

HW- 1 [Cloud Computation]

By: Aanchal Rai

Configurations of the System:

Host Computer Details:

- Model Identifier: MacBookPro17,1
- Operating System: macOS Big Sur Version 11.6
- Chip: Apple M1
- Total Number of Cores: 8 (4 performance and 4 efficiency)
- Memory: 8 GB

QEMU Configurations: (img below)

- OS: Ubuntu 20.04.3
- Memory Size: 1.93 GB
- # Of Cores: 1
- Machine Type: Qemu

```
System:   Kernel: 5.4.0-88-generic aarch64 bits: 64 compiler: gcc v: 9.3.0 Console: tty 1
          Distro: Ubuntu 20.04.3 LTS (Focal Fossa)
Machine:  Type: Qemu System: QEMU product: QEMU Virtual Machine v: virt-5.2 serial: <filter>
          MoBo: N/A model: N/A serial: N/A UEFI: EFI Development Kit II / OVMF v: 0.0.0
          date: 02/06/2015
CPU:      Topology: Single Core model: N/A bits: 64 type: UP arch: ARMv8
          features: Use -f option to see features bogomips: 0
          Speed: N/A min/max: N/A Core speed (MHz): No speed data found for 1 cores.
Graphics: Message: No ARM data found for this feature.
          Display: server: No display server data found. Headless machine? tty: 100x37
          Message: Advanced graphics data unavailable in console. Try -G --display
Audio:    Message: No ARM data found for this feature.
Network:  Message: No ARM data found for this feature.
          IF-ID-1: eth0 state: up speed: -1 duplex: unknown mac: <filter>
Drives:   Local Storage: total: 10.00 GiB used: 3.97 GiB (39.7%)
          ID-1: /dev/vda model: N/A size: 10.00 GiB
Partition: ID-1: / size: 8.30 GiB used: 3.86 GiB (46.5%) fs: ext4 dev: /dev/dm-0
          ID-2: /boot size: 975.9 MiB used: 107.4 MiB (11.0%) fs: ext4 dev: /dev/vda2
Sensors:  Message: No sensors data was found. Is sensors configured?
Info:     Processes: 88 Uptime: 1m Memory: 1.93 GiB used: 200.2 MiB (10.1%) Init: systemd
          runlevel: 5 Compilers: gcc: N/A Shell: bash v: 5.0.17 inxi: 3.0.38
```

Docker Configurations: (img below)

- OS: Ubuntu 20.0.4
- Memory Size: 1.94GB
- # Of Cores: 4

```
System:
  Kernel: 5.10.47-linuxkit aarch64 bits: 64 compiler: N/A Console: tty 0
  Distro: Ubuntu 20.04.3 LTS (Focal Fossa)
Machine:
  Message: No machine data: try newer kernel. Is dmidecode installed? Try -M --dmidecode.
CPU:
  Topology: Quad Core model: N/A variant: armv8 bits: 64 type: MCP
  arch: ARMv8
  features: Use -f option to see features bogomips: 0
  Speed: N/A min/max: N/A
  Core speeds (MHz): No speed data found for 4 cores.
Graphics:
  Message: No Device data found.
  Display: server: No display server data found. Headless machine?
  tty: 80x24
  Message: Advanced graphics data unavailable in console for root.
Audio:
  Message: No Device data found.
Network:
  Message: No ARM data found for this feature.
  IF-ID-1: eth0 state: up speed: 10000 Mbps duplex: full mac: <filter>
  IF-ID-2: ip6tnl0 state: down mac: <filter>
  IF-ID-3: tunl0 state: down mac: <filter>
Drives:
  Local Storage: total: 59.60 GiB used: 10.99 GiB (18.4%)
  ID-1: /dev/vda model: N/A size: 59.60 GiB
Partition:
  ID-1: / size: 58.42 GiB used: 5.49 GiB (9.4%) fs: overlay source: ERR-102
Sensors:
  Message: No sensors data was found. Is sensors configured?
Info:
  Processes: 3 Uptime: 4m Memory: 1.94 GiB used: 376.9 MiB (19.0%) Init: N/A
  Compilers: gcc: N/A Shell: bash v: 5.0.17 inxi: 3.0.38
root@1a63676aff72:/#
```

