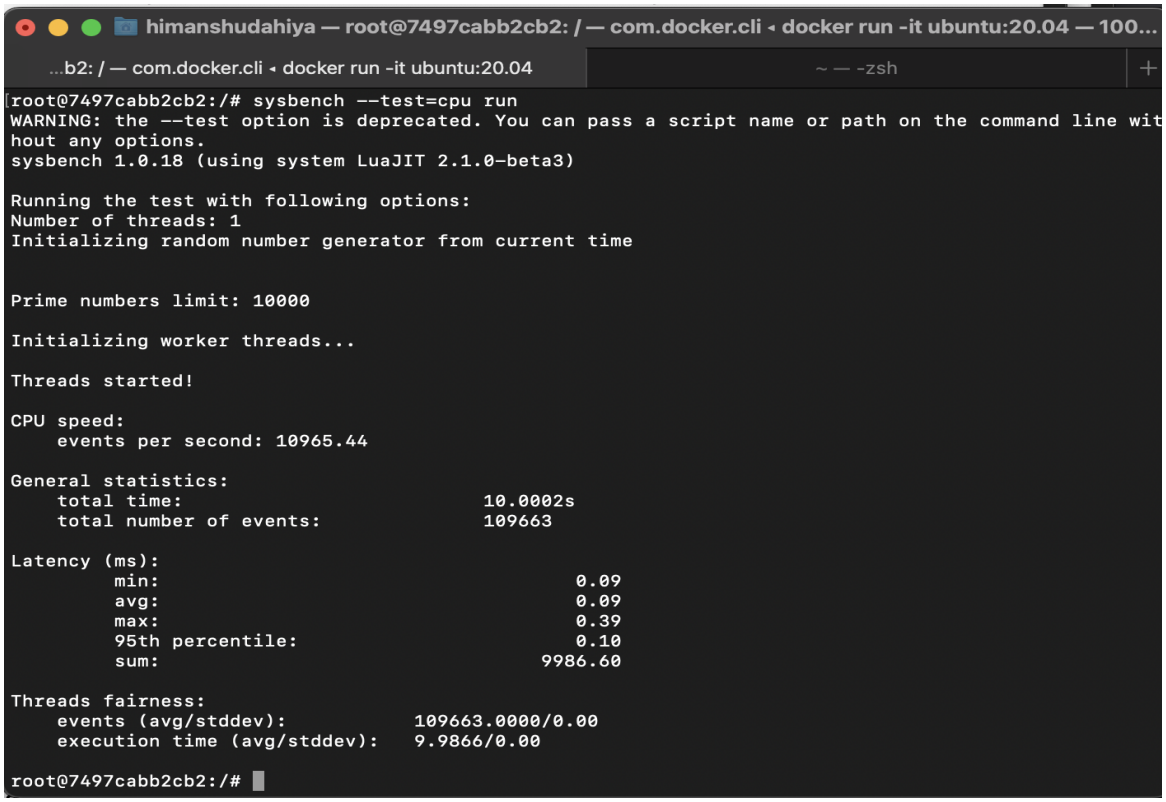

QEMU - (Press ctrl + alt + g to release Mouse)
execution time (avg/stddev): 9.9875/0.00
hdahiya@hd:~$ sysbench --test=cpu run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line wit
hout any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 10982.85
General statistics:
total time: 10.0002s
total number of events: 109836
Latency (ms):
min: 0.09
avg: 0.09
max: 0.41
95th percentile: 0.10
sum: 9979.64
Threads fairness:
events (avg/stddev): 109836.0000/0.00
execution time (avg/stddev): 9.9796/0.00
hdahiya@hd:~$

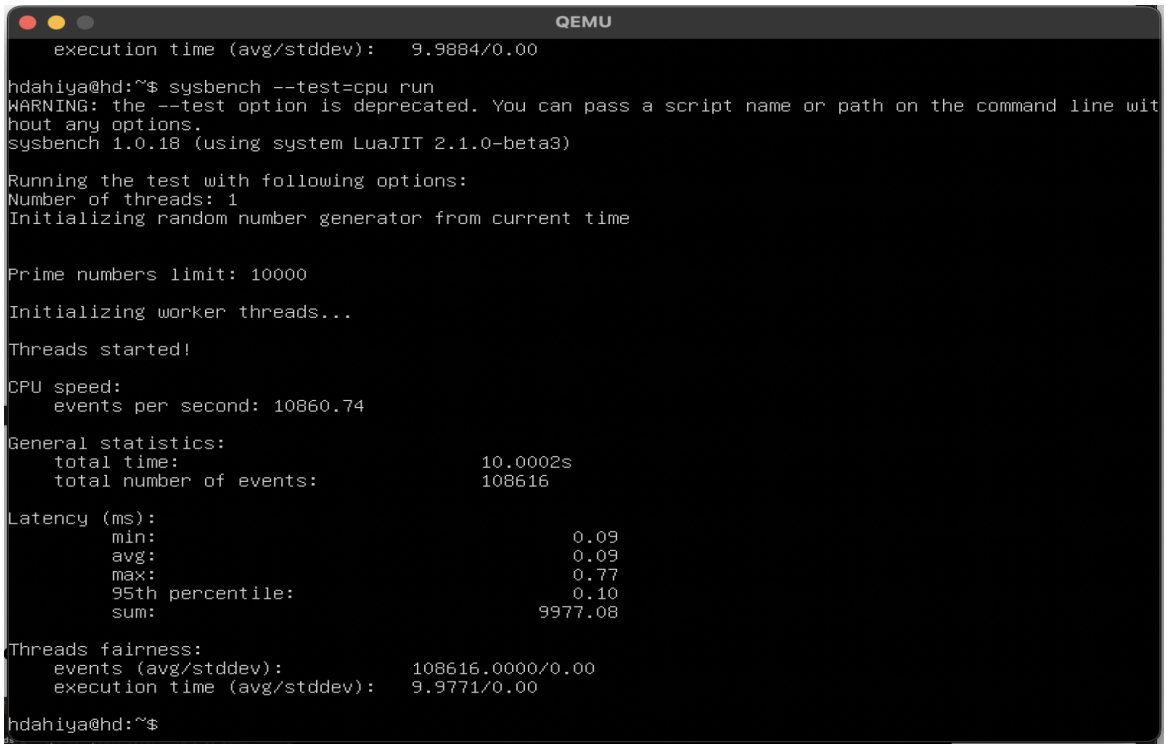
Run 4


root@7497cabb2cb2:/# sysbench --test=cpu run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without
any options.
sysbench 1.0.18 (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: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 11042.58
General statistics:
total time: 10.0003s
total number of events: 110441
Latency (ms):
min: 0.09
avg: 0.09
max: 0.31
95th percentile: 0.10
sum: 9989.32
Threads fairness:
events (avg/stddev): 110441.0000/0.00
execution time (avg/stddev): 9.9893/0.00
root@7497cabb2cb2:/#

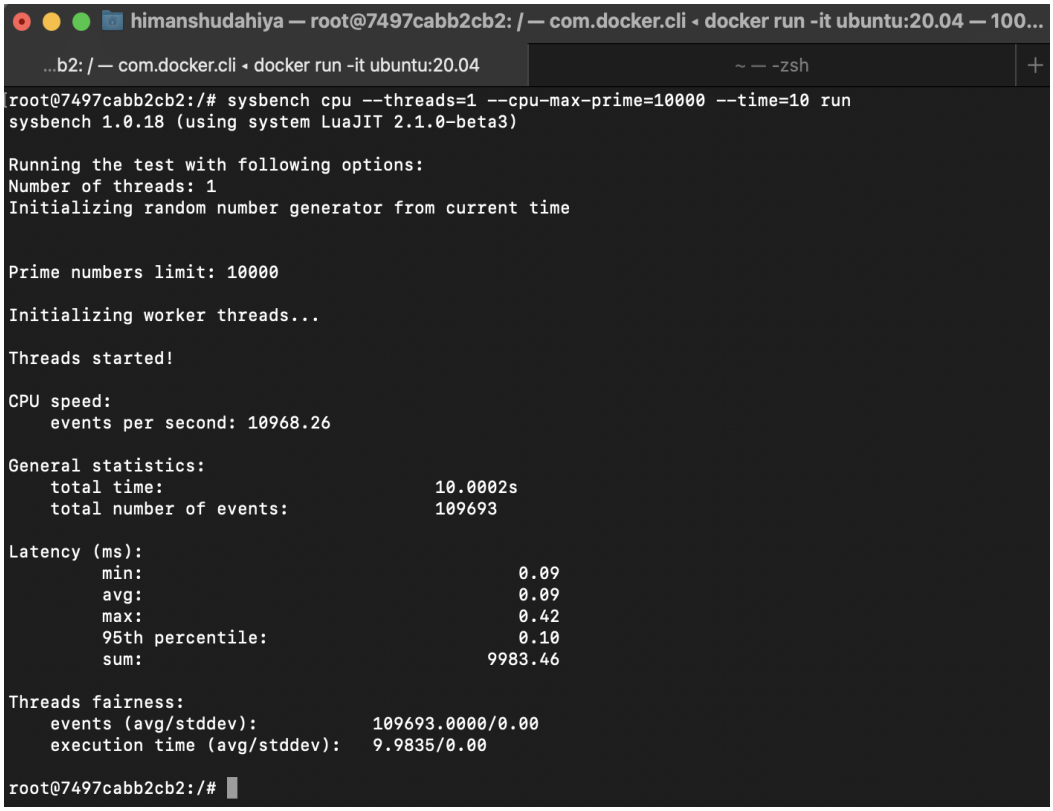

QEMU - (Press ctrl + alt + g to release Mouse)
execution time (avg/stddev): 9.9796/0.00
hdahiya@hd:~$ sysbench --test=cpu run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line wit
hout any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 11070.32
General statistics:
total time: 10.0001s
total number of events: 110710
Latency (ms):
min: 0.09
avg: 0.09
max: 0.77
95th percentile: 0.10
sum: 9988.45
Threads fairness:
events (avg/stddev): 110710.0000/0.00
execution time (avg/stddev): 9.9884/0.00
hdahiya@hd:~$

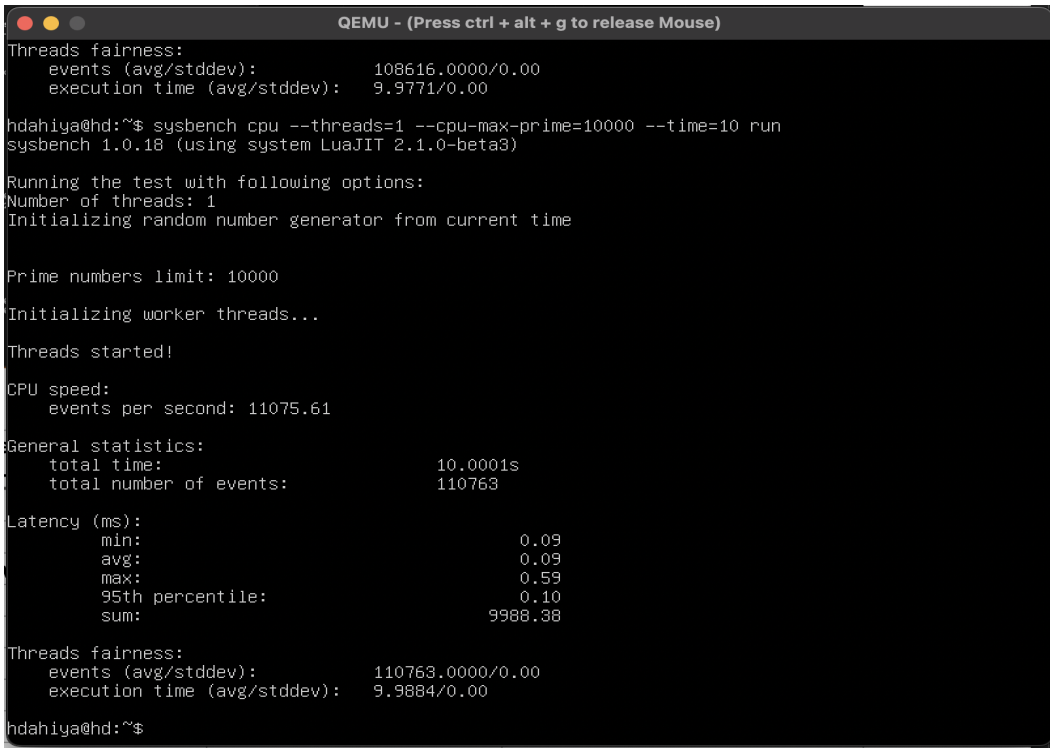
Run 5


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 100...
..b2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench --test=cpu run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line wit
hout any options.
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: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 10965.44
General statistics:
total time: 10.0002s
total number of events: 109663
Latency (ms):
min: 0.09
avg: 0.09
max: 0.39
95th percentile: 0.10
sum: 9986.60
Threads fairness:
events (avg/stddev): 109663.0000/0.00
execution time (avg/stddev): 9.9866/0.00
root@7497cabb2cb2:/#

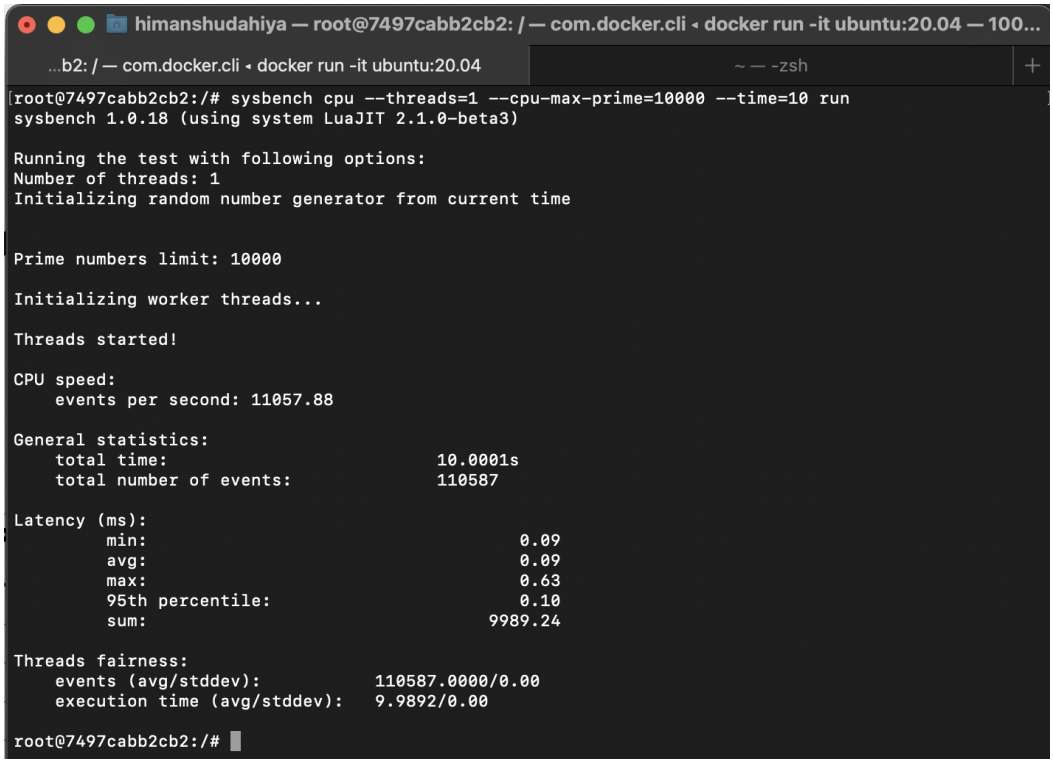

QEMU
execution time (avg/stddev): 9.9884/0.00
hdahiya@hd:~$ sysbench --test=cpu run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line wit
hout any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 10860.74
General statistics:
total time: 10.0002s
total number of events: 108616
Latency (ms):
min: 0.09
avg: 0.09
max: 0.77
95th percentile: 0.10
sum: 9977.08
Threads fairness:
events (avg/stddev): 108616.0000/0.00
execution time (avg/stddev): 9.9771/0.00
hdahiya@hd:~$

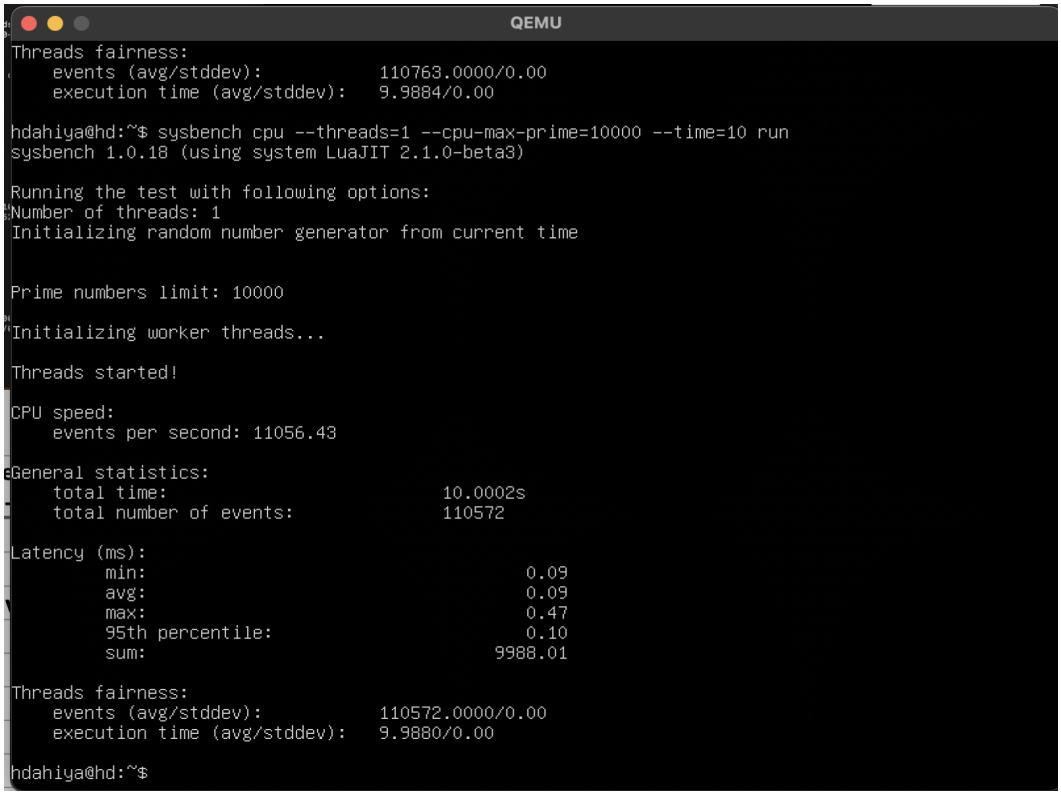
Testcase 2
sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
Run 1