Steps to Enable QEMU:

1. Installed Ubuntu Guest VM from: https://cdimage.ubuntu.com/releases/20.04/release/ubuntu-20.04.3-live-server-arm64.iso?_ga=2.258759180.749508015.1632596737-506099903.1632596737
2. Cloned QEMU and checkout version 5.2.0 with below commands:
 - a. git clone https://github.com/qemu/qemu
 - b. cd qemu
 - c. git checkout v5.2.0
3. Applied patch series with below command:
`curl https://patchwork.kernel.org/series/418581/mbox/ | git am --exclude=MAINTAINERS`
4. Downloaded Xcode from App Store.

5. Applied patch to fix QEMU Build using below command:

```
git apply xcode-12-4.patch
```

6. Installed Homebrew using Rosetta 2

```
arch -x86_64 /bin/bash -c "$(curl -fsSL
```

```
https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

7. Added homebrew to PATH variable using below commands:

- o export PATH="/opt/homebrew/bin:/usr/local/bin:\$PATH"
- o alias ibrew='arch -x86_64 /usr/local/bin/brew'

8. Installed packages for QEMU Build using below command:

```
brew install libffi gettext pkg-config autoconf automake pixman
```

9. Ran below commands to build QEMU

- o mkdir build
- o cd build
- o ./configure --target-list=aarch64-softmmu --disable-gnutls
- o make -j8
- o sudo make install

10. Created the Ubuntu VM with below commands:

a. Created a hard disk with below command:

```
qemu-img create -f qcow2 disk.qcow2 10G
```

b. Created an empty file for persisting UEFI Variables:

```
dd if=/dev/zero conv=sync bs=1m count=64 of=ovmf_vars.fd
```

c. Ran QEMU with following command:

```
qemu-system-aarch64 \  
  -accel hvf \  
  -m 2048 \  
  -cpu cortex-a57 -M virt,highmem=off \  
  -drive file=/usr/local/share/qemu/edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \  
  \  
  -drive file=ovmf_vars.fd,if=pflash,format=raw \  
  -serial telnet::4444,server,nowait \  
  -drive if=none,file=disk.qcow2,format=qcow2,id=hd0 \  
  -device virtio-blk-device,drive=hd0,serial="dummyserial" \  
  -device virtio-net-device,netdev=net0 \  
  -netdev user,id=net0 \  
  -vga none -device ramfb \  
  -cdrom /path/to/ubuntu.iso \  
  -device usb-ehci -device usb-kbd -device usb-mouse -usb \  
  -monitor stdio
```

Arguments used in the previous command:

1. -accel name[,prop=value[,...]]

To enable accelerator. Depending on the target architecture, kvm, xen, hax, hvf, nvmm, whpx or tcg can be available. By default, tcg is used. If there is more than one accelerator specified, the next one is used if the previous one fails to initialize.

2. -m [size=]megs[,slots=n,maxmem=size]

Sets guest startup RAM size to megs megabytes. Default is 128 MiB. Optionally, a suffix of "M" or "G" can be used to signify a value in megabytes or gigabytes respectively. Optional pair slots, maxmem could be used to set amount of hotpluggable memory slots and maximum amount of memory. Note that maxmem must be aligned to the page size.

3. -cpu model

Select CPU model (-cpu help for list and additional feature selection)

4. -drive option[,option[,option[,...]]]

Define a new drive. This includes creating a block driver node (the backend) as well as a guest device, and is mostly a shortcut for defining the corresponding -blockdev and -device options.

5. -serial dev

Redirect the virtual serial port to host character device dev. The default device is vc in graphical mode and stdio in non graphical mode.

6. -netdev user,id=id[,option][,option][,...]

Configure user mode host network backend which requires no administrator privilege to run.

7. -vga type

Select type of VGA card to emulate.

8. -cdrom file

Use file as CD-ROM image (you cannot use -hdc and -cdrom at the same time). You can use the host CD-ROM by using /dev/cdrom as filename.

9. -device driver[,prop[=value][,...]]

Add device driver. prop=value sets driver properties. Valid properties depend on the driver.

10. -monitor dev :

Redirect the monitor to host device dev (same devices as the serial port). The default device is vc in graphical mode and stdio in non graphical mode. Use -monitor none to disable the default monitor.

Steps to Enable Docker:

1. Downloaded Docker Desktop for Apple Silicon from: <https://docs.docker.com/desktop/mac/apple-silicon/>
2. Dragged the Docker App into the Applications Folder
3. Double clicked the Docker App to start the Docker Engine
4. Opened terminal and ran the following command to pull Ubuntu Image with Sysbench:
 - o docker pull ubuntu
5. Ran ubuntu using :
 - a. docker run -it ubuntu
6. Updated the system using below command:
 - o apt-get update

7. Installed sysbench using:
apt-get install sysbench

Some Important Docker Container Commands:

- ls: to list running containers
- create: to create container from image
- start: to start a container
- run: to create new container and start it
- stop: stop main process in container
- rm: to delete a stopped container

Proof of Experiments:

To provide comparison in performance between OS and System Virtualization, ‘sysbench’ tool was utilized in Ubuntu (at Docker and Qemu). Below you can find the 2 benchmarking experiments by running the commands that test the CPU and File I/O performance in 3 different scenarios. Each scenario was tested 5 times.

EXP-1: CPU Performance Check by running the Max Prime Test:

This test finds the maximum prime number value for a given number limit ‘n’.

QEMU Results:

Scenario 1 Command: “sysbench --test=cpu --cpu-max-prime=**30000** run”

Test1 (In QEMU: Max Prime Test with limit =30000**)**