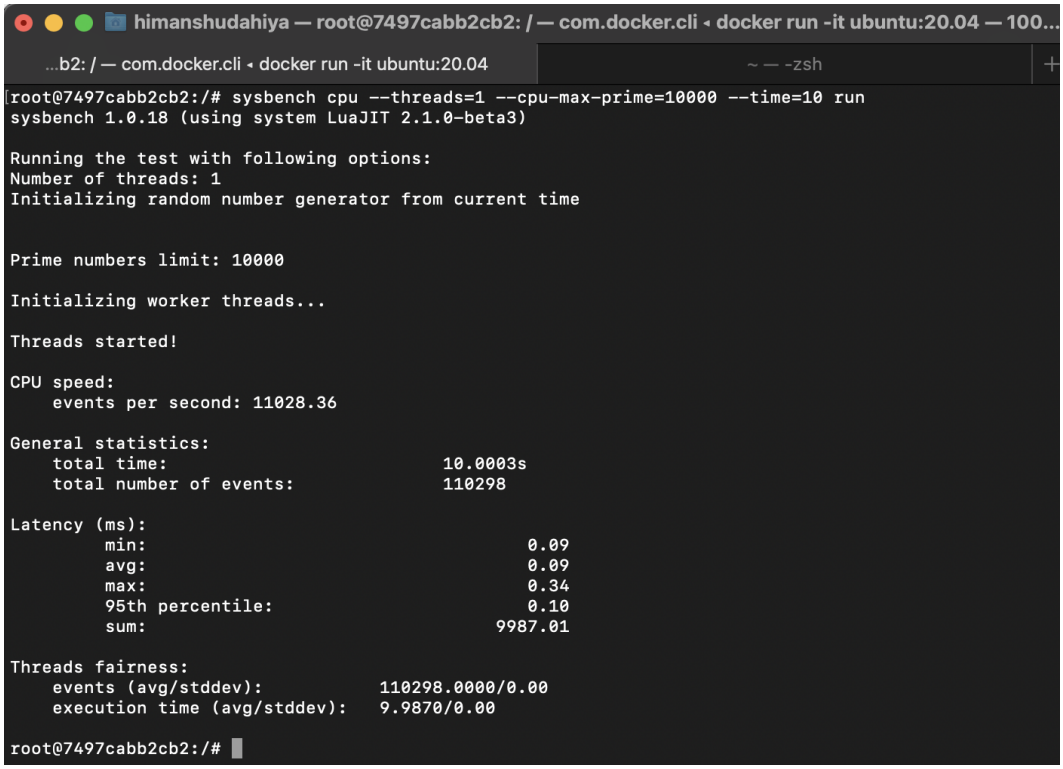


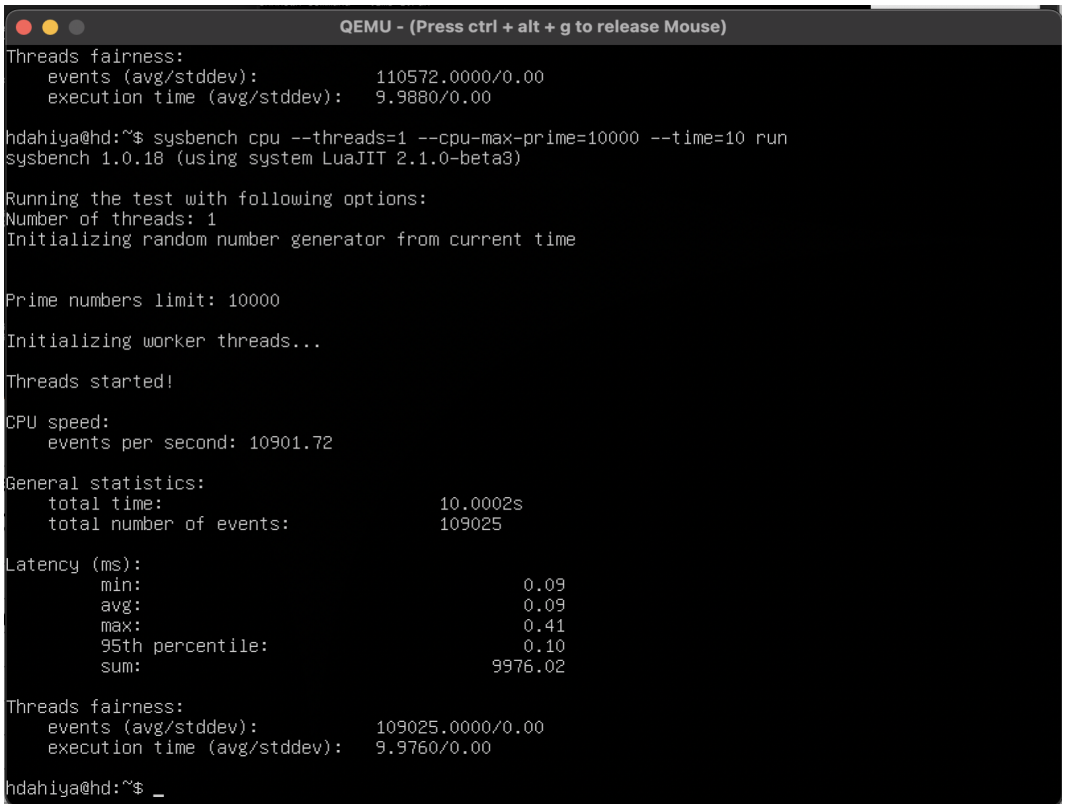
Run 2


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 100...
...b2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
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: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 11057.88
General statistics:
total time: 10.0001s
total number of events: 110587
Latency (ms):
min: 0.09
avg: 0.09
max: 0.63
95th percentile: 0.10
sum: 9989.24
Threads fairness:
events (avg/stddev): 110587.0000/0.00
execution time (avg/stddev): 9.9892/0.00
root@7497cabb2cb2:/#

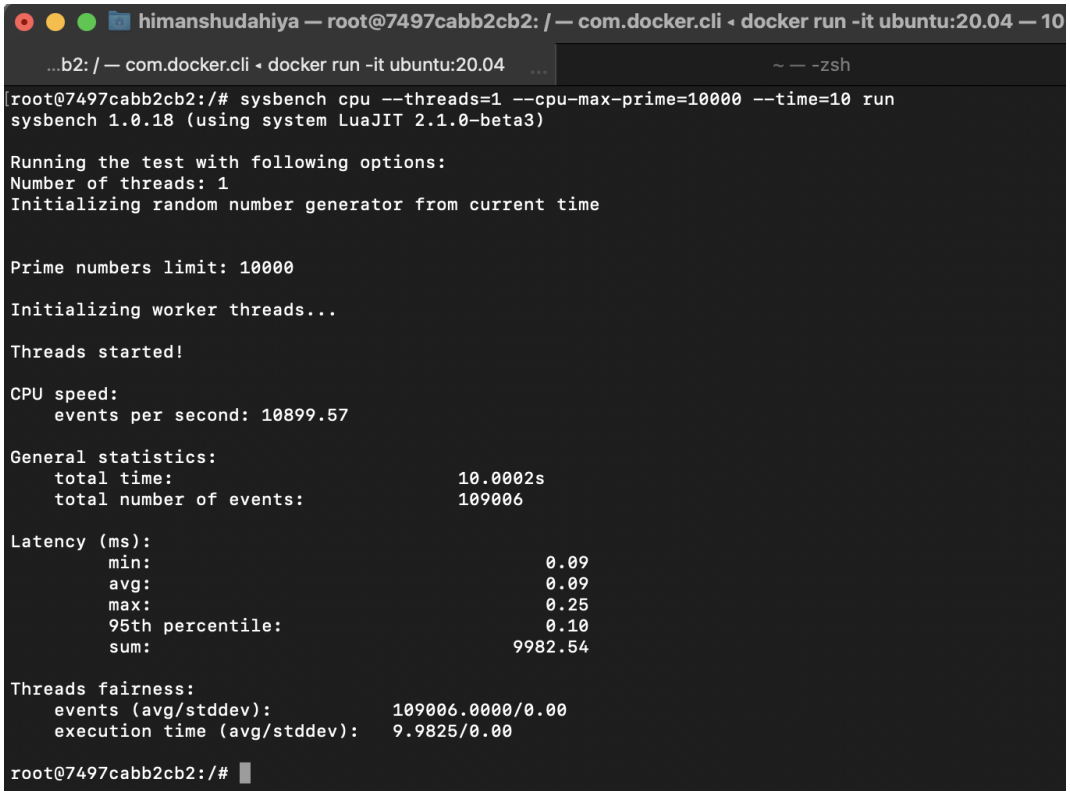

QEMU
Threads fairness:
events (avg/stddev): 110763.0000/0.00
execution time (avg/stddev): 9.9884/0.00
hdahiya@hd:~$ sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 11056.43
General statistics:
total time: 10.0002s
total number of events: 110572
Latency (ms):
min: 0.09
avg: 0.09
max: 0.47
95th percentile: 0.10
sum: 9988.01
Threads fairness:
events (avg/stddev): 110572.0000/0.00
execution time (avg/stddev): 9.9880/0.00
hdahiya@hd:~$

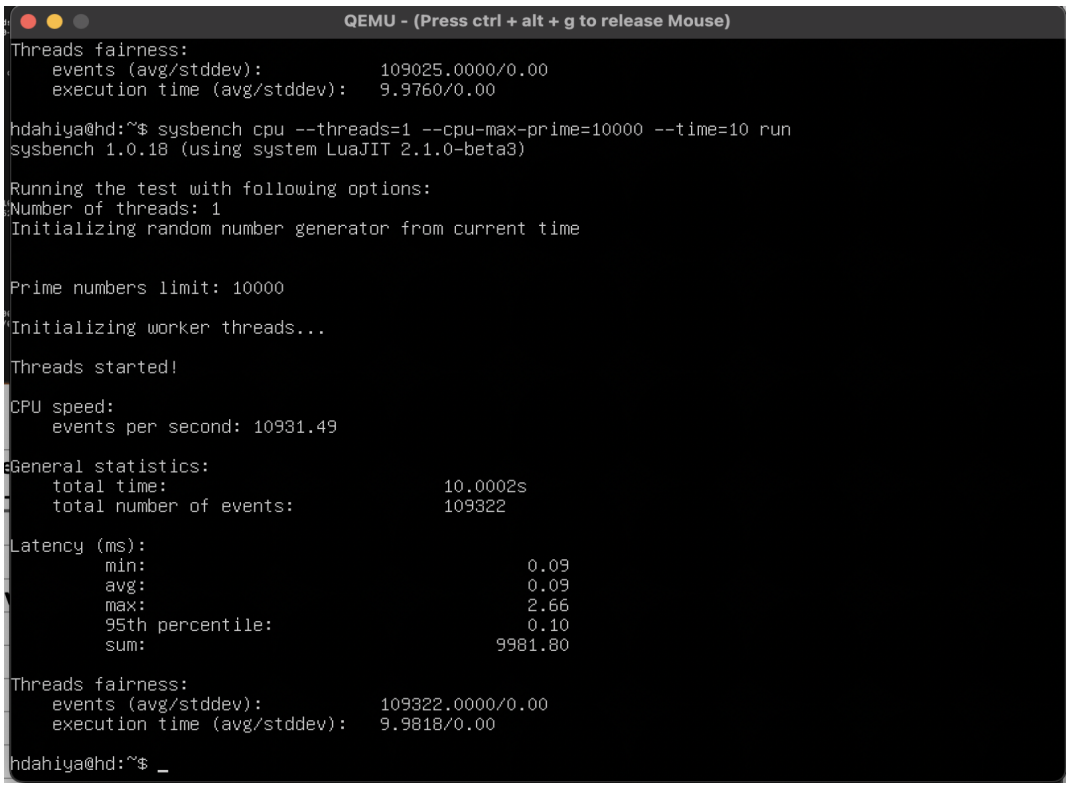
Run 3


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 100...
...b2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
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: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 11028.36
General statistics:
total time: 10.0003s
total number of events: 110298
Latency (ms):
min: 0.09
avg: 0.09
max: 0.34
95th percentile: 0.10
sum: 9987.01
Threads fairness:
events (avg/stddev): 110298.0000/0.00
execution time (avg/stddev): 9.9870/0.00
root@7497cabb2cb2:/#

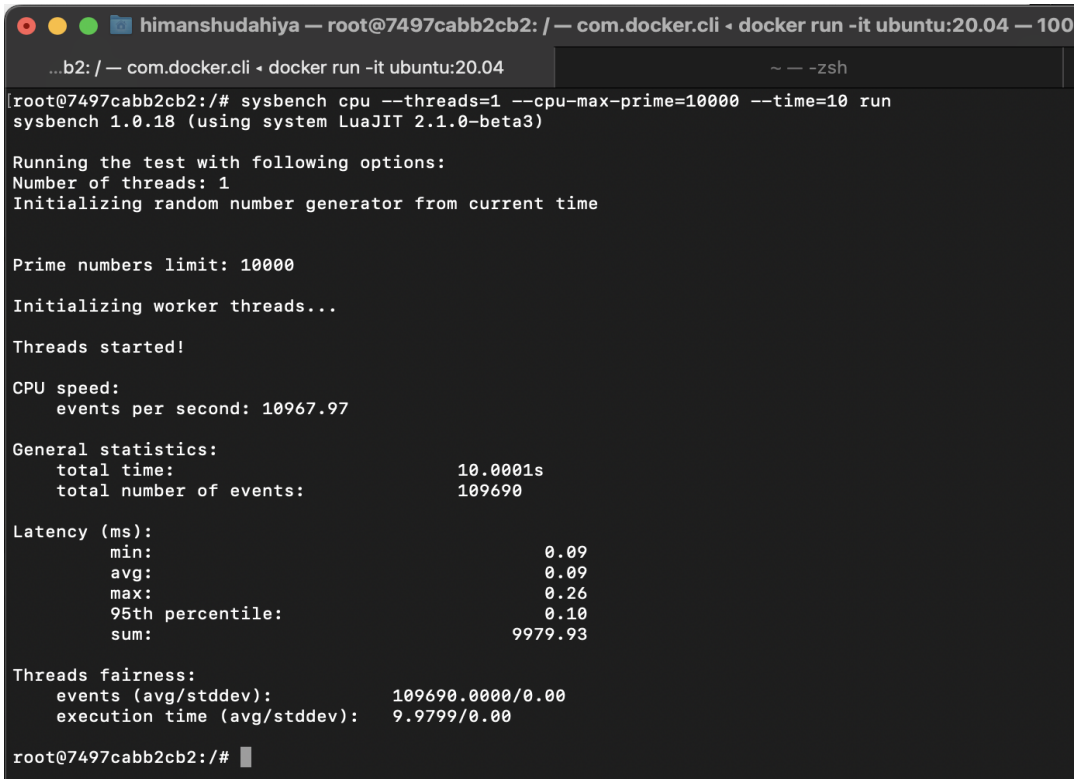

QEMU - (Press ctrl + alt + g to release Mouse)
Threads fairness:
events (avg/stddev): 110572.0000/0.00
execution time (avg/stddev): 9.9880/0.00
hdahiya@hd:~$ sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 10901.72
General statistics:
total time: 10.0002s
total number of events: 109025
Latency (ms):
min: 0.09
avg: 0.09
max: 0.41
95th percentile: 0.10
sum: 9976.02
Threads fairness:
events (avg/stddev): 109025.0000/0.00
execution time (avg/stddev): 9.9760/0.00
hdahiya@hd:~$

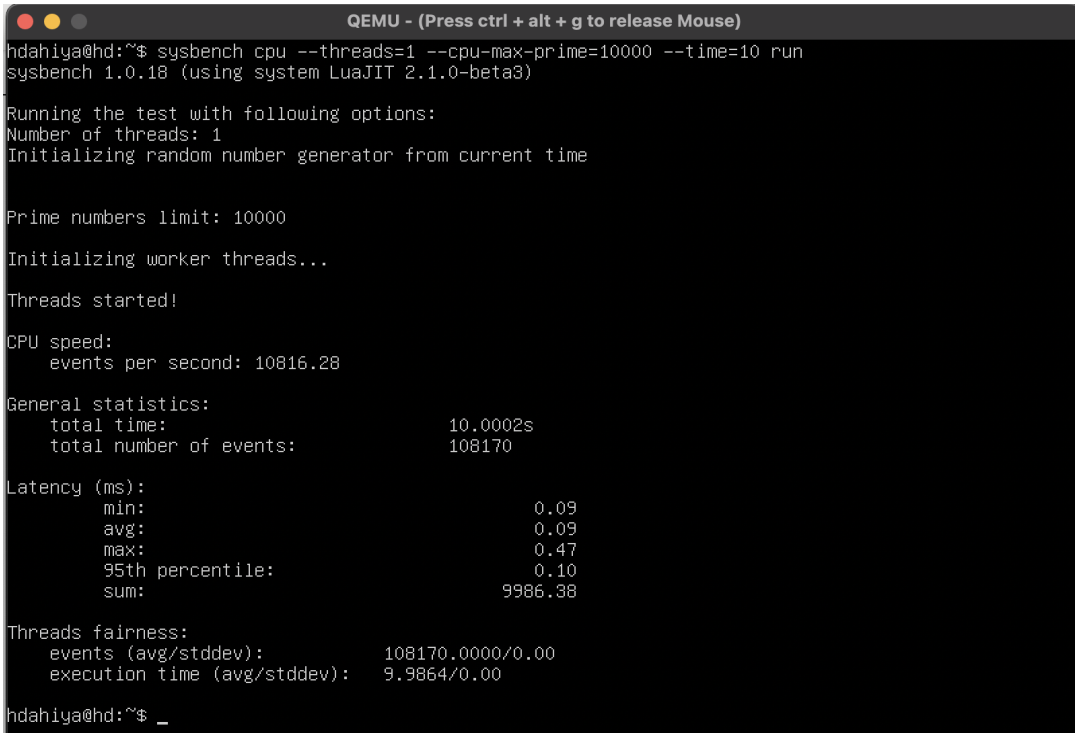
Run 4


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 10
...b2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
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: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 10899.57
General statistics:
total time: 10.0002s
total number of events: 109006
Latency (ms):
min: 0.09
avg: 0.09
max: 0.25
95th percentile: 0.10
sum: 9982.54
Threads fairness:
events (avg/stddev): 109006.0000/0.00
execution time (avg/stddev): 9.9825/0.00
root@7497cabb2cb2:/#

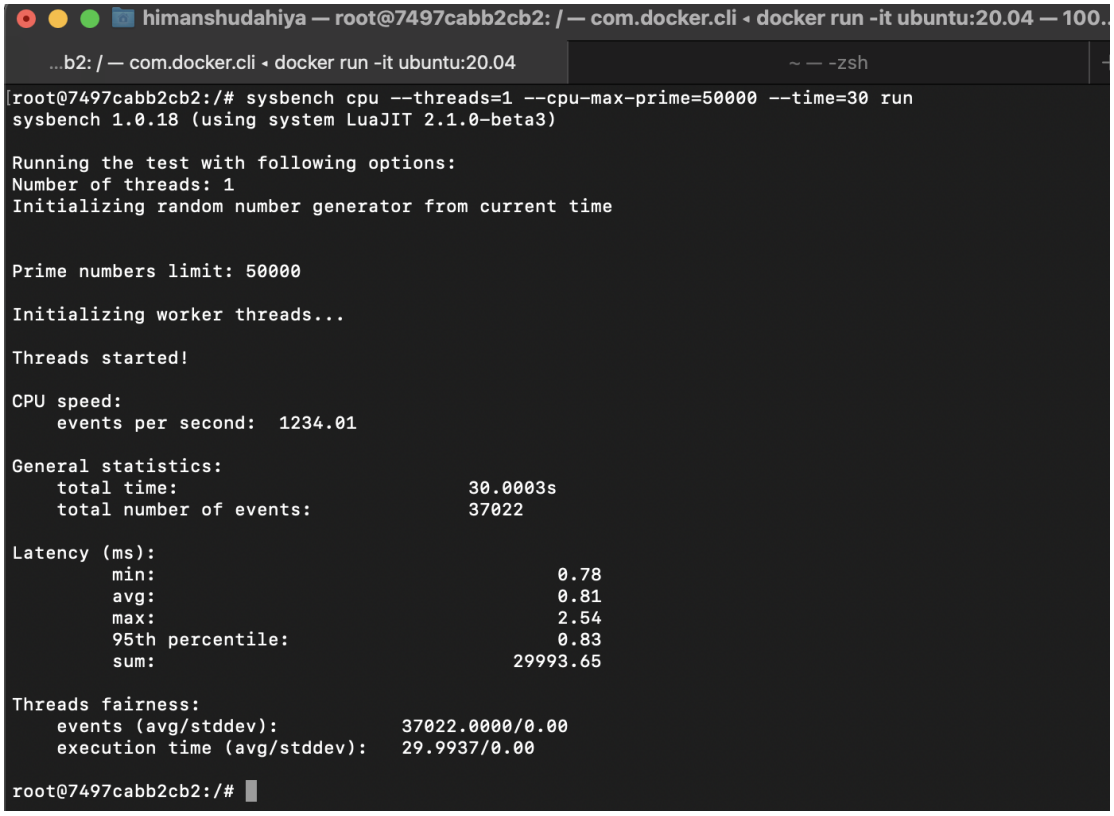

QEMU - (Press ctrl + alt + g to release Mouse)
Threads fairness:
events (avg/stddev): 109025.0000/0.00
execution time (avg/stddev): 9.9760/0.00
hdahiya@hd:~$ sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 10931.49
General statistics:
total time: 10.0002s
total number of events: 109322
Latency (ms):
min: 0.09
avg: 0.09
max: 2.66
95th percentile: 0.10
sum: 9981.80
Threads fairness:
events (avg/stddev): 109322.0000/0.00
execution time (avg/stddev): 9.9818/0.00
hdahiya@hd:~$

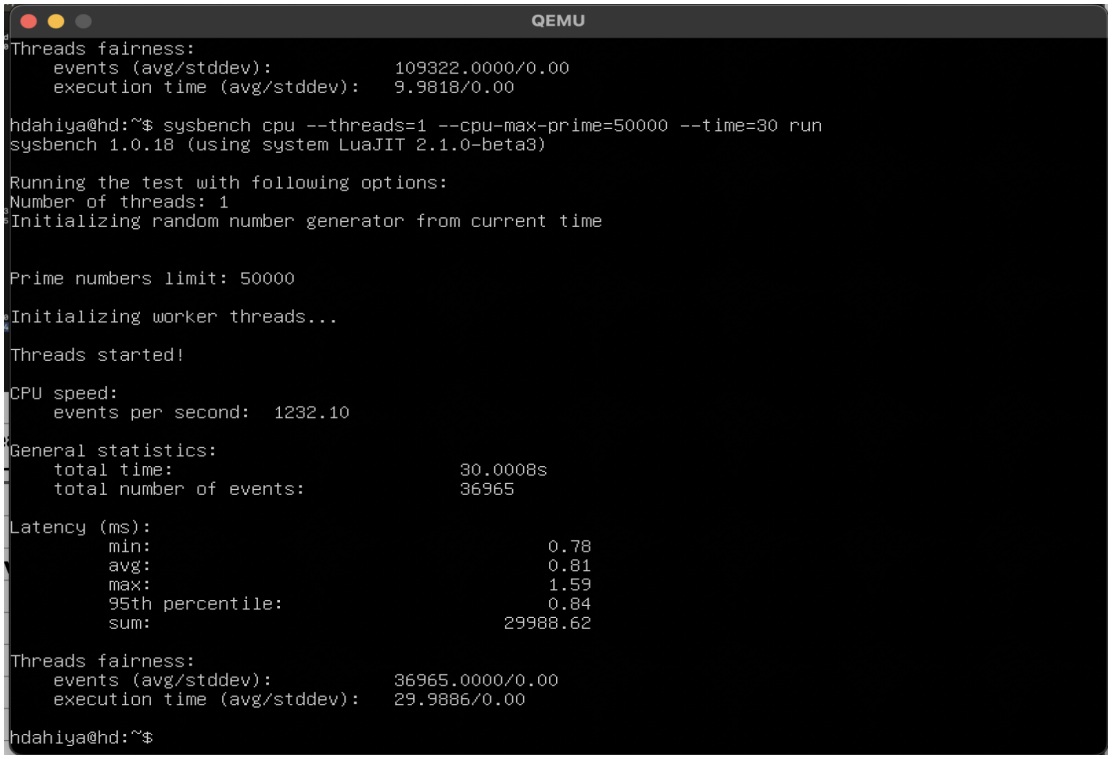
Run 5


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 100
...b2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
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: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 10967.97
General statistics:
total time: 10.0001s
total number of events: 109690
Latency (ms):
min: 0.09
avg: 0.09
max: 0.26
95th percentile: 0.10
sum: 9979.93
Threads fairness:
events (avg/stddev): 109690.0000/0.00
execution time (avg/stddev): 9.9799/0.00
root@7497cabb2cb2:/#

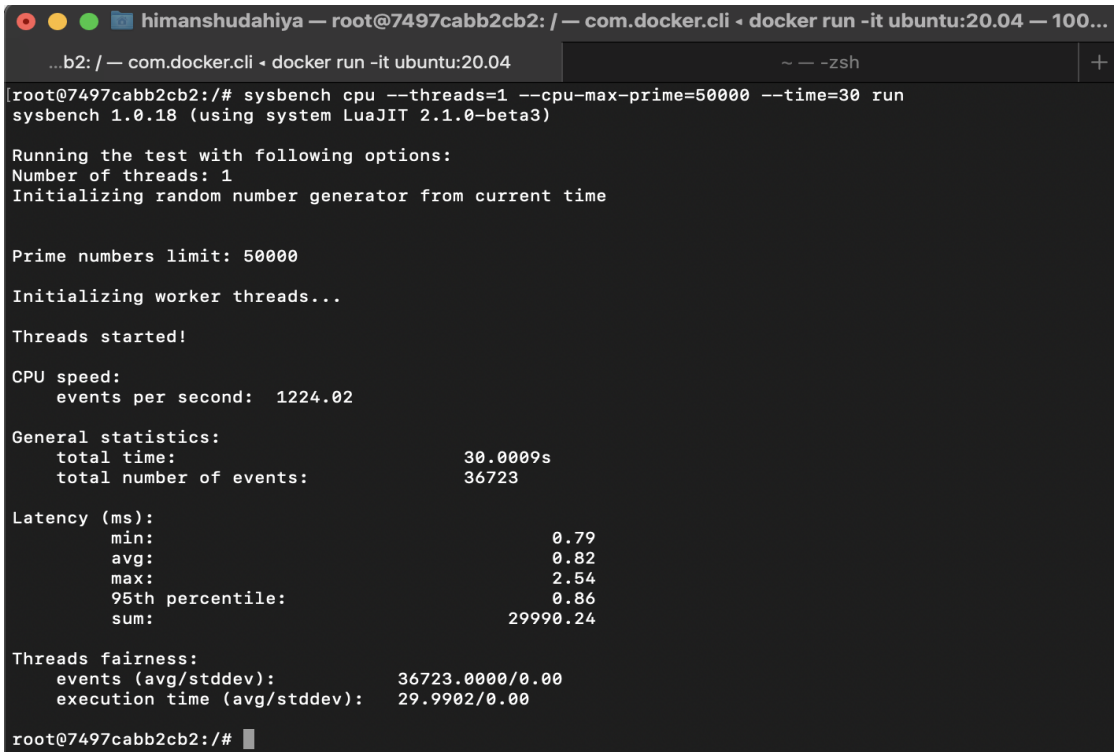

QEMU - (Press ctrl + alt + g to release Mouse)
hdahiya@hd:~$ sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 10000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 10816.28
General statistics:
total time: 10.0002s
total number of events: 108170
Latency (ms):
min: 0.09
avg: 0.09
max: 0.47
95th percentile: 0.10
sum: 9986.38
Threads fairness:
events (avg/stddev): 108170.0000/0.00
execution time (avg/stddev): 9.9864/0.00
hdahiya@hd:~$

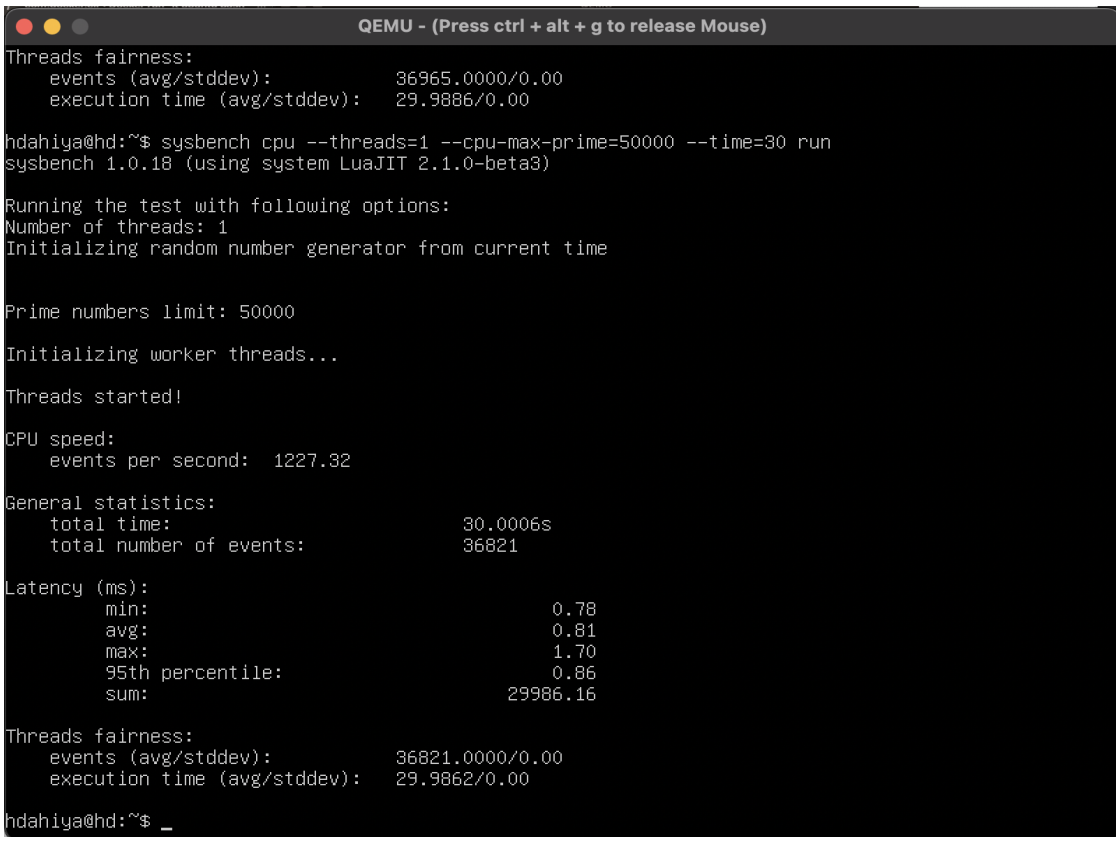
Testcase 3
sysbench cpu –threads=1 --cpu-max-prime=50000 --time=30 run
Run 1



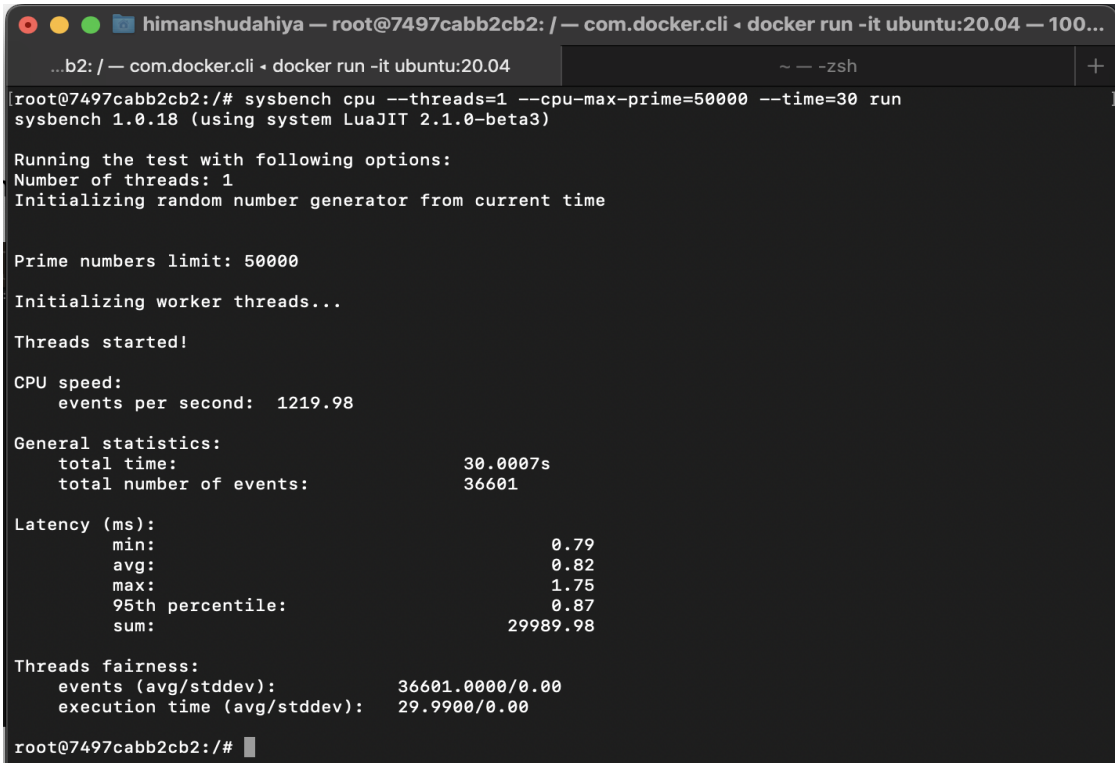


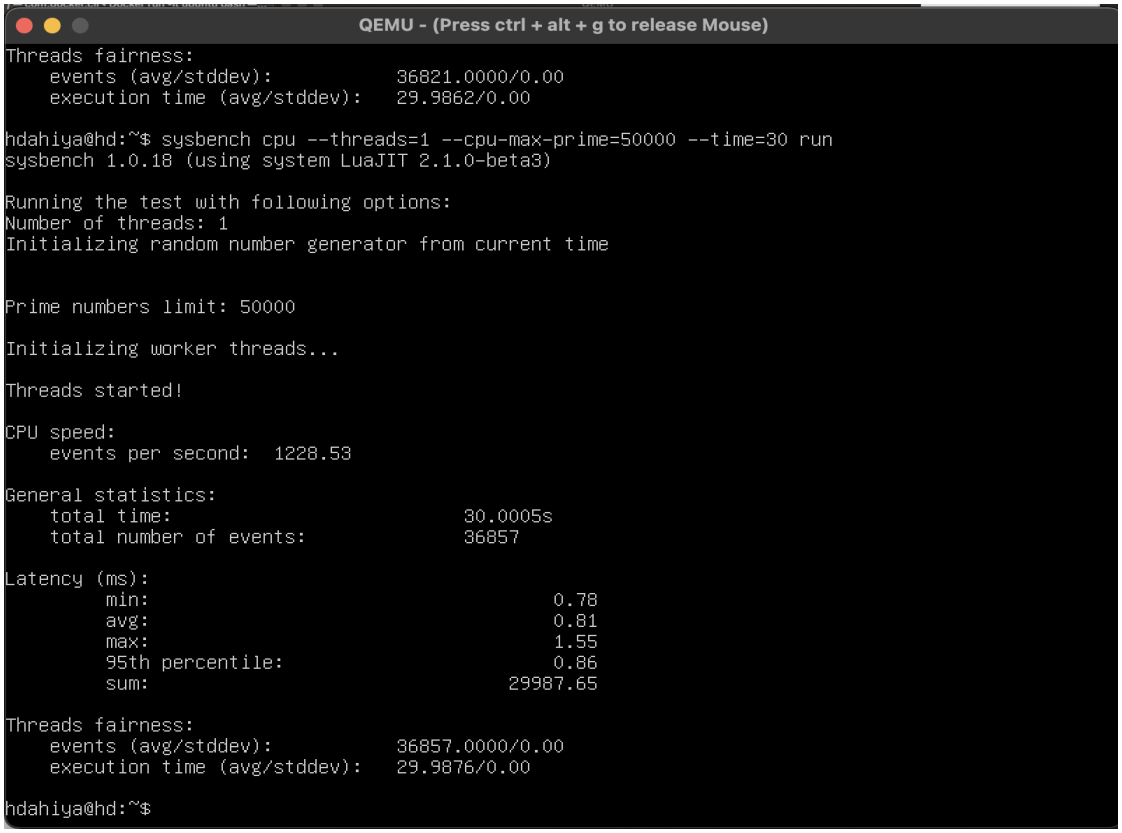
Run 2


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 100...
...b2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench cpu --threads=1 --cpu-max-prime=50000 --time=30 run
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: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 1224.02
General statistics:
total time: 30.0009s
total number of events: 36723
Latency (ms):
min: 0.79
avg: 0.82
max: 2.54
95th percentile: 0.86
sum: 29990.24
Threads fairness:
events (avg/stddev): 36723.0000/0.00
execution time (avg/stddev): 29.9902/0.00
root@7497cabb2cb2:/#

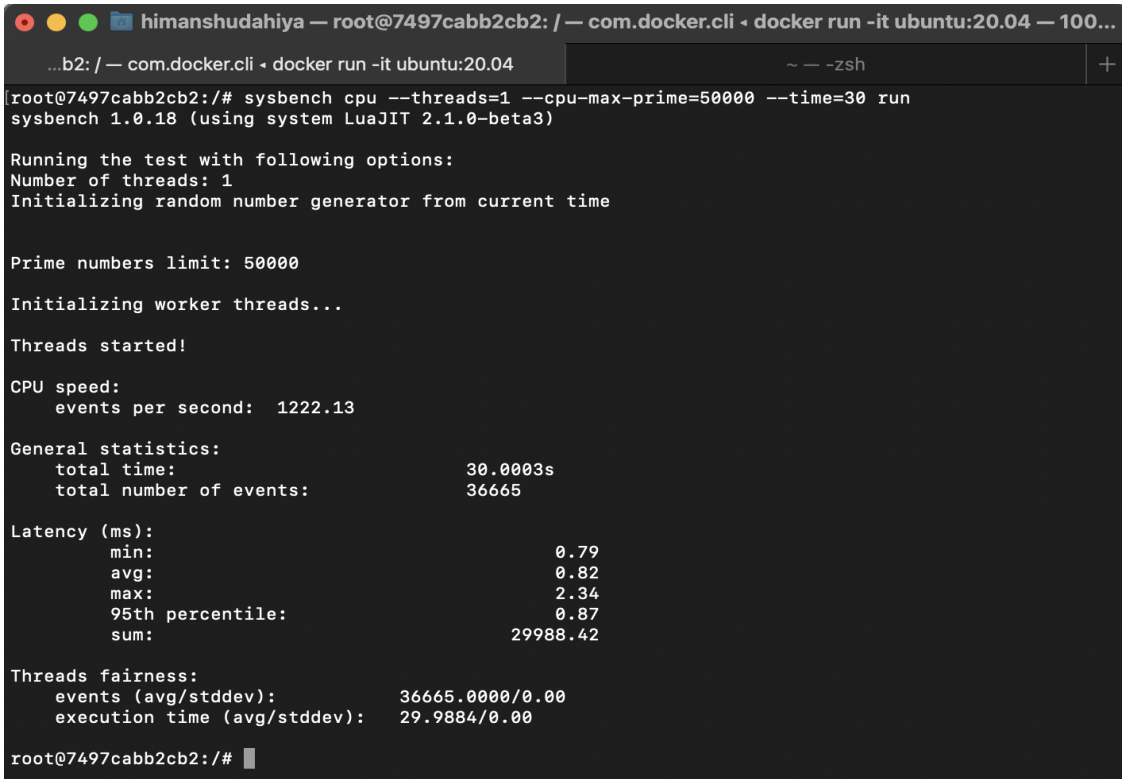

QEMU - (Press ctrl + alt + g to release Mouse)
Threads fairness:
events (avg/stddev): 36965.0000/0.00
execution time (avg/stddev): 29.9886/0.00
hdahiya@hd:~$ sysbench cpu --threads=1 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 1227.32
General statistics:
total time: 30.0006s
total number of events: 36821
Latency (ms):
min: 0.78
avg: 0.81
max: 1.70
95th percentile: 0.86
sum: 29986.16
Threads fairness:
events (avg/stddev): 36821.0000/0.00
execution time (avg/stddev): 29.9862/0.00
hdahiya@hd:~$

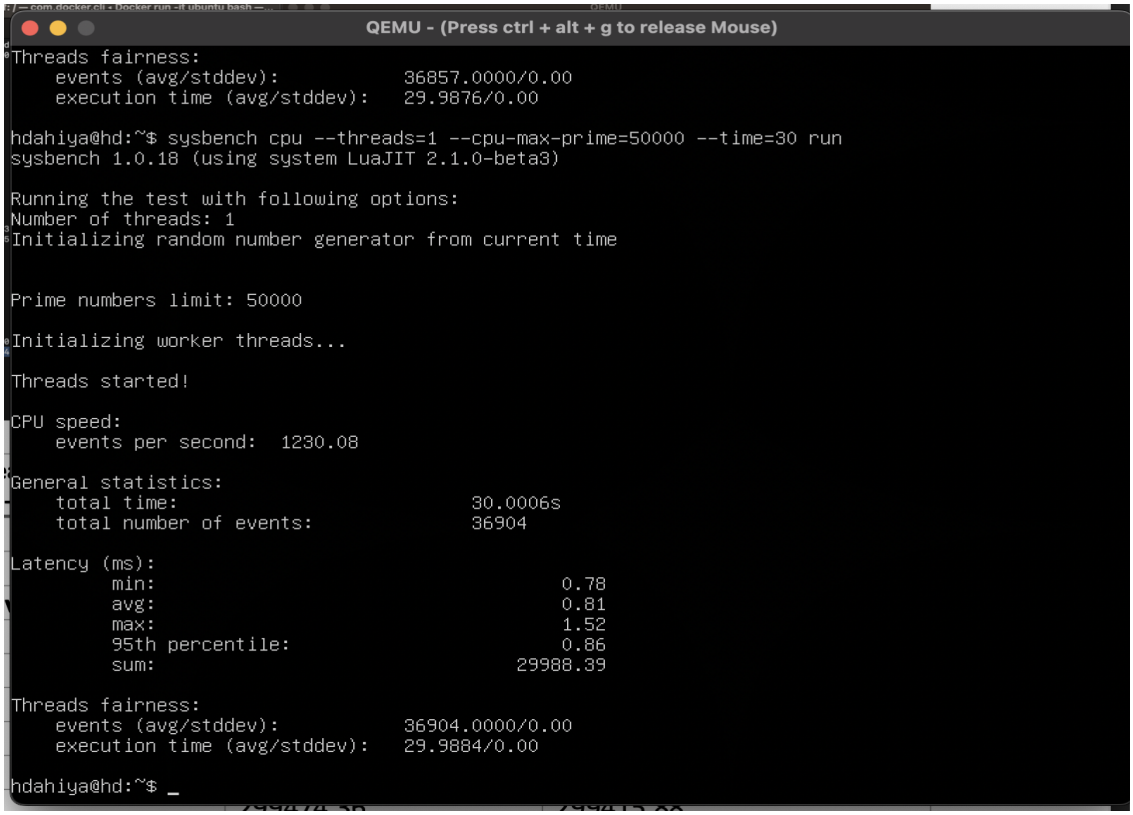
Run 3


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 100...
...b2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench cpu --threads=1 --cpu-max-prime=50000 --time=30 run
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: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 1219.98
General statistics:
total time: 30.0007s
total number of events: 36601
Latency (ms):
min: 0.79
avg: 0.82
max: 1.75
95th percentile: 0.87
sum: 29989.98
Threads fairness:
events (avg/stddev): 36601.0000/0.00
execution time (avg/stddev): 29.9900/0.00
root@7497cabb2cb2:/#