```
aanch@ubuntu:~$ sysbench --test=cpu --cpu-max-prime=30000 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: 30000
Initializing worker threads...
Threads started!

CPU speed:
  events per second: 2465.95

General statistics:
  total time:          10.0002s
  total number of events: 24663

Latency (ms):
  min:                 0.39
  avg:                 0.41
  max:                 1.45
  95th percentile:    0.42
  sum:                9990.08

Threads fairness:
  events (avg/stddev): 24663.0000/0.00
  execution time (avg/stddev): 9.9901/0.00
```

Test-2 (In QEMU: Max Prime Test with limit =**30000**)

```
aanch@ubuntu:~$ sysbench --test=cpu --cpu-max-prime=30000 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: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 2464.41

General statistics:
  total time:          10.00005s
  total number of events: 24647

Latency (ms):
  min:                0.39
  avg:                0.41
  max:                0.67
  95th percentile:    0.42
  sum:                9993.57

Threads fairness:
  events (avg/stddev): 24647.0000/0.00
  execution time (avg/stddev): 9.9936/0.00
```

Test-3 (In QEMU: Max Prime Test with limit =30000)

QEMU

```
banch@ubuntu:~$ sysbench --test=cpu --cpu-max-prime=30000 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: 30000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 2466.02

General statistics:
    total time:          10.00001s
    total number of events: 24662

Latency (ms):
    min:                  0.39
    avg:                  0.41
    max:                  0.61
    95th percentile:      0.42
    sum:                 9994.24

Threads fairness:
    events (avg/stddev): 24662.0000/0.00
    execution time (avg/stddev): 9.9942/0.00
```

Test-4 (In QEMU: Max Prime Test with limit =30000)

```
aanch@ubuntu:~$ sysbench --test=cpu --cpu-max-prime=30000 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: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 2464.61

General statistics:
  total time:          10.0003s
  total number of events: 24648

Latency (ms):
  min:                 0.39
  avg:                 0.41
  max:                 0.61
  95th percentile:    0.42
  sum:                9994.07

Threads fairness:
  events (avg/stddev): 24648.0000/0.00
  execution time (avg/stddev): 9.9941/0.00
```

Test-5 (In QEMU: Max Prime Test with limit =30000)

```
aanch@ubuntu:~$ sysbench --test=cpu --cpu-max-prime=30000 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: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 2467.39

General statistics:
  total time:          10.0003s
  total number of events: 24676

Latency (ms):
  min:                 0.39
  avg:                 0.41
  max:                 0.60
  95th percentile:    0.42
  sum:                9994.47

Threads fairness:
  events (avg/stddev): 24676.0000/0.00
  execution time (avg/stddev): 9.9945/0.00
```

Scenario 2 Command: “sysbench --test=cpu --cpu-max-prime=**50000** run”

Test-1 (In QEMU: Max Prime Test with limit =50000**)**

```
Test #: 1
WARNING: the --test option is deprecated. You can pass a script file
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: 50000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 1220.64

General statistics:
    total time:                      10.0008s
    total number of events:           12208

Latency (ms):
    min:                            0.79
    avg:                            0.82
    max:                            1.21
    95th percentile:                0.84
    sum:                           9997.43

Threads fairness:
    events (avg/stddev):          12208.0000/0.00
    execution time (avg/stddev):  9.9974/0.00
```

Test-2 (In QEMU: Max Prime Test with limit =50000)

Test #: 2

WARNING: the --test option is deprecated. You can pass 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: 50000

Initializing worker threads...

Threads started!

CPU speed:

events per second: 1213.35

General statistics:

total time: 10.0006s

total number of events: 12135

Latency (ms):

min: 0.79

avg: 0.82

max: 1.41

95th percentile: 0.86

sum: 9995.44

Threads fairness:

events (avg/stddev): 12135.0000/0.00

execution time (avg/stddev): 9.9954/0.00

Test-3 (In QEMU: Max Prime Test with limit =50000)

Test #: 3

WARNING: the --test option is deprecated. You can pass a script name
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: 50000

Initializing worker threads...

Threads started!

CPU speed:

events per second: 1218.16

General statistics:

total time: 10.0006s

total number of events: 12183

Latency (ms):

min: 0.79

avg: 0.82

max: 1.44

95th percentile: 0.86

sum: 9996.03

Threads fairness:

events (avg/stddev): 12183.0000/0.00

execution time (avg/stddev): 9.9960/0.00

Test-4 (In QEMU: Max Prime Test with limit =50000)

Test #: 4

WARNING: the --test option is deprecated. You can pass a script name
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: 50000

Initializing worker threads...

Threads started!

CPU speed:

events per second: 1221.77

General statistics:

total time: 10.0005s

total number of events: 12219

Latency (ms):

min: 0.79

avg: 0.82

max: 1.53

95th percentile: 0.84

sum: 9996.55

Threads fairness:

events (avg/stddev): 12219.0000/0.00

execution time (avg/stddev): 9.9965/0.00

Test-5 (In QEMU: Max Prime Test with limit =50000)

```
Test #: 5
WARNING: the --test option is deprecated. You can pass a script
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: 50000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 1223.75

General statistics:
    total time:          10.0007s
    total number of events: 12239

Latency (ms):
    min:                 0.79
    avg:                 0.82
    max:                 1.36
    95th percentile:     0.84
    sum:                9996.93