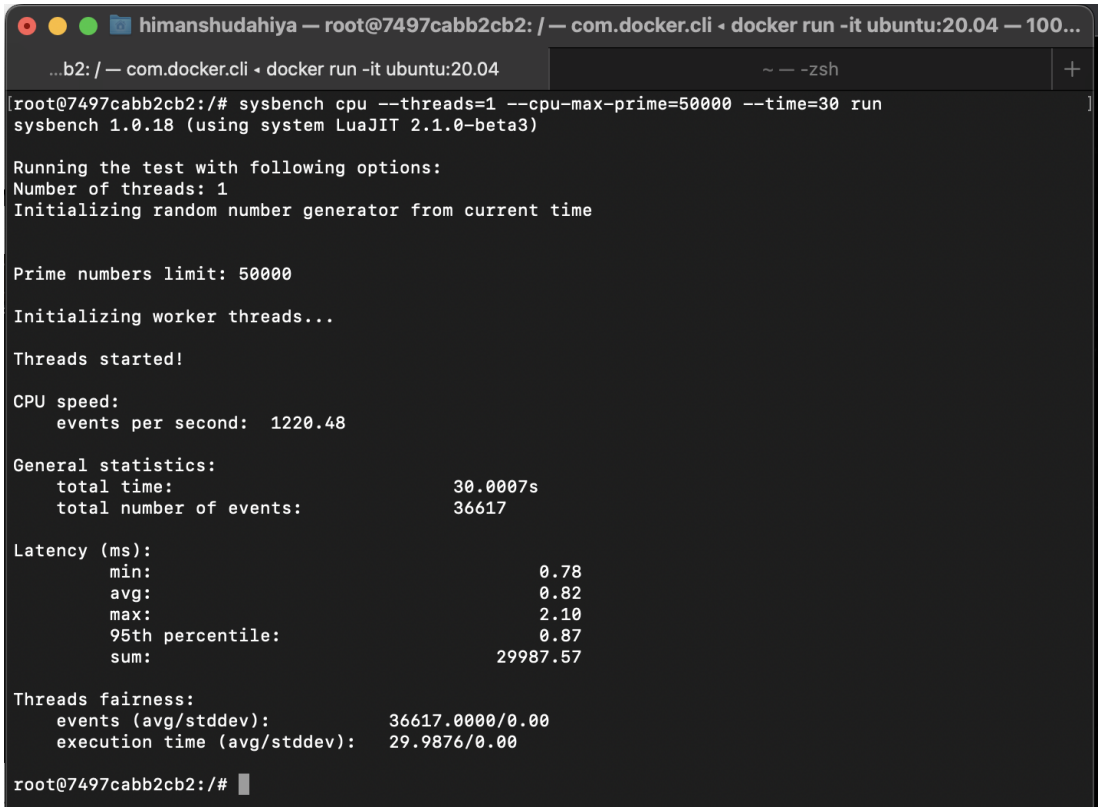

QEMU - (Press ctrl + alt + g to release Mouse)
Threads fairness:
events (avg/stddev): 36821.0000/0.00
execution time (avg/stddev): 29.9862/0.00
hdahiya@hd:~$ sysbench cpu --threads=1 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 1228.53
General statistics:
total time: 30.0005s
total number of events: 36857
Latency (ms):
min: 0.78
avg: 0.81
max: 1.55
95th percentile: 0.86
sum: 29987.65
Threads fairness:
events (avg/stddev): 36857.0000/0.00
execution time (avg/stddev): 29.9876/0.00
hdahiya@hd:~$

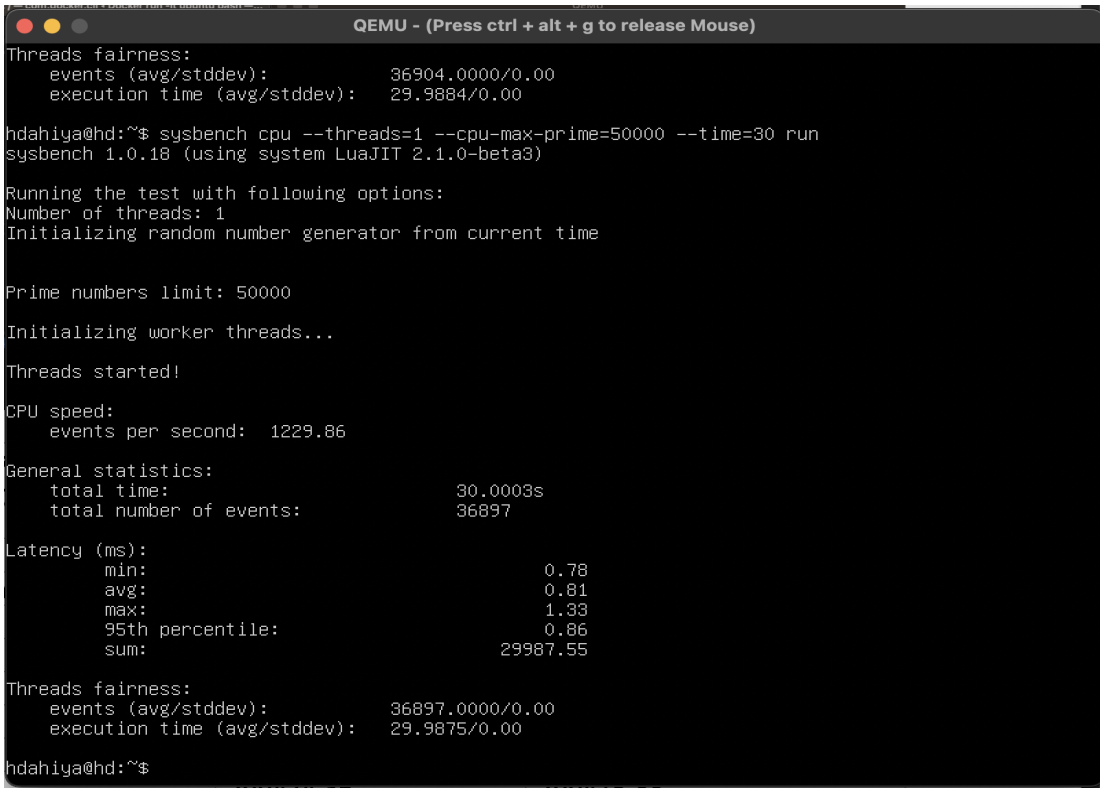
Run 4


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 100...
...b2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench cpu --threads=1 --cpu-max-prime=50000 --time=30 run
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: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 1222.13
General statistics:
total time: 30.0003s
total number of events: 36665
Latency (ms):
min: 0.79
avg: 0.82
max: 2.34
95th percentile: 0.87
sum: 29988.42
Threads fairness:
events (avg/stddev): 36665.0000/0.00
execution time (avg/stddev): 29.9884/0.00
root@7497cabb2cb2:/#

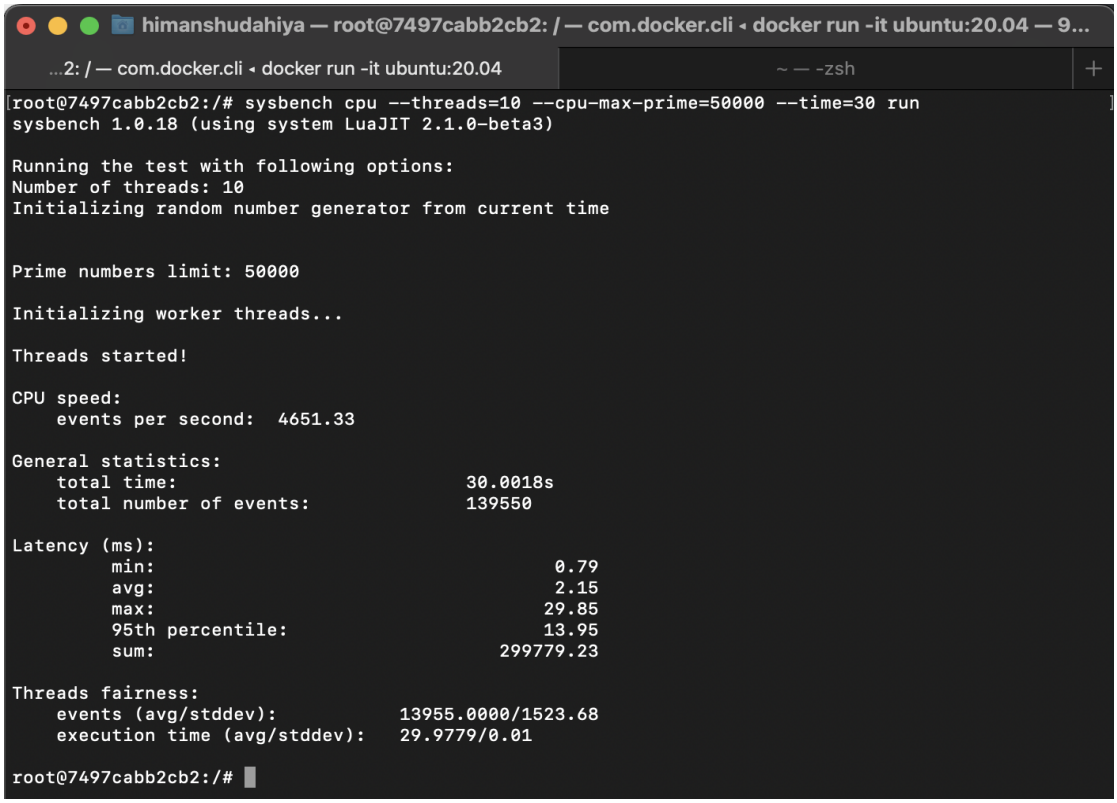

QEMU - (Press ctrl + alt + g to release Mouse)
Threads fairness:
events (avg/stddev): 36857.0000/0.00
execution time (avg/stddev): 29.9876/0.00
hdahiya@hd:~$ sysbench cpu --threads=1 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 1230.08
General statistics:
total time: 30.0006s
total number of events: 36904
Latency (ms):
min: 0.78
avg: 0.81
max: 1.52
95th percentile: 0.86
sum: 29988.39
Threads fairness:
events (avg/stddev): 36904.0000/0.00
execution time (avg/stddev): 29.9884/0.00
hdahiya@hd:~$

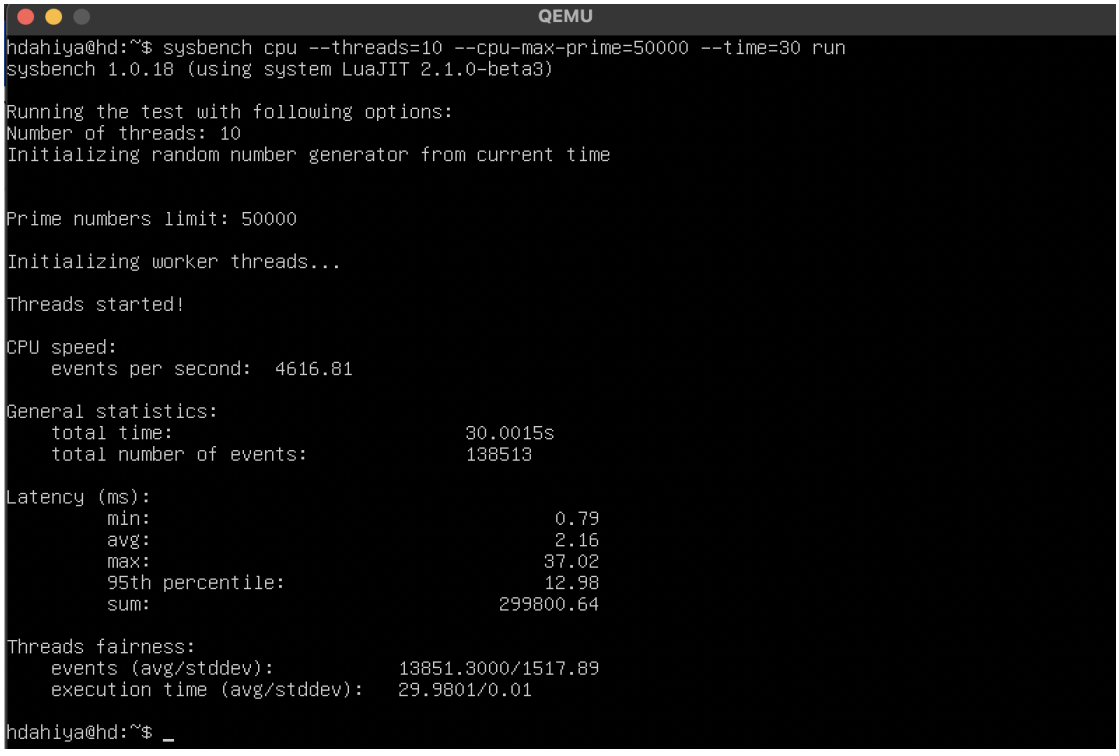
Run 5


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 100...
...b2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
[root@7497cabb2cb2:/# sysbench cpu --threads=1 --cpu-max-prime=50000 --time=30 run
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: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 1220.48
General statistics:
total time: 30.0007s
total number of events: 36617
Latency (ms):
min: 0.78
avg: 0.82
max: 2.10
95th percentile: 0.87
sum: 29987.57
Threads fairness:
events (avg/stddev): 36617.0000/0.00
execution time (avg/stddev): 29.9876/0.00
root@7497cabb2cb2:/#

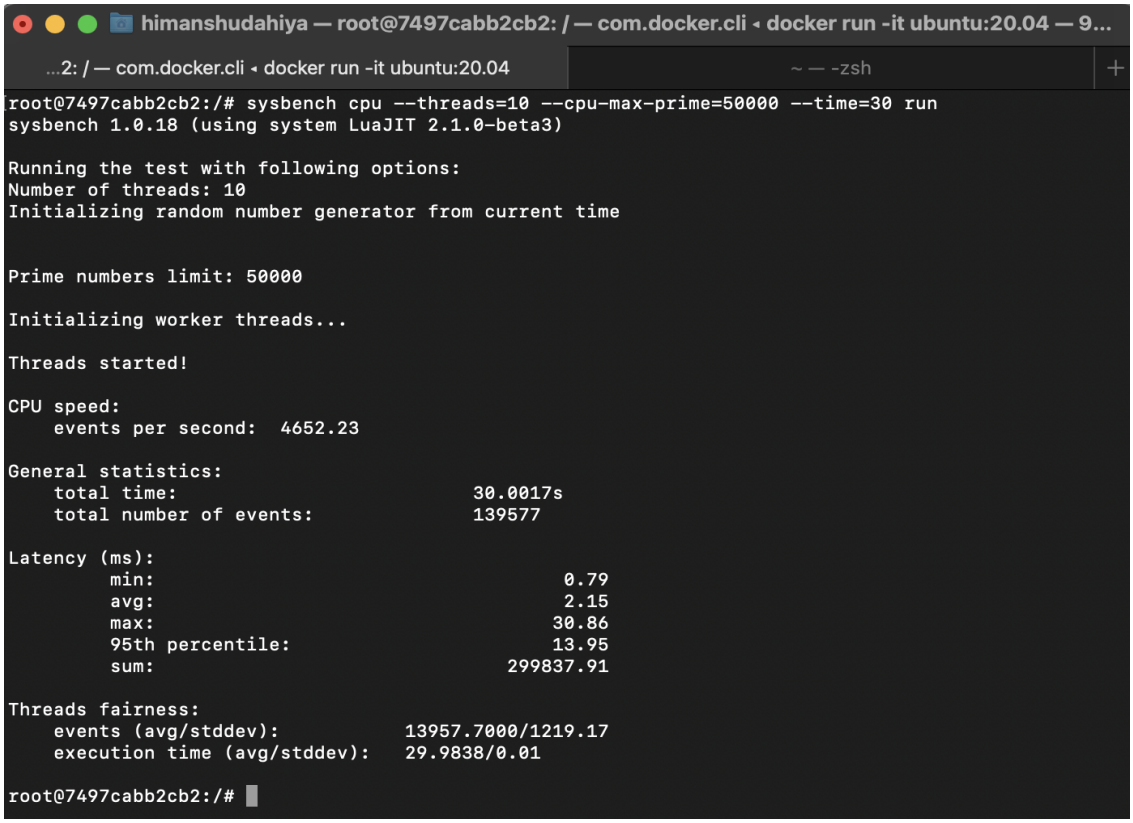

QEMU - (Press ctrl + alt + g to release Mouse)
Threads fairness:
events (avg/stddev): 36904.0000/0.00
execution time (avg/stddev): 29.9884/0.00
hdahiya@hd:~$ sysbench cpu --threads=1 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 1229.86
General statistics:
total time: 30.0003s
total number of events: 36897
Latency (ms):
min: 0.78
avg: 0.81
max: 1.33
95th percentile: 0.86
sum: 29987.55
Threads fairness:
events (avg/stddev): 36897.0000/0.00
execution time (avg/stddev): 29.9875/0.00
hdahiya@hd:~$

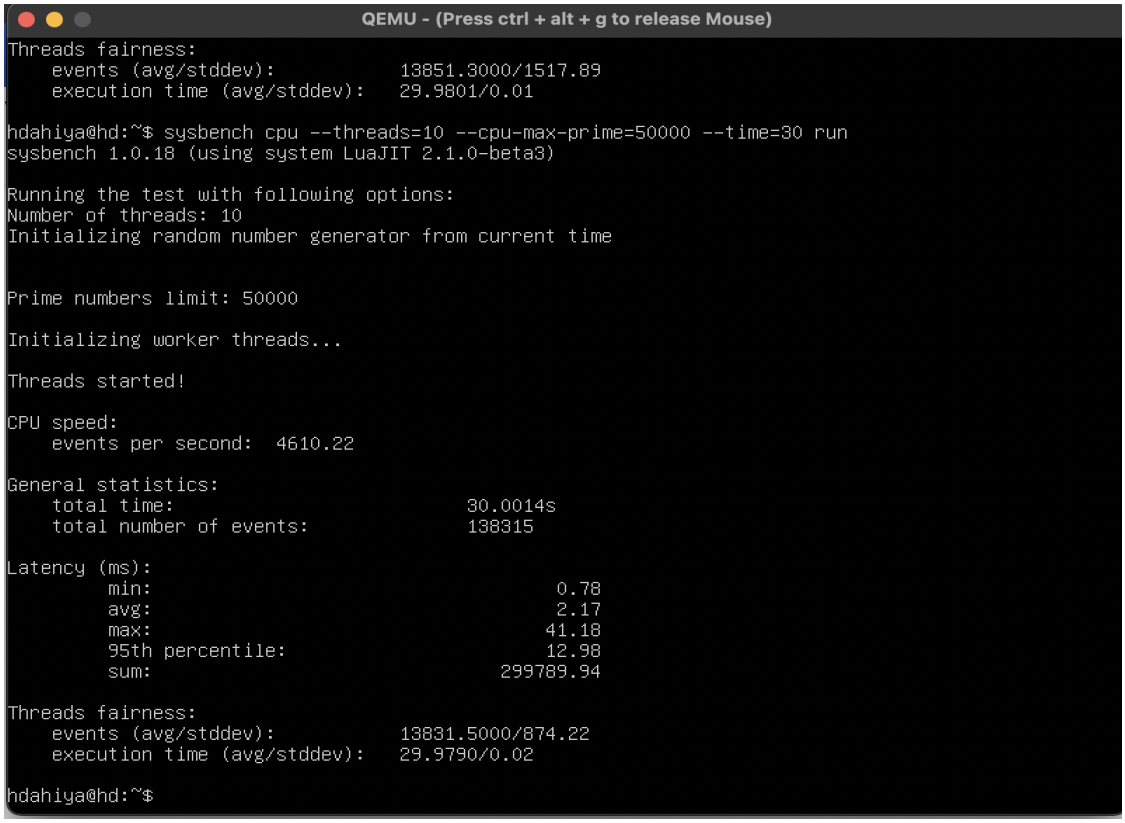
Testcase 4
sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run
Run 1