Threads fairness:
    events (avg/stddev): 12239.0000/0.00
    execution time (avg/stddev): 9.9969/0.00
```

Scenario 3 Command: “sysbench --test=cpu --cpu-max-prime=**10000** run”

Test-1 (In QEMU: Max Prime Test with limit =**10000**)

Test #: 1

WARNING: the --test option is deprecated. You can pass a test name 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: 10947.95

General statistics:

total time: 10.0001s

total number of events: 109486

Latency (ms):

min: 0.09

avg: 0.09

max: 0.29

95th percentile: 0.10

sum: 9987.54

Threads fairness:

events (avg/stddev): 109486.0000/0.00

execution time (avg/stddev): 9.9875/0.00

Test-2 (In QEMU: Max Prime Test with limit =10000)

Test #: 2

WARNING: the --test option is deprecated. You can pass a
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: 10969.42

General statistics:

total time: 10.0001s

total number of events: 109701

Latency (ms):

min: 0.09

avg: 0.09

max: 0.21

95th percentile: 0.10

sum: 9988.07

Threads fairness:

events (avg/stddev): 109701.0000/0.00

execution time (avg/stddev): 9.9881/0.00

Test-3 (In QEMU: Max Prime Test with limit =10000)

```
Test #: 3
WARNING: the --test option is deprecated. You can pass a
about 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: 10955.57

General statistics:
  total time:          10.0001s
  total number of events: 109563

Latency (ms):
  min:                0.09
  avg:                0.09
  max:                1.46
  95th percentile:    0.10
  sum:               9988.33

Threads fairness:
  events (avg/stddev):   109563.0000/0.00
  execution time (avg/stddev): 9.9883/0.00
```

Test-4 (In QEMU: Max Prime Test with limit =10000)

```
Test #: 4
WARNING: the --test option is deprecated. You can pass a
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: 10934.59

General statistics:
    total time:          10.0001s
    total number of events: 109353

Latency (ms):
    min:                  0.09
    avg:                  0.09
    max:                  0.17
    95th percentile:      0.10
    sum:                 9988.37

Threads fairness:
    events (avg/stddev): 109353.0000/0.00
    execution time (avg/stddev): 9.9884/0.00
```

Test-5 (In QEMU: Max Prime Test with limit =10000)

Test #: 5

WARNING: the --test option is deprecated. You can pass 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: 10960.87

General statistics:

total time: 10.0001s

total number of events: 109616

Latency (ms):

min: 0.09

avg: 0.09

max: 0.20

95th percentile: 0.10

sum: 9988.18

Threads fairness:

events (avg/stddev): 109616.0000/0.00

execution time (avg/stddev): 9.9882/0.00

Docker Results:

Scenario 1 Command: “sysbench --test=cpu --cpu-max-prime=**30000** run”

Test 1 (In Docker: Max Prime Test with limit =30000**)**

Test #: 1

WARNING: the --test option is deprecated. You can pass a s
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: 30000

Initializing worker threads...

Threads started!

CPU speed:

events per second: 2450.93

General statistics:

total time: 10.0005s

total number of events: 24512

Latency (ms):

min: 0.39

avg: 0.41

max: 1.86

95th percentile: 0.42

sum: 9994.81

Threads fairness:

events (avg/stddev): 24512.0000/0.00

execution time (avg/stddev): 9.9948/0.00

Test 2(In Docker: Max Prime Test with limit =30000**)**

```
Test #: 2
WARNING: the --test option is deprecated. You can pass a scr
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: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 2445.76

General statistics:
  total time:          10.0005s
  total number of events: 24461

Latency (ms):
  min:                0.39
  avg:                0.41
  max:                1.56
  95th percentile:    0.42
  sum:               9995.20

Threads fairness:
  events (avg/stddev): 24461.0000/0.00
  execution time (avg/stddev): 9.9952/0.00
```

Test 3 (In Docker: Max Prime Test with limit =30000)

Test #: 3

WARNING: the --test option is deprecated. You can pass a script directly to 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: 30000

Initializing worker threads...

Threads started!

CPU speed:

events per second: 2461.29

General statistics:

total time: 10.0003s

total number of events: 24616

Latency (ms):

min: 0.39

avg: 0.41

max: 1.63

95th percentile: 0.42

sum: 9995.92

Threads fairness:

events (avg/stddev): 24616.0000/0.00

execution time (avg/stddev): 9.9959/0.00

Test4 (In Docker: Max Prime Test with limit =**30000**)

Test #: 4

WARNING: the --test option is deprecated. You can pass a command directly to 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: 30000

Initializing worker threads...

Threads started!

CPU speed:

events per second: 2444.06

General statistics:

total time:	10.0003s
total number of events:	24444

Latency (ms):

min:	0.39
avg:	0.41
max:	1.48
95th percentile:	0.42
sum:	9995.02

Threads fairness:

events (avg/stddev):	24444.0000/0.00
execution time (avg/stddev):	9.9950/0.00

Test 5(In Docker: Max Prime Test with limit =30000)

Test #: 5

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: 30000

Initializing worker threads...

Threads started!

CPU speed:

events per second: 2453.43

General statistics:

total time: 10.0002s

total number of events: 24538

Latency (ms):

min: 0.39

avg: 0.41

max: 2.72

95th percentile: 0.42

sum: 9995.04

Threads fairness:

events (avg/stddev): 24538.0000/0.00

execution time (avg/stddev): 9.9950/0.00

Scenario 2 Command: “sysbench --test=cpu --cpu-max-prime=**50000** run”

Test 1 (In Docker: Max Prime Test with limit =50000**)**

Test #: 1

WARNING: the --test option is deprecated. You can pass a script
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: 1219.53

General statistics:

total time: 10.0007s

total number of events: 12197

Latency (ms):

min: 0.79

avg: 0.82

max: 4.36

95th percentile: 0.84

sum: 9996.29

Threads fairness:

events (avg/stddev): 12197.0000/0.00

execution time (avg/stddev): 9.9963/0.00

Test 2 (In Docker : Max Prime Test with limit =50000**)**

```
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: 1219.76  
  
General statistics:  
    total time: 10.0004s  
    total number of events: 12199  
  
Latency (ms):  
    min: 0.79  
    avg: 0.82  
    max: 4.08  
    95th percentile: 0.84  
    sum: 9995.37  
  
Threads fairness:  
    events (avg/stddev): 12199.0000/0.00  
    execution time (avg/stddev): 9.9954/0.00
```

Test 3(In Docker: Max Prime Test with limit =50000)

```
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: 1220.05  
  
General statistics:  
    total time: 10.0003s  
    total number of events: 12202  
  
Latency (ms):  
    min: 0.79  
    avg: 0.82  
    max: 3.37  
    95th percentile: 0.84  
    sum: 9994.97  
  
Threads fairness:  
    events (avg/stddev): 12202.0000/0.00  
    execution time (avg/stddev): 9.9950/0.00
```

```
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: 1218.06  
  
General statistics:  
    total time: 10.0003s  
    total number of events: 12182  
  
Latency (ms):  
    min: 0.79  
    avg: 0.82  
    max: 3.31  
    95th percentile: 0.84  
    sum: 9985.78  
  
Threads fairness:  
    events (avg/stddev): 12182.0000/0.00  
    execution time (avg/stddev): 9.9858/0.00
```

Test 5 (In Docker : Max Prime Test with limit =50000)

```
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: 1222.21  
  
General statistics:  
    total time: 10.0006s  
    total number of events: 12224  
  
Latency (ms):  
    min: 0.79  
    avg: 0.82  
    max: 2.93  
    95th percentile: 0.84  
    sum: 9996.71  
  