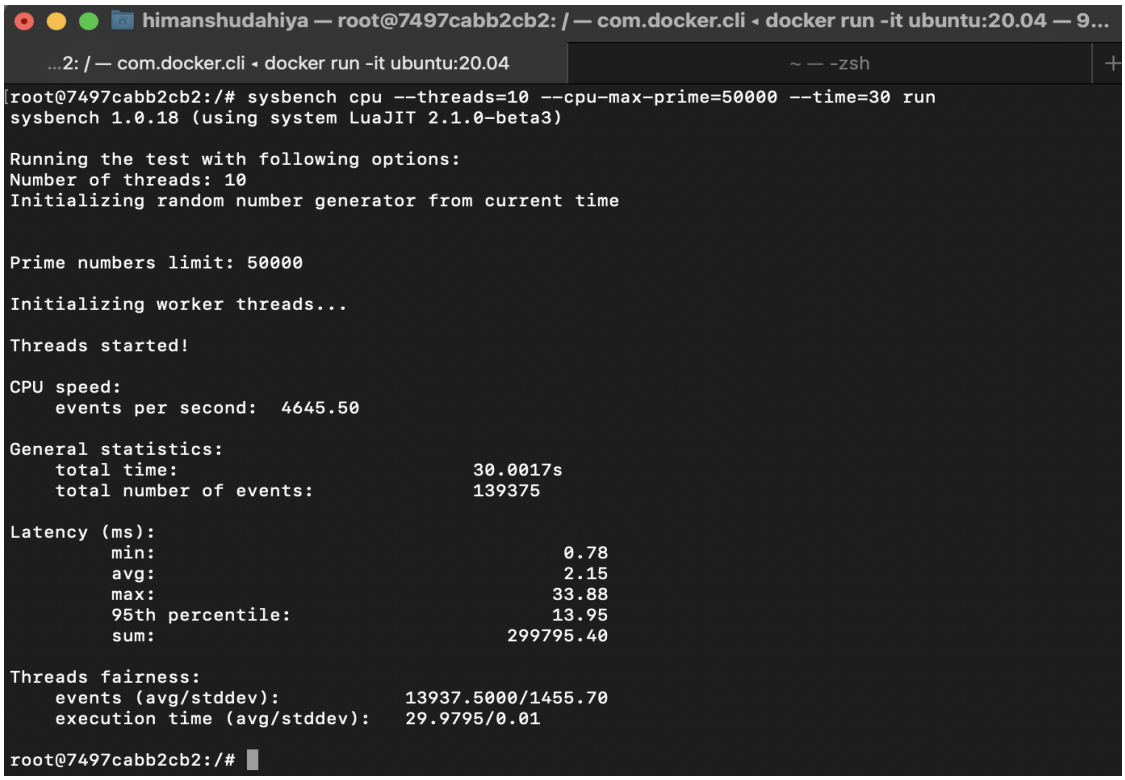


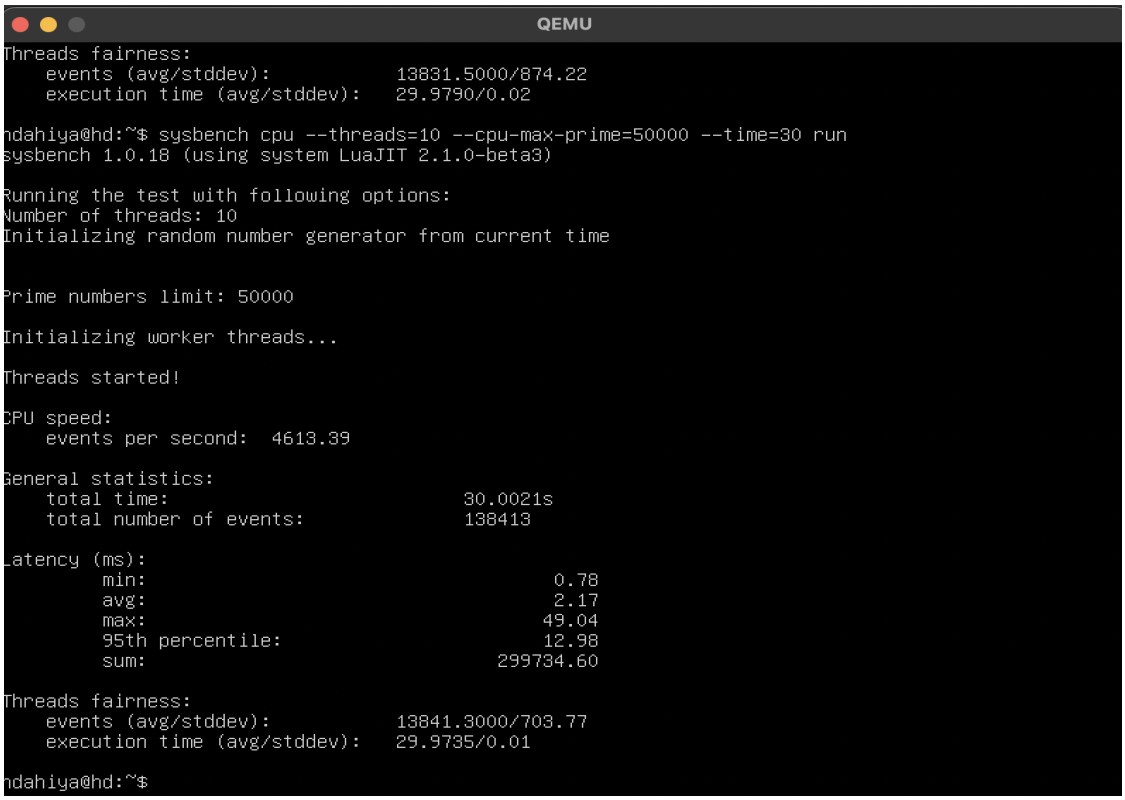
Run 2


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 9...
...2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 10
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 4652.23
General statistics:
total time: 30.0017s
total number of events: 139577
Latency (ms):
min: 0.79
avg: 2.15
max: 30.86
95th percentile: 13.95
sum: 299837.91
Threads fairness:
events (avg/stddev): 13957.7000/1219.17
execution time (avg/stddev): 29.9838/0.01
root@7497cabb2cb2:/#

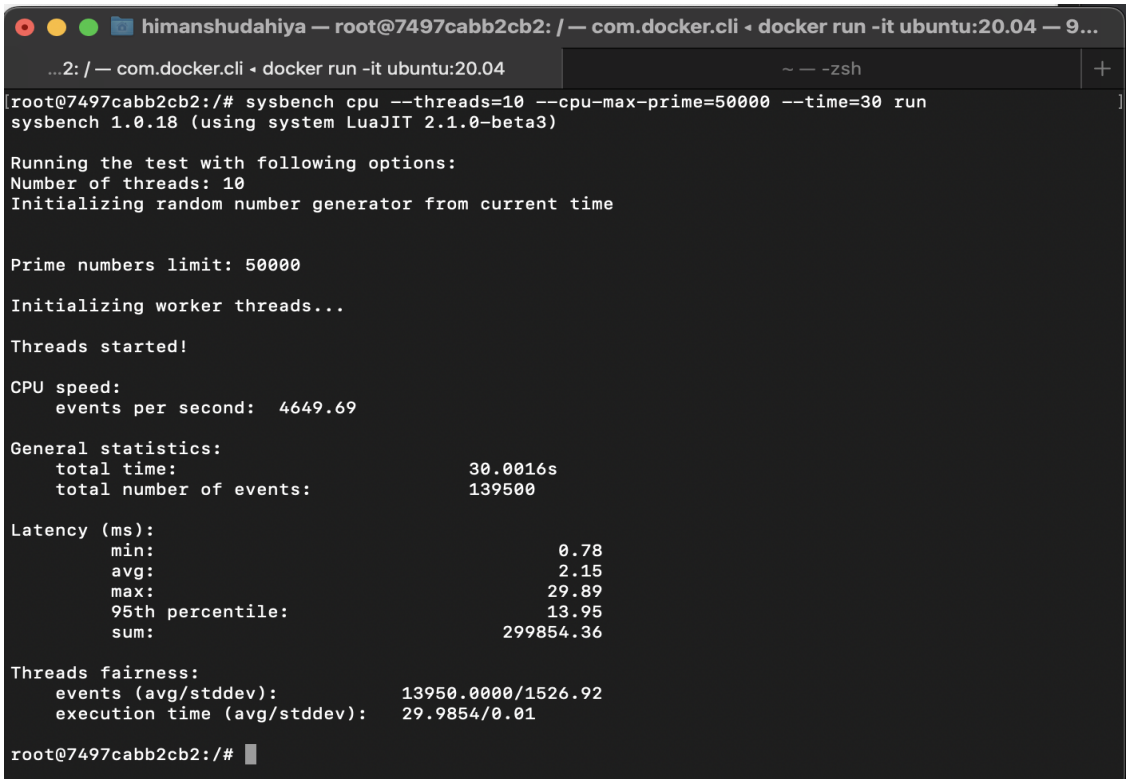

QEMU - (Press ctrl + alt + g to release Mouse)
Threads fairness:
events (avg/stddev): 13851.3000/1517.89
execution time (avg/stddev): 29.9801/0.01
hdahiya@hd:~$ sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 10
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 4610.22
General statistics:
total time: 30.0014s
total number of events: 138315
Latency (ms):
min: 0.78
avg: 2.17
max: 41.18
95th percentile: 12.98
sum: 299789.94
Threads fairness:
events (avg/stddev): 13831.5000/874.22
execution time (avg/stddev): 29.9790/0.02
hdahiya@hd:~$

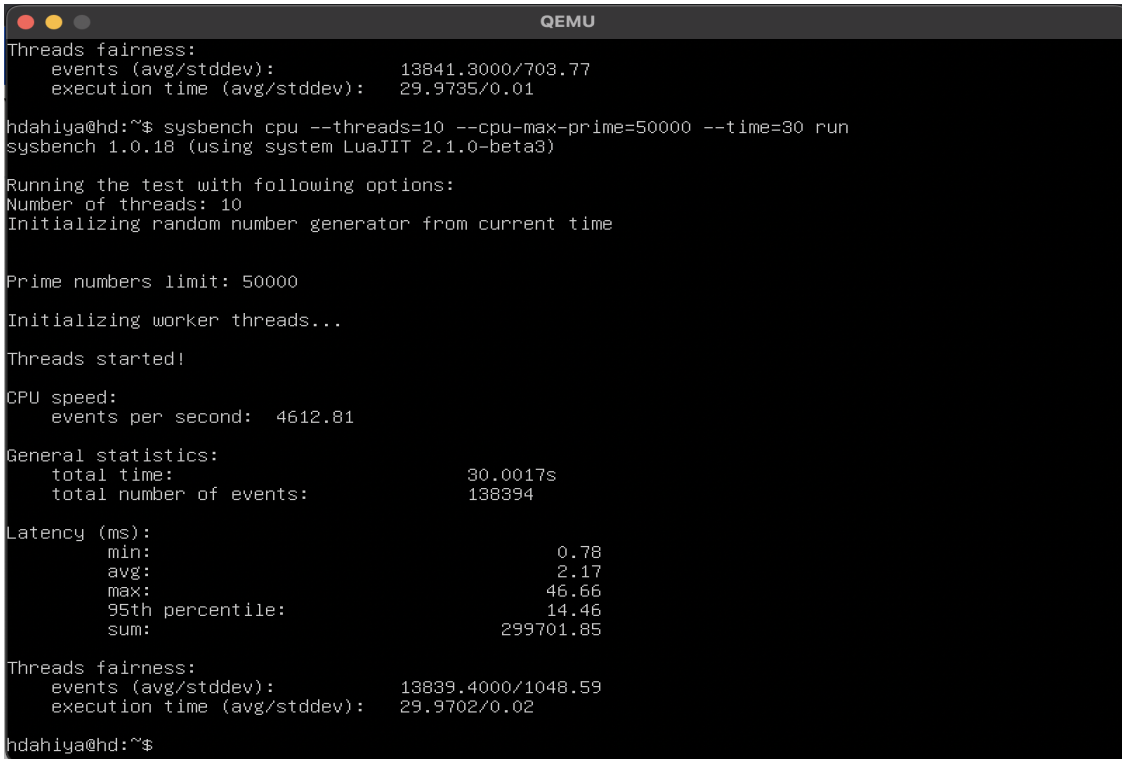
Run 3


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 9...
...2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
root@7497cabb2cb2:/# sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 10
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 4645.50
General statistics:
total time: 30.0017s
total number of events: 139375
Latency (ms):
min: 0.78
avg: 2.15
max: 33.88
95th percentile: 13.95
sum: 299795.40
Threads fairness:
events (avg/stddev): 13937.5000/1455.70
execution time (avg/stddev): 29.9795/0.01
root@7497cabb2cb2:/#

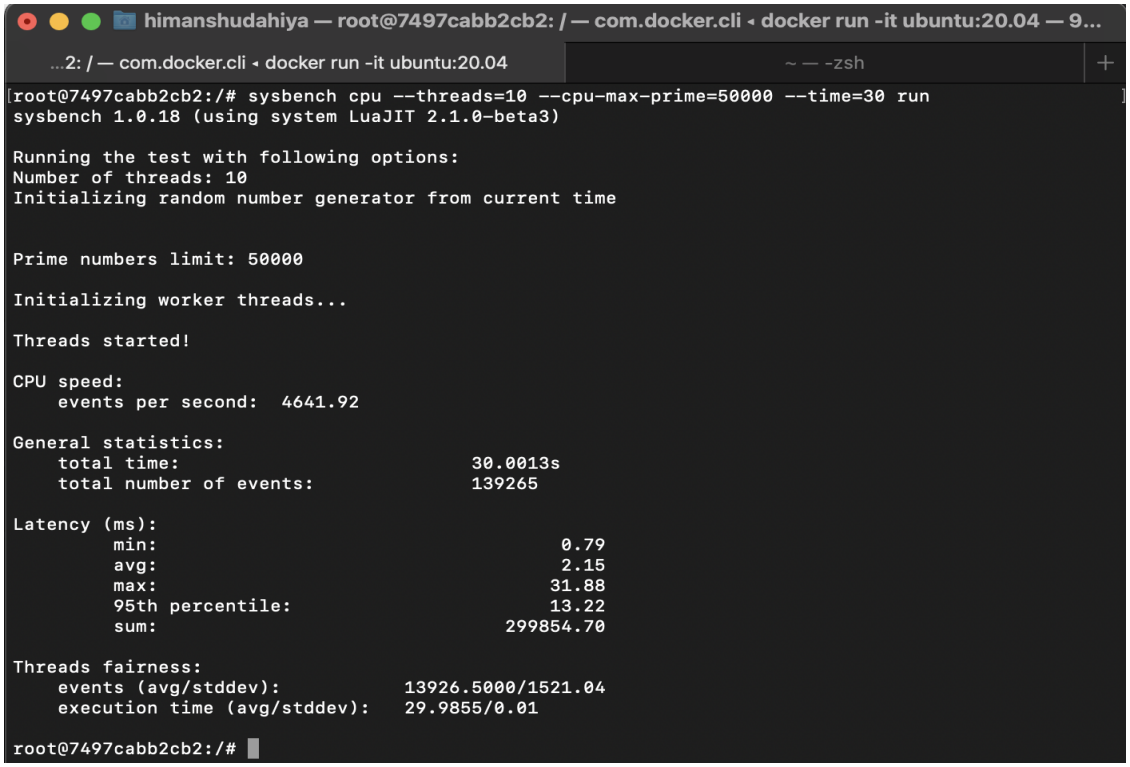

QEMU
Threads fairness:
events (avg/stddev): 13831.5000/874.22
execution time (avg/stddev): 29.9790/0.02
ndahiya@hd:~$ sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 10
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 4613.39
General statistics:
total time: 30.0021s
total number of events: 138413
Latency (ms):
min: 0.78
avg: 2.17
max: 49.04
95th percentile: 12.98
sum: 299734.60
Threads fairness:
events (avg/stddev): 13841.3000/703.77
execution time (avg/stddev): 29.9735/0.01
ndahiya@hd:~$

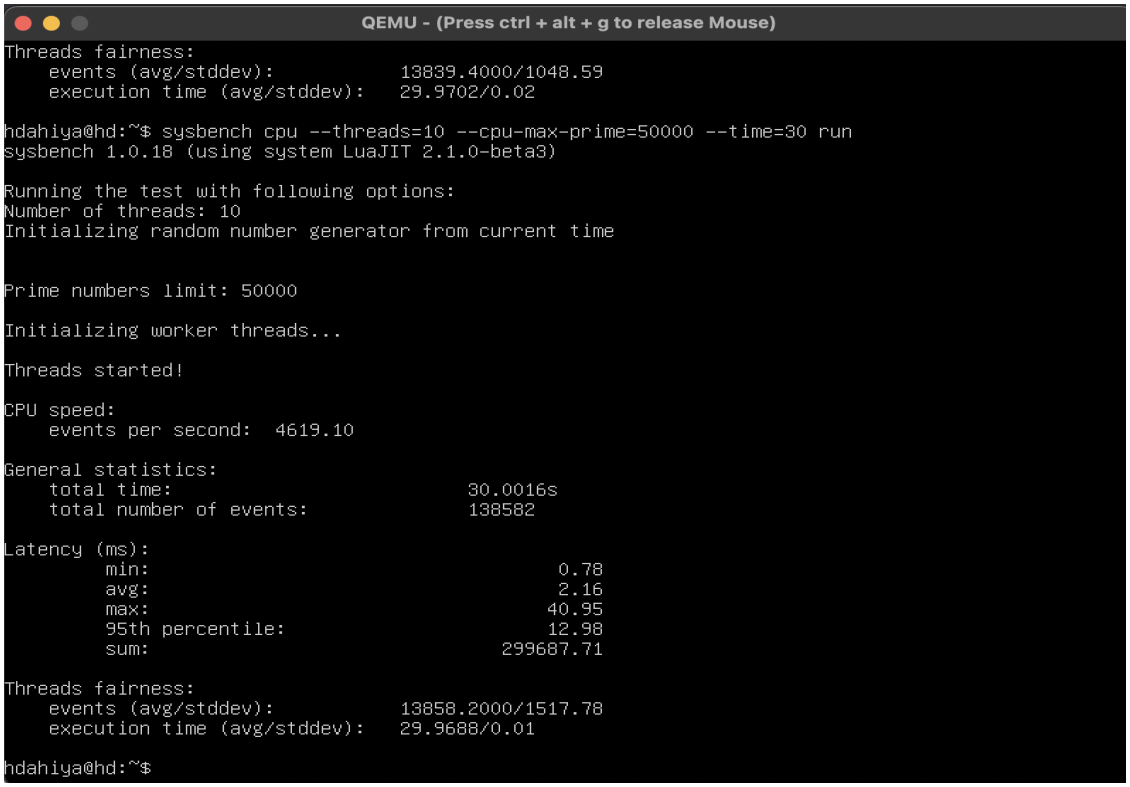
Run 4


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 9...
...2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
[root@7497cabb2cb2:/# sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 10
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 4649.69
General statistics:
total time: 30.0016s
total number of events: 139500
Latency (ms):
min: 0.78
avg: 2.15
max: 29.89
95th percentile: 13.95
sum: 299854.36
Threads fairness:
events (avg/stddev): 13950.0000/1526.92
execution time (avg/stddev): 29.9854/0.01
root@7497cabb2cb2:/#