Threads fairness:  
    events (avg/stddev): 12224.0000/0.00  
    execution time (avg/stddev): 9.9967/0.00
```

Scenario 3 Command: “sysbench --test=cpu --cpu-max-prime=10000 run”

Test 1(In Docker: Max Prime Test with limit =10000)**Test #:** 1

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: 10000

Initializing worker threads...

Threads started!

CPU speed:

events per second: 10952.52

General statistics:

total time: 10.0002s

total number of events: 109535

Latency (ms):

min: 0.09

avg: 0.09

max: 0.46

95th percentile: 0.10

sum: 9987.37

Threads fairness:

events (avg/stddev): 109535.0000/0.00

execution time (avg/stddev): 9.9874/0.00

Test 2(In Docker: Max Prime Test with limit =10000)

Test #: 2

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: 10000

Initializing worker threads...

Threads started!

CPU speed:

events per second: 10919.33

General statistics:

total time: 10.0002s

total number of events: 109203

Latency (ms):

min: 0.09

avg: 0.09

max: 1.02

95th percentile: 0.10

sum: 9987.01

Threads fairness:

events (avg/stddev): 109203.0000/0.00

execution time (avg/stddev): 9.9870/0.00

Test 3(In Docker: Max Prime Test with limit =**10000**)

Test #: 3

WARNING: the --test option is deprecated. You can pass test 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: 10938.22

General statistics:

total time: 10.0002s

total number of events: 109393

Latency (ms):

min: 0.09

avg: 0.09

max: 0.74

95th percentile: 0.10

sum: 9987.13

Threads fairness:

events (avg/stddev): 109393.0000/0.00

execution time (avg/stddev): 9.9871/0.00

Test 4(In Docker: Max Prime Test with limit =**10000**)

Test #: 4

WARNING: the --test option is deprecated. You can pass test 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: 10825.39

General statistics:

total time: 10.0002s

total number of events: 108264

Latency (ms):

min: 0.09

avg: 0.09

max: 0.65

95th percentile: 0.10

sum: 9977.96

Threads fairness:

events (avg/stddev): 108264.0000/0.00

execution time (avg/stddev): 9.9780/0.00

Test 5(In Docker: Max Prime Test with limit =10000)

Test #: 5

WARNING: the --test option is deprecated. You can pass a test 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: 10877.65

General statistics:

total time: 10.0002s

total number of events: 108786

Latency (ms):

min: 0.09

avg: 0.09

max: 0.75

95th percentile: 0.10

sum: 9982.43

Threads fairness:

events (avg/stddev): 108786.0000/0.00

execution time (avg/stddev): 9.9824/0.00

EXP-2: File I/O Performance Check by running the File I/O Test:

This test will determine the throughput for file read and write commands.

QEMU Results:

Scenario 1 Command :

- a. “sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct prepare”
- b. “sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct run”
- c. “sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct cleanup”

** at HOST terminal ran the below command before starting each test to clear the cache:
“sync && sudo purge”

Test-1 (QEMU FILE I/O Test with 3GB)

```
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:          5889.77
  writes/s:         3926.51
  fsyncs/s:         12568.04

Throughput:
  read, MiB/s:      92.03
  written, MiB/s:   61.35

General statistics:
  total time:        30.0031s
  total number of events: 671485

Latency (ms):
  min:                0.02
  avg:                0.04
  max:                30.70
  95th percentile:    0.13
  sum:               29829.73

Threads fairness:
  events (avg/stddev): 671485.0000/0.00
  execution time (avg/stddev): 29.8297/0.00
```

Test-2 (QEMU FILE I/O Test with 3GB)

QEMU

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:	5484.45
writes/s:	3656.28
fsyncs/s:	11701.80

Throughput:

read, MiB/s:	85.69
written, MiB/s:	57.13

General statistics:

total time:	30.0037s
total number of events:	625237

Latency (ms):

min:	0.01
avg:	0.05
max:	28.24
95th percentile:	0.13
sum:	29846.26

Threads fairness:

events (avg/stddev):	625237.0000/0.00
execution time (avg/stddev):	29.8463/0.00

Test-3(QEMU FILE I/O Test with 3GB)

QEMU

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:	5389.41
writes/s:	3592.94
fsyncs/s:	11497.44

Throughput:

read, MiB/s:	84.21
written, MiB/s:	56.14

General statistics:

total time:	30.0027s
total number of events:	614333

Latency (ms):

min:	0.01
avg:	0.05
max:	27.22
95th percentile:	0.13
sum:	29846.41

Threads fairness:

events (avg/stddev):	614333.0000/0.00
execution time (avg/stddev):	29.8464/0.00

Test – 4: (QEMU FILE I/O Test with 3GB)

QEMU

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:	5357.88
writes/s:	3571.90
fsyncs/s:	11433.28

Throughput:

read, MiB/s:	83.72
written, MiB/s:	55.81

General statistics:

total time:	30.0030s
total number of events:	610838

Latency (ms):

min:	0.01
avg:	0.05
max:	33.85
95th percentile:	0.13
sum:	29842.23

Threads fairness:

events (avg/stddev):	610838.0000/0.00
execution time (avg/stddev):	29.8422/0.00

Test-5(QEMU FILE I/O Test with 3GB)