QEMU
Threads fairness:
events (avg/stddev): 13841.3000/703.77
execution time (avg/stddev): 29.9735/0.01
hdahiya@hd:~$ sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 10
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 4612.81
General statistics:
total time: 30.0017s
total number of events: 138394
Latency (ms):
min: 0.78
avg: 2.17
max: 46.66
95th percentile: 14.46
sum: 299701.85
Threads fairness:
events (avg/stddev): 13839.4000/1048.59
execution time (avg/stddev): 29.9702/0.02
hdahiya@hd:~$

Run 5


himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 9...
...2: / — com.docker.cli ◂ docker run -it ubuntu:20.04
~ — -zsh
[root@7497cabb2cb2:/# sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 10
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 4641.92
General statistics:
total time: 30.0013s
total number of events: 139265
Latency (ms):
min: 0.79
avg: 2.15
max: 31.88
95th percentile: 13.22
sum: 299854.70
Threads fairness:
events (avg/stddev): 13926.5000/1521.04
execution time (avg/stddev): 29.9855/0.01
root@7497cabb2cb2:/#


QEMU - (Press ctrl + alt + g to release Mouse)
Threads fairness:
events (avg/stddev): 13839.4000/1048.59
execution time (avg/stddev): 29.9702/0.02
hdahiya@hd:~$ sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 10
Initializing random number generator from current time
Prime numbers limit: 50000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 4619.10
General statistics:
total time: 30.0016s
total number of events: 138582
Latency (ms):
min: 0.78
avg: 2.16
max: 40.95
95th percentile: 12.98
sum: 299687.71
Threads fairness:
events (avg/stddev): 13858.2000/1517.78
execution time (avg/stddev): 29.9688/0.01
hdahiya@hd:~$

Experiment Analysis

QEMU vs Docker

Test 1

sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run

	min	Average	Max	No. of events
Docker	0.09	0.09	0.42	109963
QEMU	0.09	0.09	0.59	110763

Test 2

sysbench cpu –threads=1 --cpu-max-prime=50000 --time=30 run

	min	Average	Max	No. of events
Docker	0.78	0.81	2.54	37022
QEMU	0.78	0.81	1.59	36365

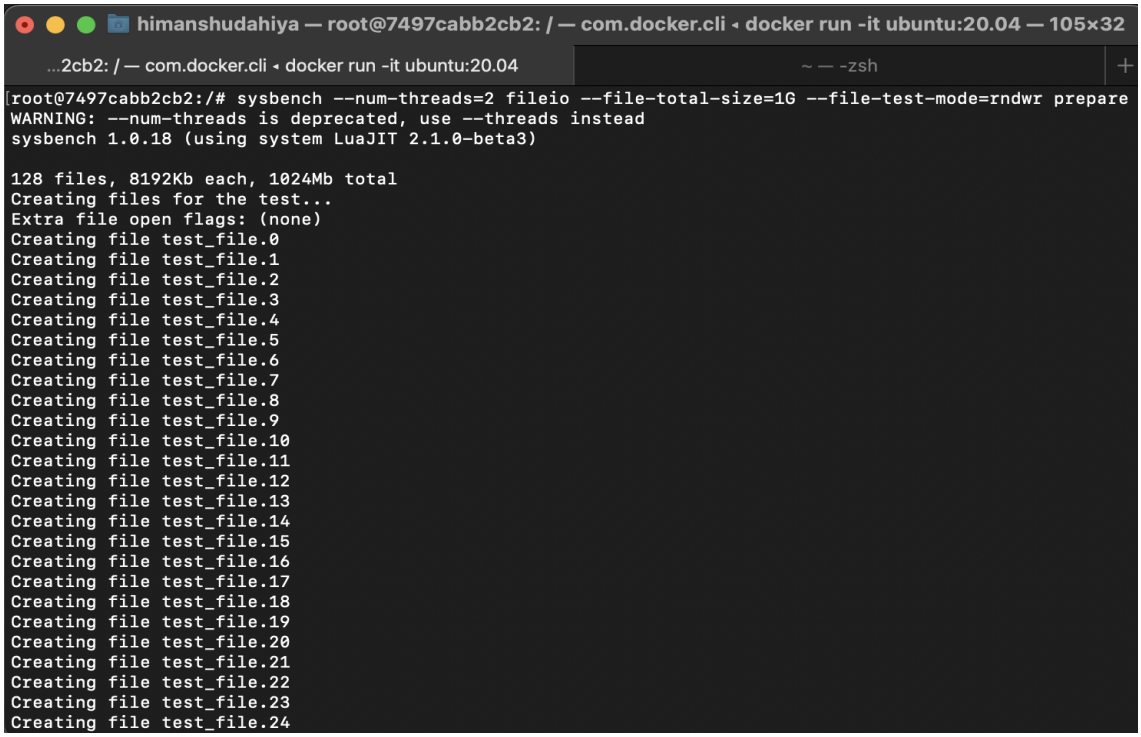
Test 3

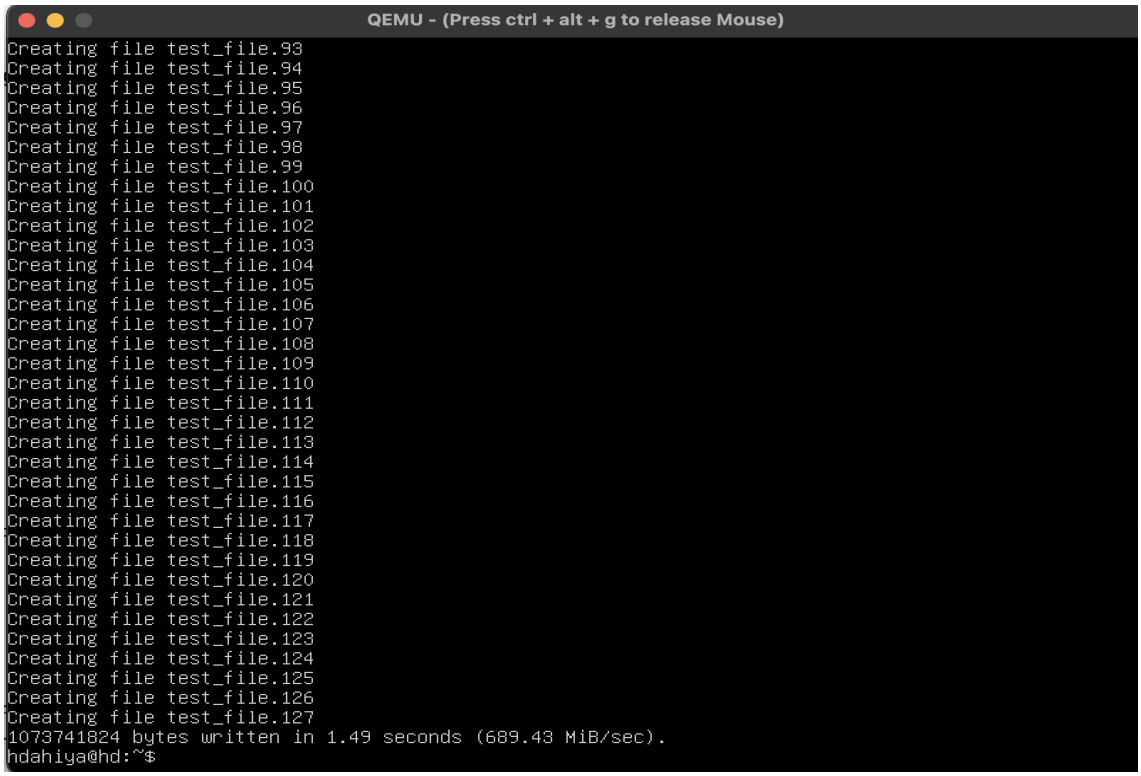
sysbench cpu --threads=10 --cpu-max-prime=50000 --time=30 run

	min	Average	Max	No. of events
Docker	0.79	2.15	29.85	13955
QEMU	0.79	2.16	37.02	13851

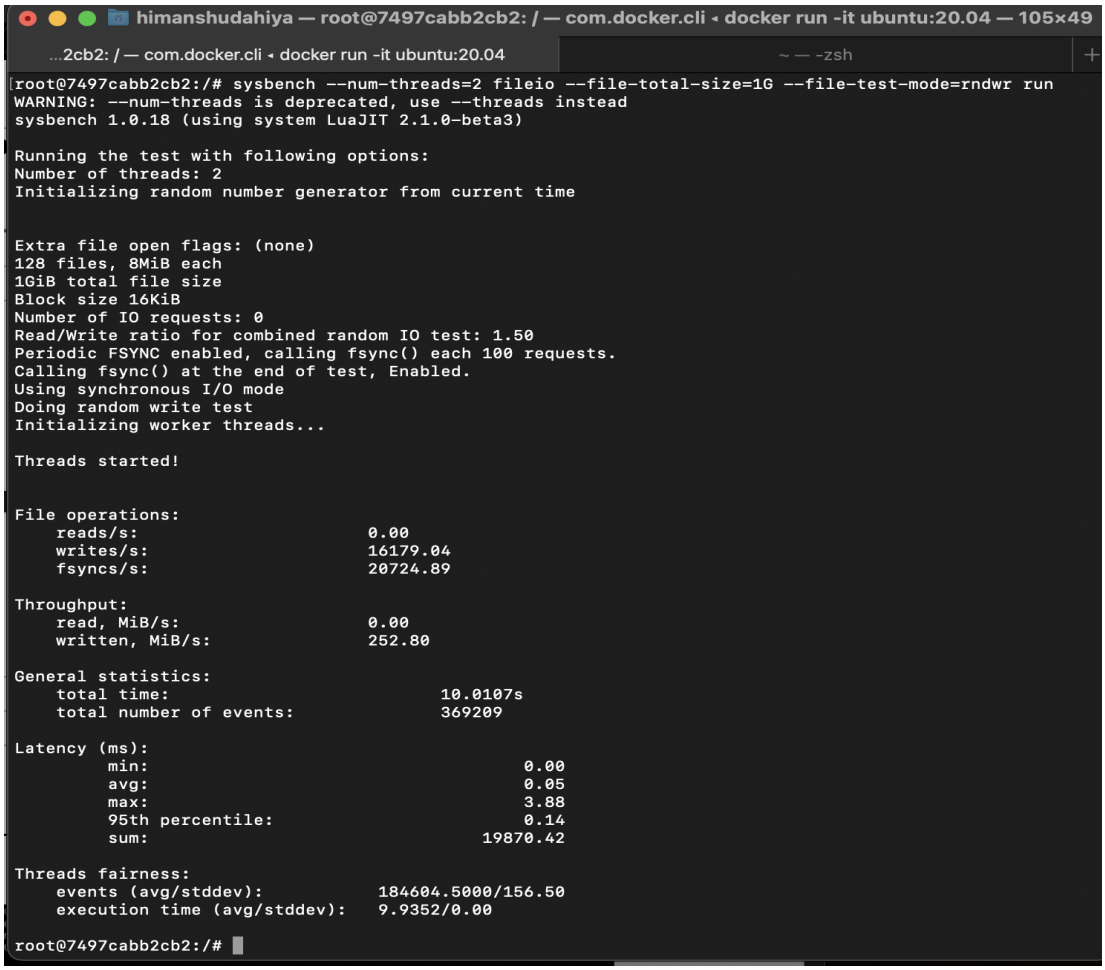
In the CPU test cases, we can conclude that they have very close performance but docker outperforms QEMU slightly.

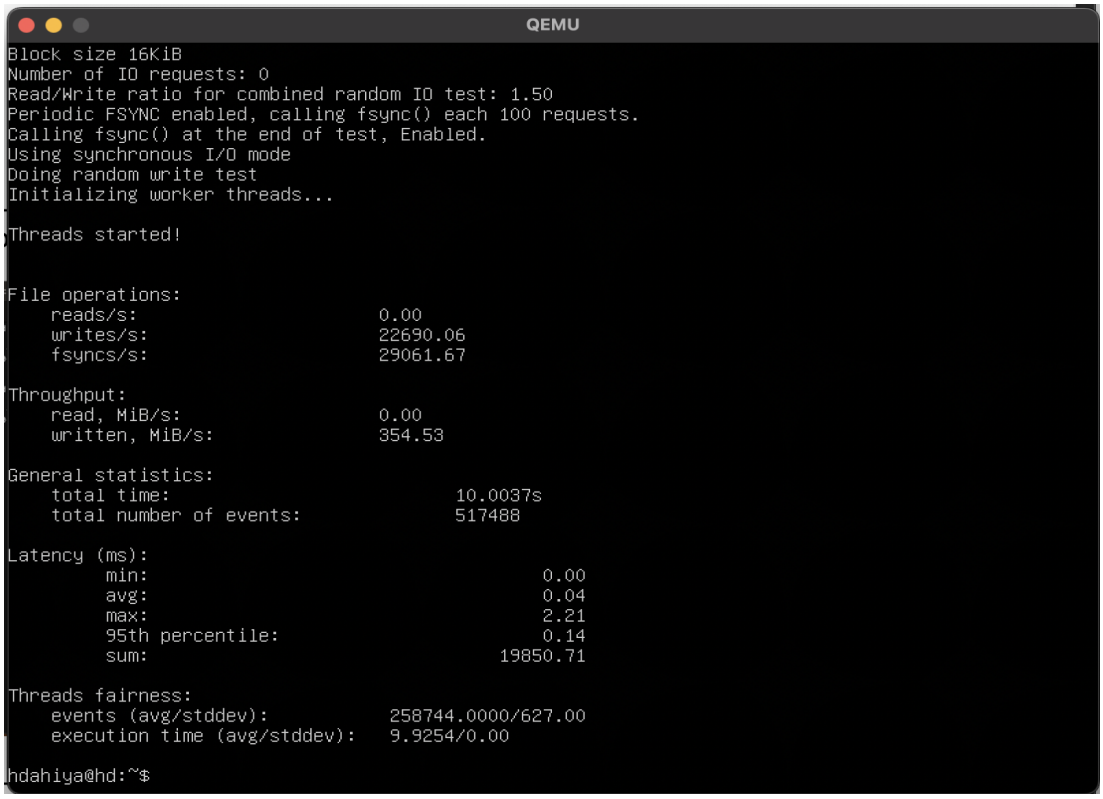
FileIO
Testcase 1
sysbench --num-threads=2 fileio --file-total-size=1G --file-test-mode=rndwr prepare





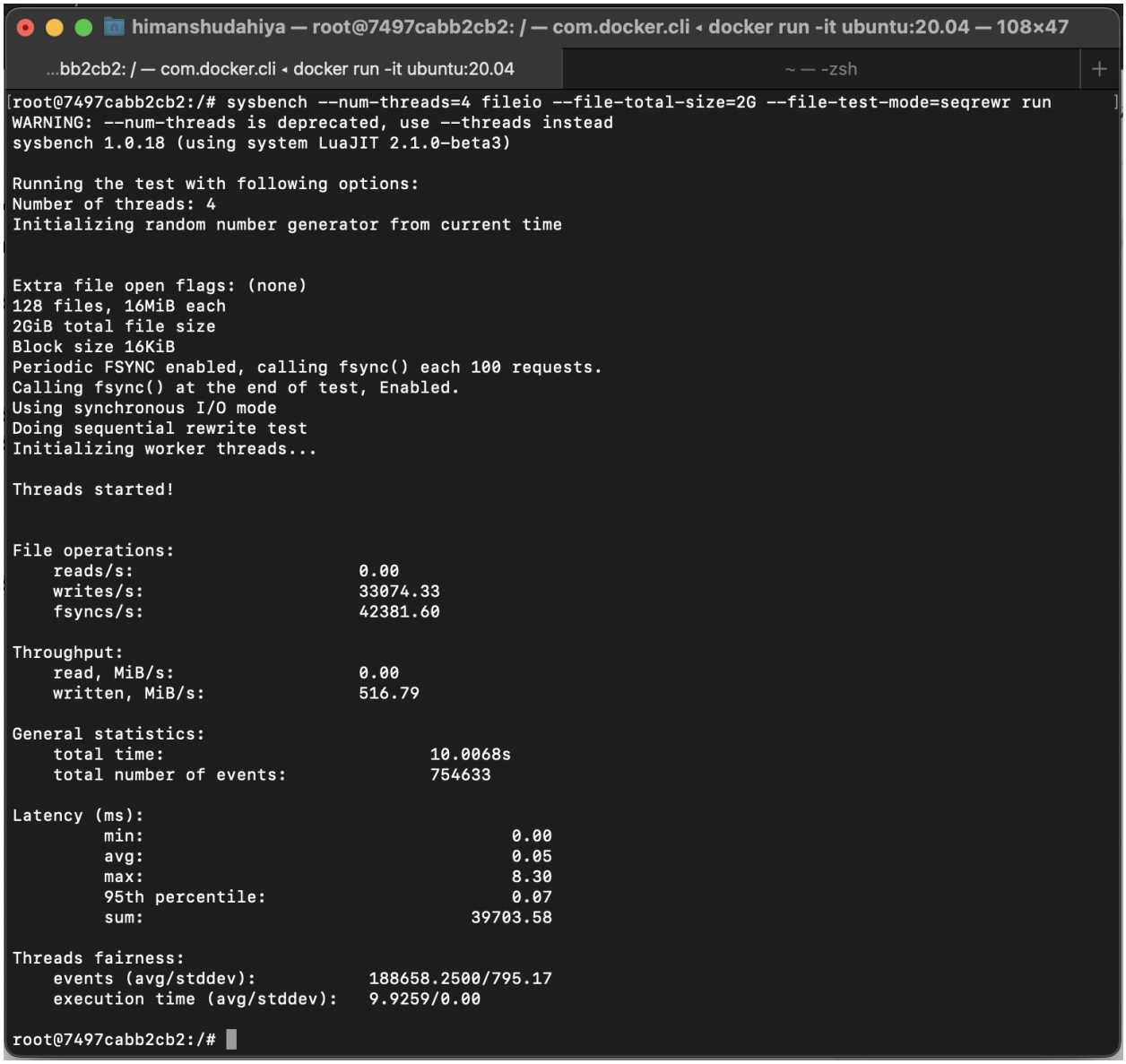
sysbench --num-threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run

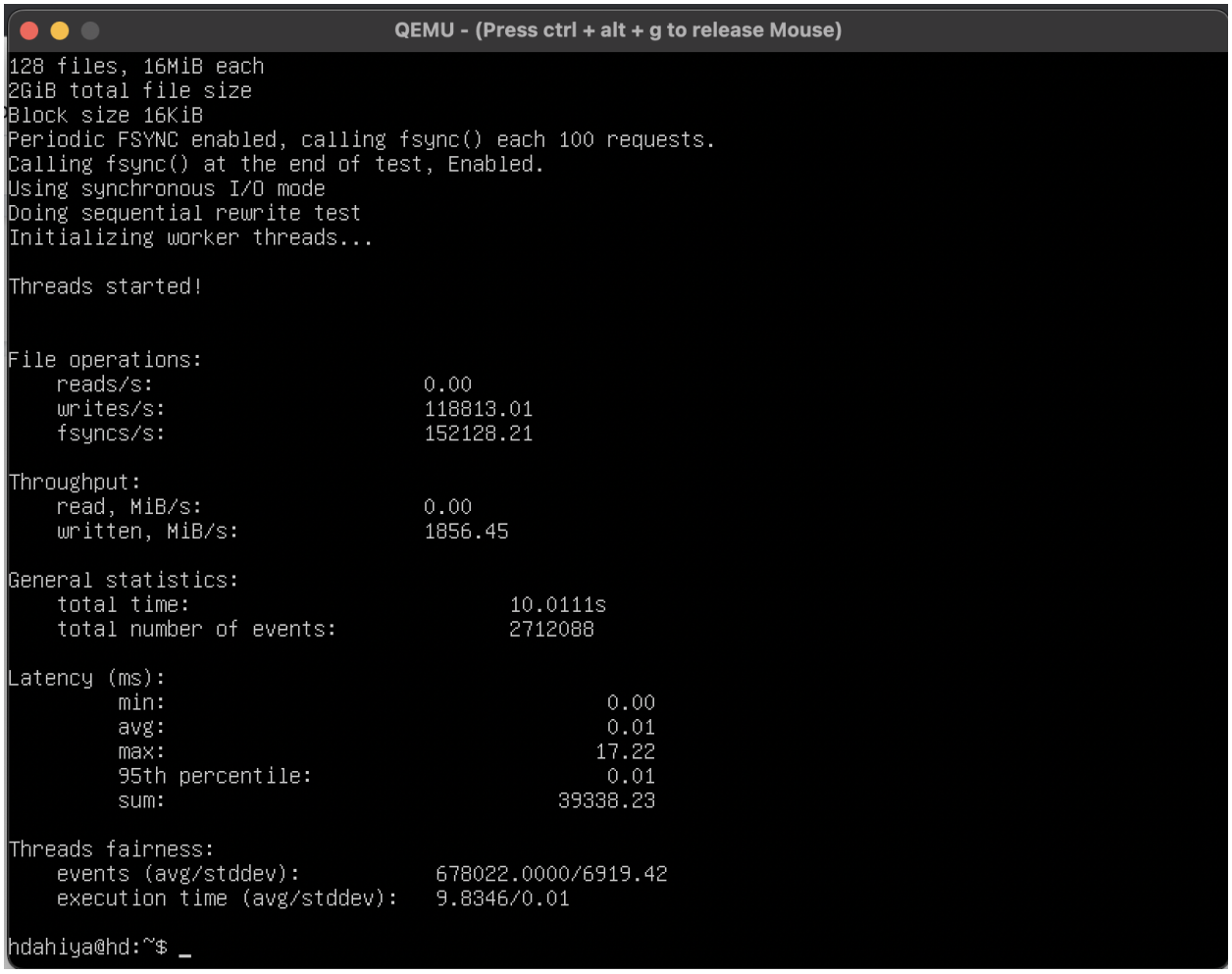




sysbench --num-threads=2 fileio --file-total-size=1G --file-test-mode=rndwr cleanup

Testcase 2
sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=seqrewr prepare
sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=seqrewr run

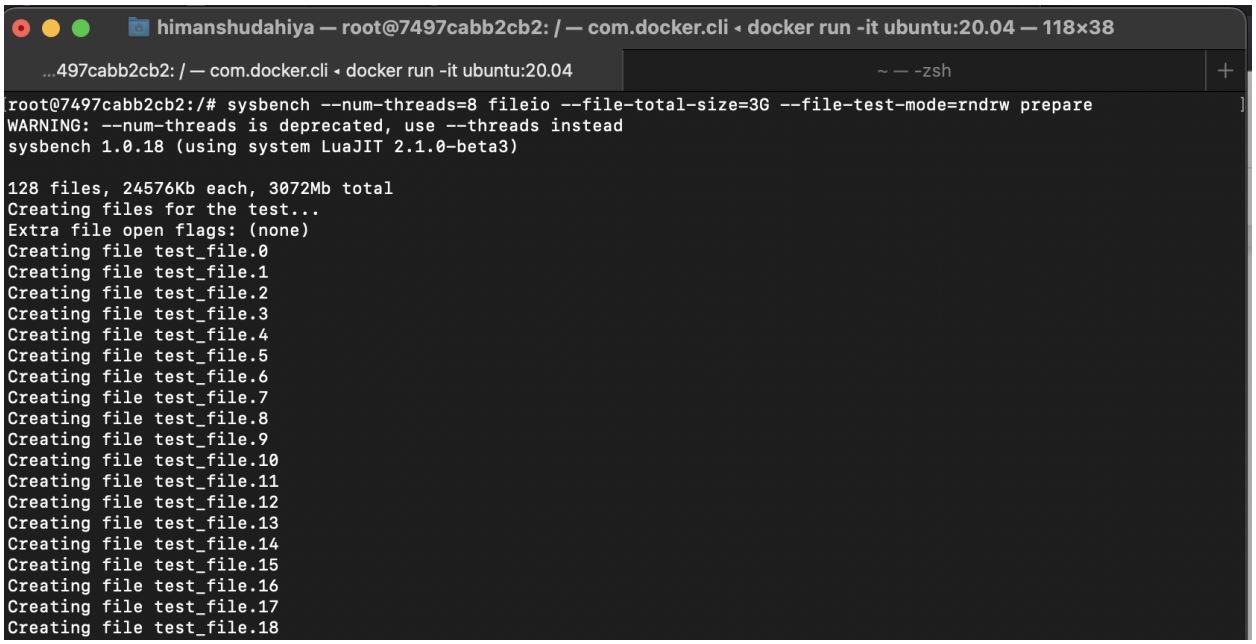


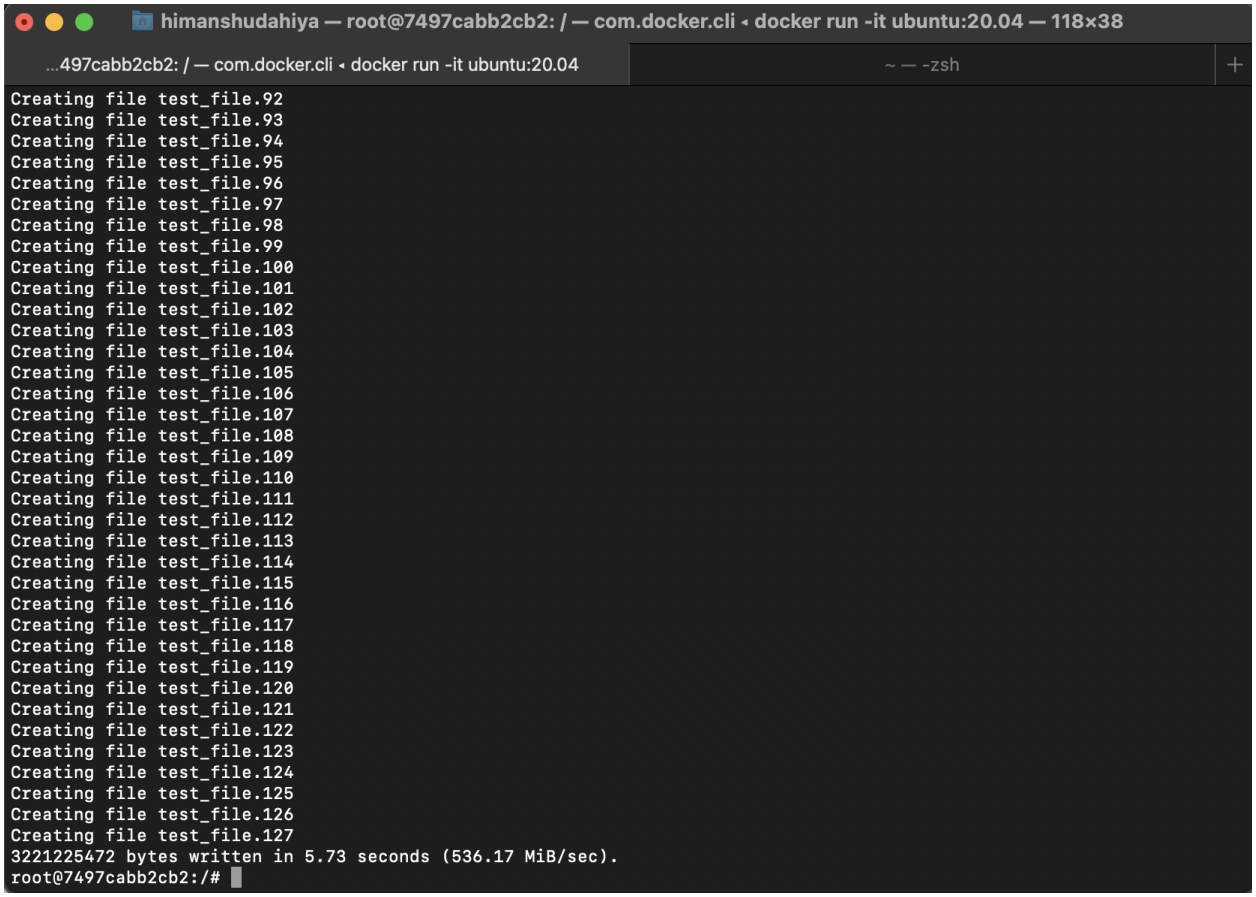


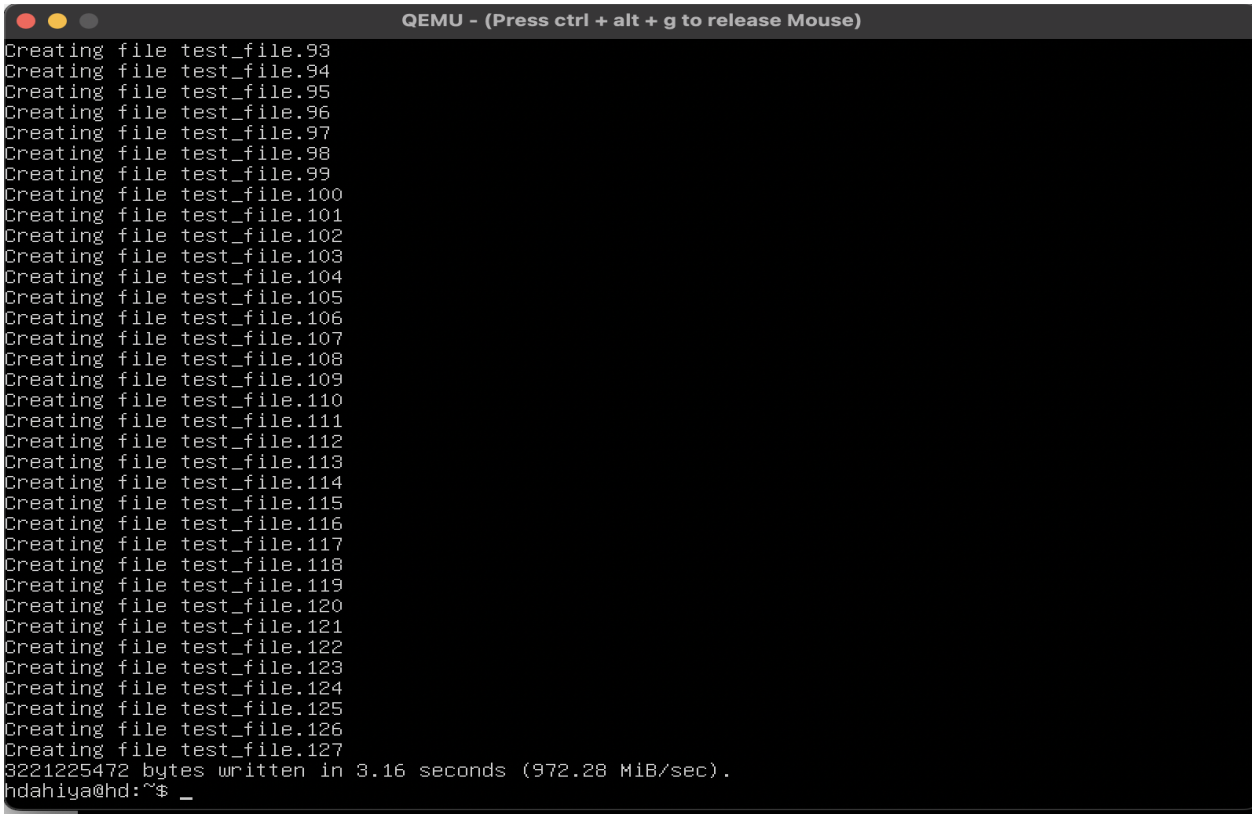
sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=seqrewr cleanup

Testcase 3

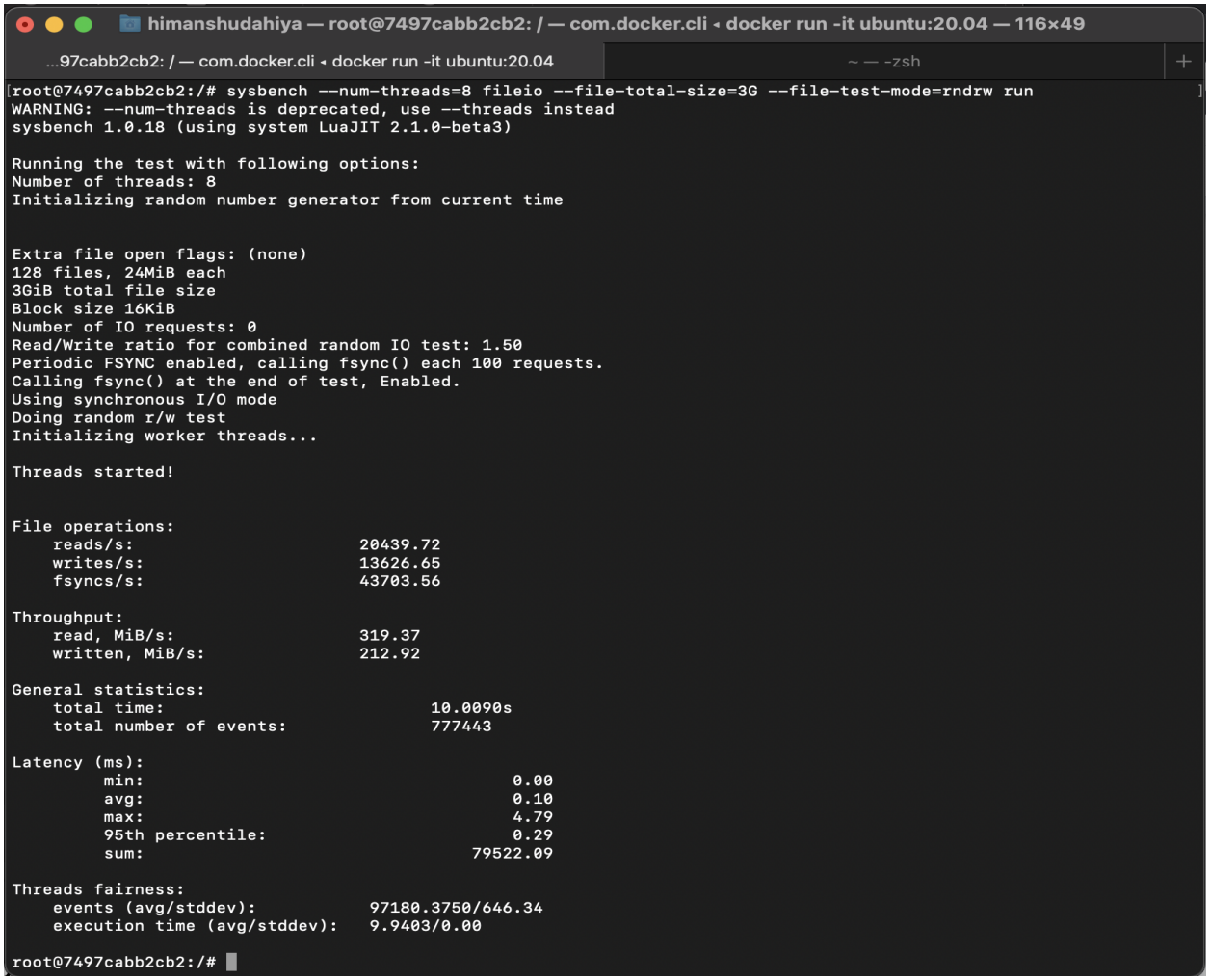
sysbench --num-threads=8 fileio --file-total-size=3G --file-test-mode=rndrw prepare

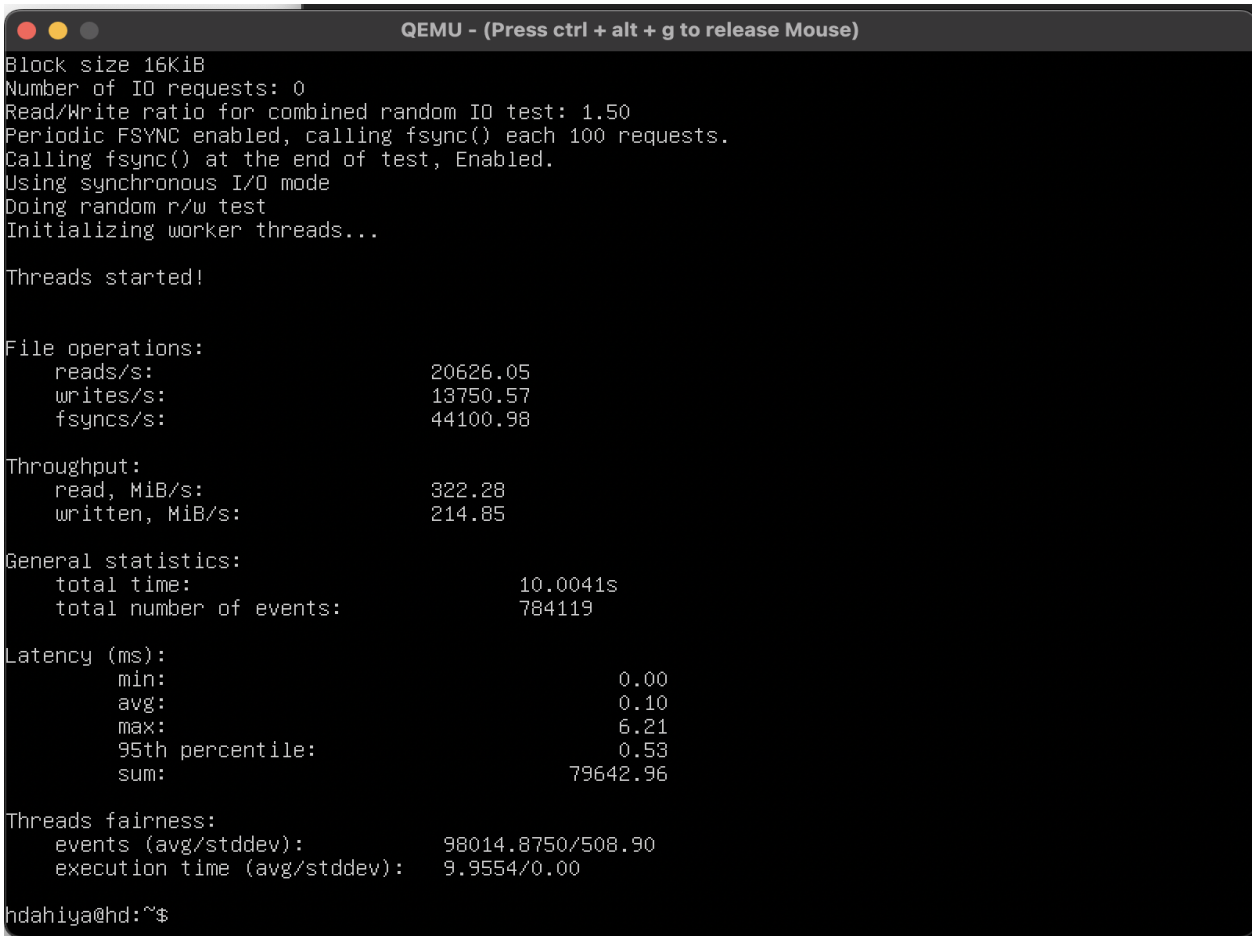




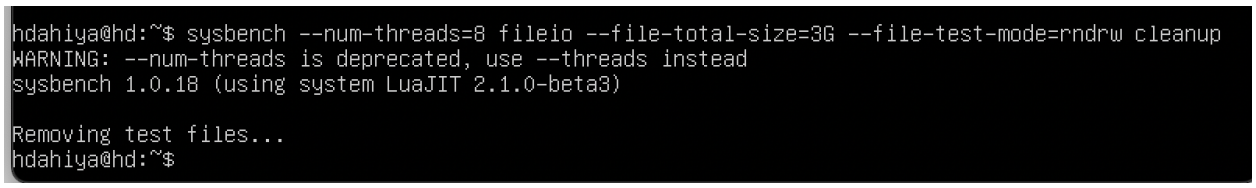

QEMU - (Press ctrl + alt + g to release Mouse)
Creating file test_file.93
Creating file test_file.94
Creating file test_file.95
Creating file test_file.96
Creating file test_file.97
Creating file test_file.98
Creating file test_file.99
Creating file test_file.100
Creating file test_file.101
Creating file test_file.102
Creating file test_file.103
Creating file test_file.104
Creating file test_file.105
Creating file test_file.106
Creating file test_file.107
Creating file test_file.108
Creating file test_file.109
Creating file test_file.110
Creating file test_file.111
Creating file test_file.112
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
3221225472 bytes written in 3.16 seconds (972.28 MiB/sec).
hdahiya@hd:~$ _

sysbench --num-threads=8 fileio --file-total-size=3G --file-test-mode=rndrw run

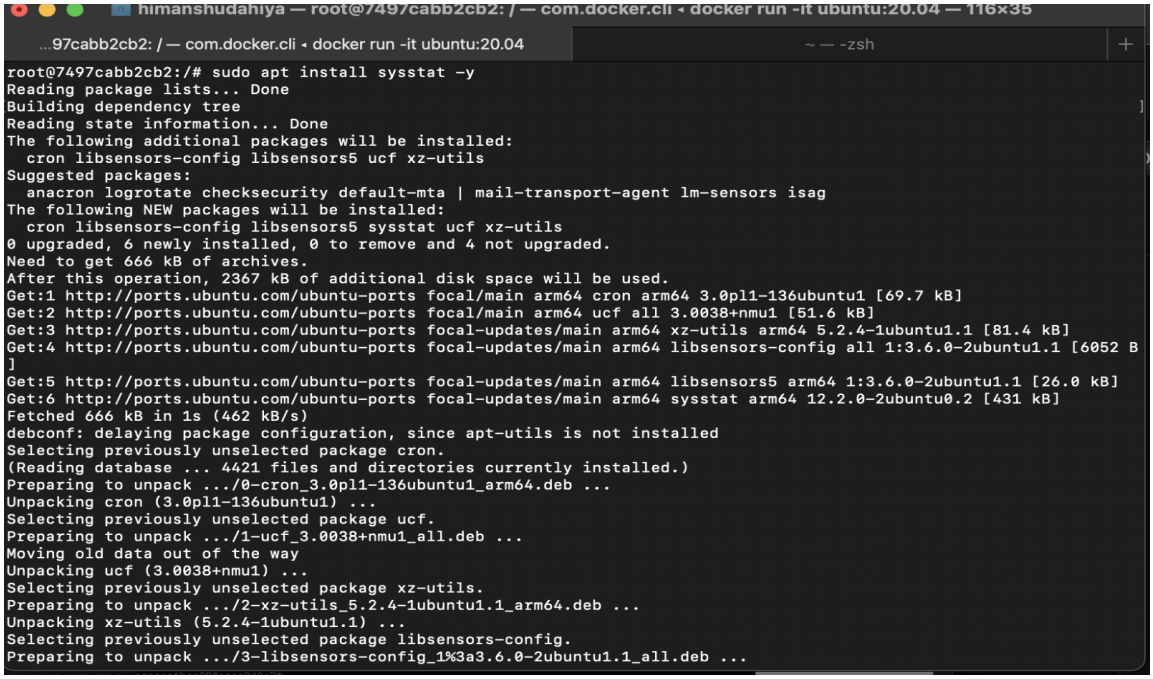


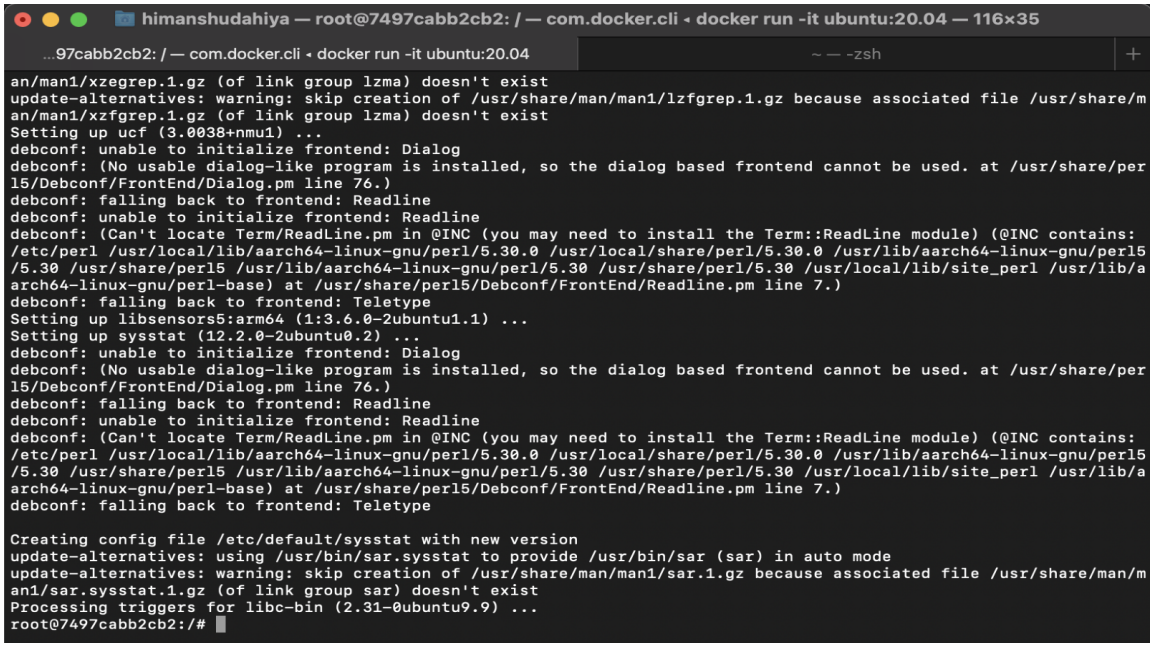


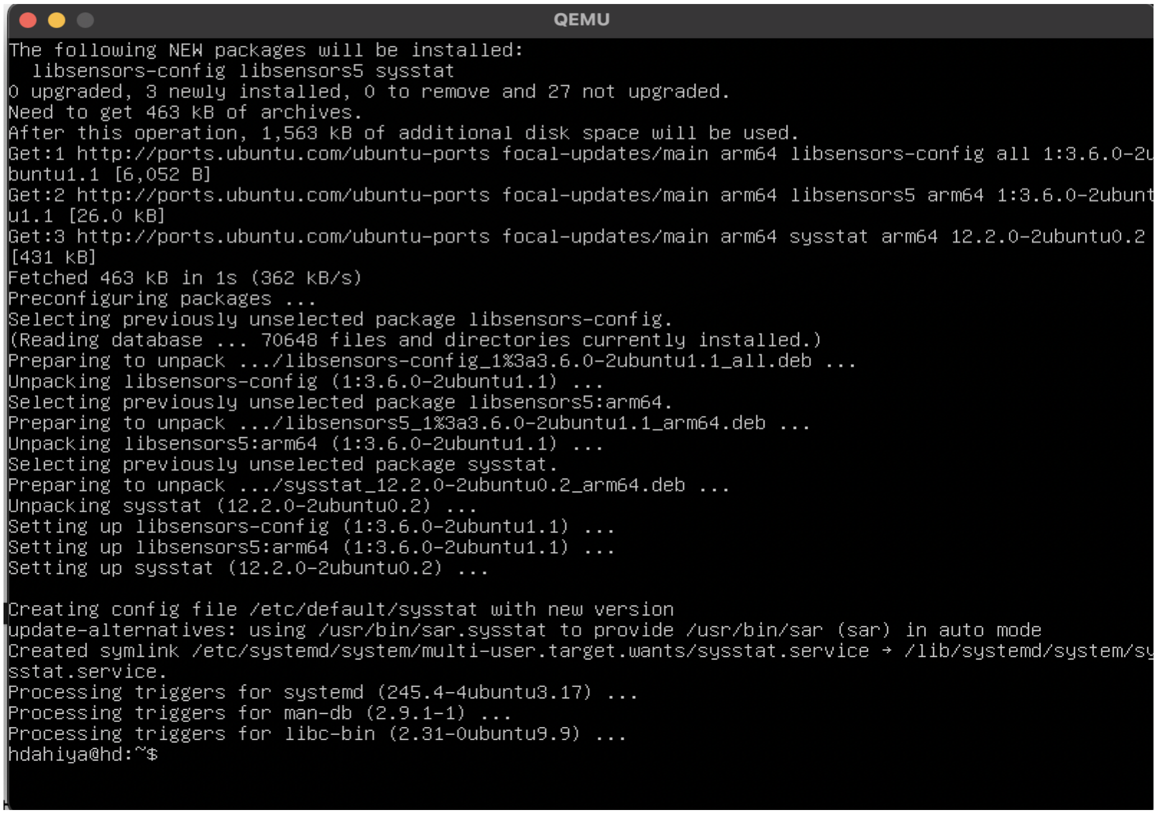
sysbench --num-threads=8 fileio --file-total-size=3G --file-test-mode=rndrw cleanup



Installing sysstat

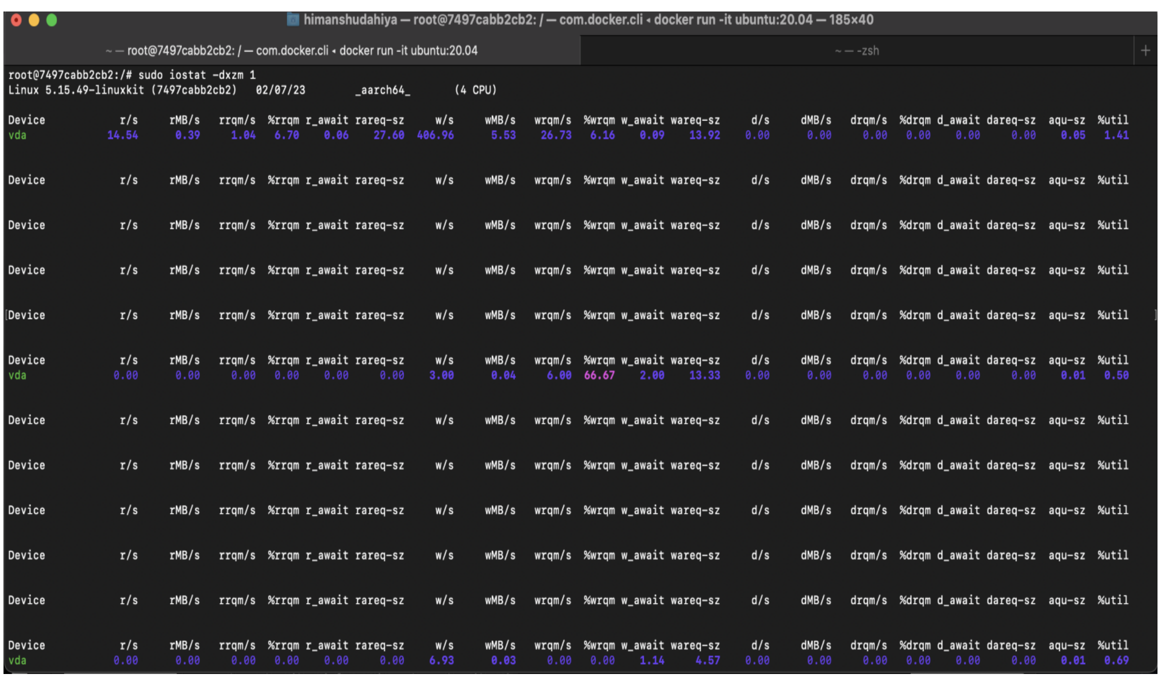




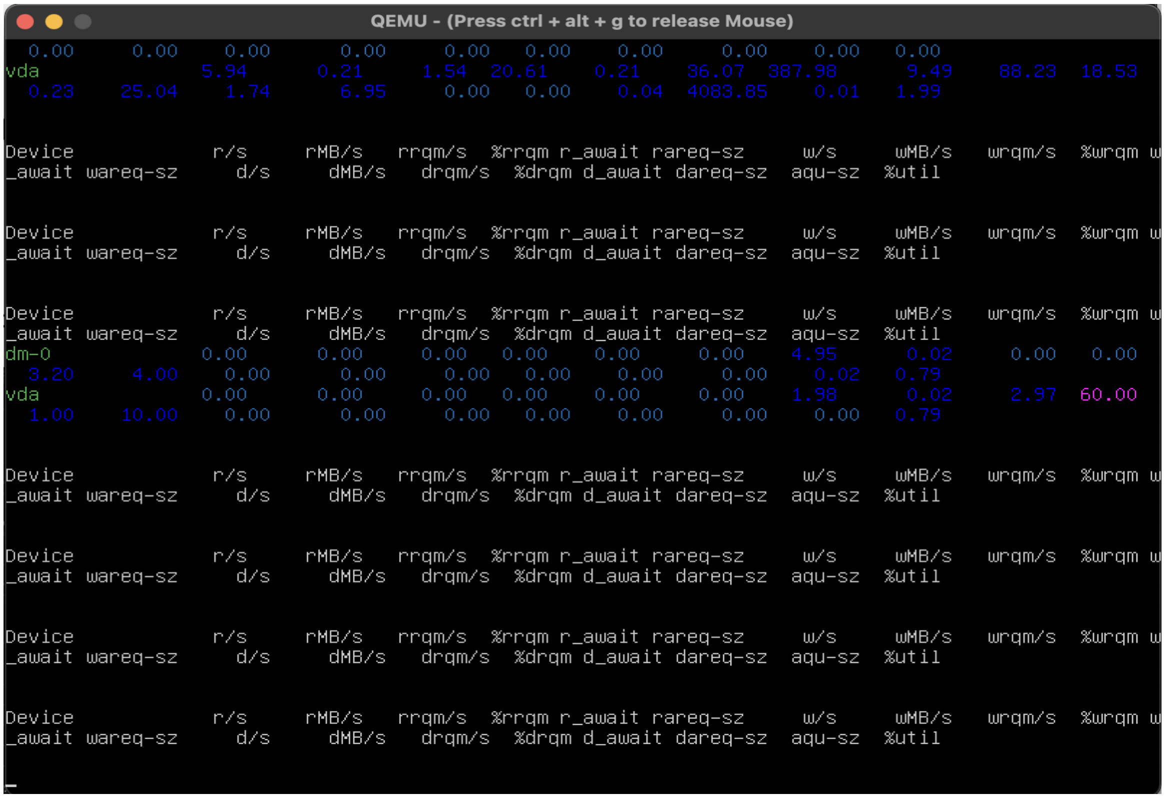


```
QEMU
The following NEW packages will be installed:
  libsensors-config libsensors5 sysstat
0 upgraded, 3 newly installed, 0 to remove and 27 not upgraded.
Need to get 463 kB of archives.
After this operation, 1,563 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libsensors-config all 1:3.6.0-2u
buntu1.1 [6,052 B]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libsensors5 arm64 1:3.6.0-2ubunt
u1.1 [26.0 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 sysstat arm64 12.2.0-2ubuntu0.2
[431 kB]
Fetched 463 kB in 1s (362 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libsensors-config.
(Reading database ... 70648 files and directories currently installed.)
Preparing to unpack .../libsensors-config_1%3a3.6.0-2ubuntu1.1_all.deb ...
Unpacking libsensors-config (1:3.6.0-2ubuntu1.1) ...
Selecting previously unselected package libsensors5:arm64.
Preparing to unpack .../libsensors5_1%3a3.6.0-2ubuntu1.1_arm64.deb ...
Unpacking libsensors5:arm64 (1:3.6.0-2ubuntu1.1) ...
Selecting previously unselected package sysstat.
Preparing to unpack .../sysstat_12.2.0-2ubuntu0.2_arm64.deb ...
Unpacking sysstat (12.2.0-2ubuntu0.2) ...
Setting up libsensors-config (1:3.6.0-2ubuntu1.1) ...
Setting up libsensors5:arm64 (1:3.6.0-2ubuntu1.1) ...
Setting up sysstat (12.2.0-2ubuntu0.2) ...

Creating config file /etc/default/sysstat with new version
update-alternatives: using /usr/bin/sar.sysstat to provide /usr/bin/sar (sar) in auto mode
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service → /lib/systemd/system/sy
sstat.service.
Processing triggers for systemd (245.4-4ubuntu3.17) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
hdahiya@hd:~$
```

sudo iostat -dxzm 1

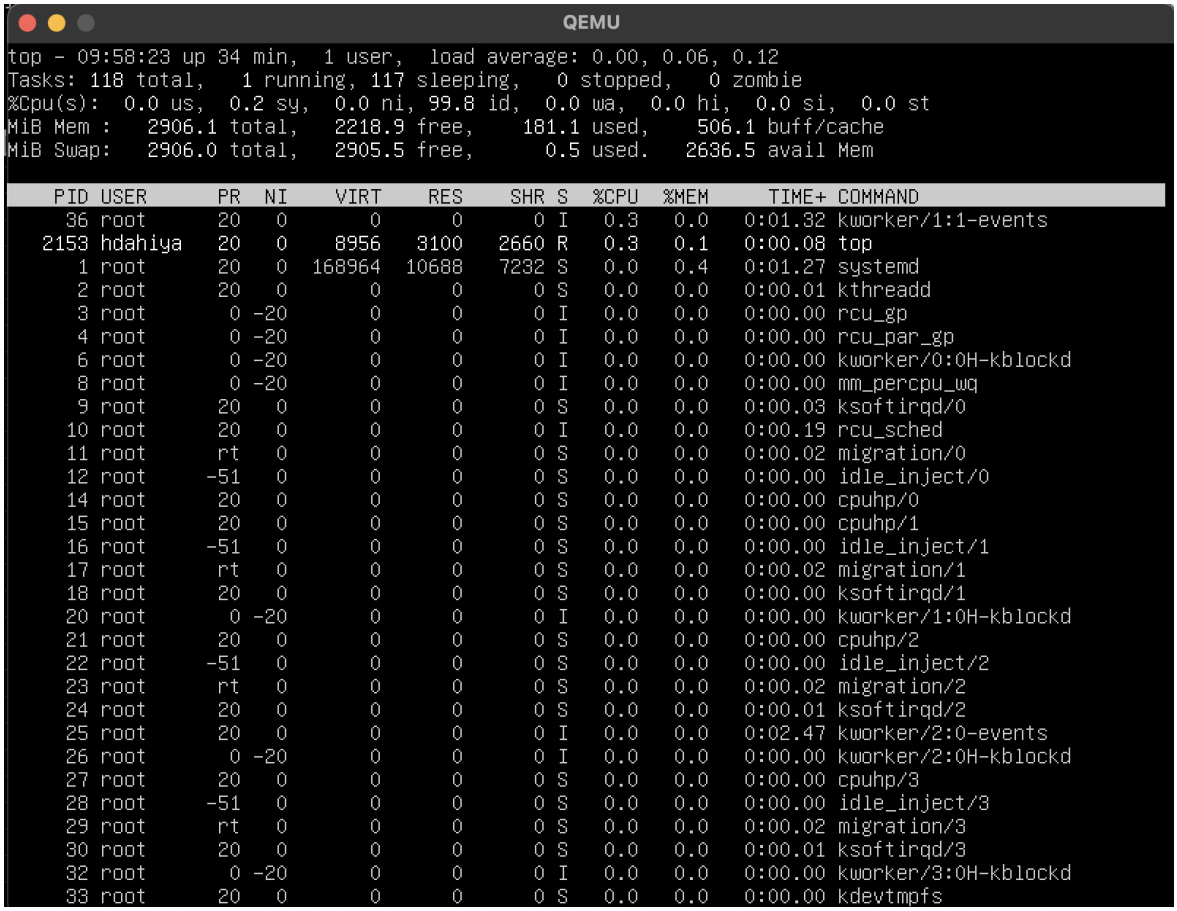


```
himanshudahiya — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04 — 185×40
~ — root@7497cabb2cb2: / — com.docker.cli ◂ docker run -it ubuntu:20.04        ~ — -zsh
root@7497cabb2cb2:/# sudo iostat -dxzm 1
Linux 5.15.49-linuxkit (7497cabb2cb2)   02/07/23        _aarch64_       (4 CPU)

Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util
vda             14.54      0.39     1.04   6.70    0.06    27.60  406.96      5.53    26.73   6.16    0.09    13.92    0.00      0.00     0.00   0.00    0.00     0.00    0.05   1.41


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util
vda              0.00      0.00     0.00   0.00    0.00     0.00    3.00      0.04     6.00  66.67    2.00    13.33    0.00      0.00     0.00   0.00    0.00     0.00    0.01   0.50


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util


Device            r/s     rMB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wMB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dMB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util
vda              0.00      0.00     0.00   0.00    0.00     0.00    6.93      0.03     0.00   0.00    1.14     4.57    0.00      0.00     0.00   0.00    0.00     0.00    0.01   0.69
```



The above screenshots show Disk IO utilization, latency throughput for QEMU and Docker.

Ubuntu processes



Experiment Analysis

QEMU vs Docker

Test 1

sysbench --num-threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run

	min	Average	Max	No. of events
Docker	0.00	0.05	3.88	184604
QEMU	0.00	0.04	2.21	258744

Test 2

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=seqrewr run

	min	Average	Max	No. of events
Docker	0.00	0.05	8.30	188658
QEMU	0.00	0.01	17.22	678022

Test 3

sysbench --num-threads=8 fileio --file-total-size=3G --file-test-mode=rndrw run

	min	Average	Max	No. of events
Docker	0.00	0.10	4.79	97180
QEMU	0.00	0.10	6.21	98014

In the CPU test cases, we can conclude that QEMU outperforms Docker.