```
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:          5357.88
  writes/s:         3571.90
  fsyncs/s:        11433.28

Throughput:
  read, MiB/s:      83.72
  written, MiB/s:   55.81

General statistics:
  total time:       30.0030s
  total number of events: 610838

Latency (ms):
  min:              0.01
  avg:              0.05
  max:             33.85
  95th percentile:  0.13
  sum:            29842.23

Threads fairness:
  events (avg/stddev): 610838.0000/0.00
  execution time (avg/stddev): 29.8422/0.00
```

Scenario 2 Command :

- a. “sysbench --test=fileio --file-total-size=2G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct prepare”
- b. “sysbench --test=fileio --file-total-size=2G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct run”
- c. “sysbench --test=fileio --file-total-size=2G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct cleanup”

** at HOST terminal ran the below command before starting each test to clear the cache:
“sync && sudo purge”

Test1(QEMU FILE I/O Test with 2GB)

Threads started!

File operations:

reads/s:	5725.39
writes/s:	3816.91
fsyncs/s:	12218.15

Throughput:

read, MiB/s:	89.46
written, MiB/s:	59.64

General statistics:

total time:	30.0034s
total number of events:	652770

Latency (ms):

min:	0.02
avg:	0.05
max:	40.08
95th percentile:	0.15
sum:	29834.66

Threads fairness:

events (avg/stddev):	652770.0000/0.00
execution time (avg/stddev):	29.8347/0.00

Test 2(QEMU FILE I/O Test with 2GB)

```
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:          5441.46
  writes/s:         3627.64
  fsyncs/s:        11612.55

Throughput:
  read, MiB/s:      85.02
  written, MiB/s:   56.68

General statistics:
  total time:       30.0025s
  total number of events: 620383

Latency (ms):
  min:              0.02
  avg:              0.05
  max:             31.78
  95th percentile:  0.15
  sum:            29837.40

Threads fairness:
  events (avg/stddev): 620383.0000/0.00
  execution time (avg/stddev): 29.8374/0.00
```

Test 3(QEMU FILE I/O Test with 2GB)

```
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:          5889.75
  writes/s:         3926.49
  fsyncs/s:         12568.07

Throughput:
  read, MiB/s:      92.03
  written, MiB/s:   61.35

General statistics:
  total time:        30.0031s
  total number of events: 671483

Latency (ms):
  min:                0.02
  avg:                0.04
  max:                31.70
  95th percentile:    0.15
  sum:               29840.64

Threads fairness:
  events (avg/stddev): 671483.0000/0.00
  execution time (avg/stddev): 29.8406/0.00
```

Test 4(QEMU FILE I/O Test with 2GB)

```
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:          6689.32
  writes/s:         4459.55
  fsyncs/s:         14273.38

Throughput:
  read, MiB/s:      104.52
  written, MiB/s:   69.68

General statistics:
  total time:        30.0026s
  total number of events: 762617

Latency (ms):
  min:                0.02
  avg:                0.04
  max:                5.41
  95th percentile:    0.14
  sum:               29824.49

Threads fairness:
  events (avg/stddev): 762617.0000/0.00
  execution time (avg/stddev): 29.8245/0.00
```

Test 5(QEMU FILE I/O Test with 2GB)

```
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:          6171.37
  writes/s:         4114.25
  fsyncs/s:         13167.03

Throughput:
  read, MiB/s:      96.43
  written, MiB/s:   64.29

General statistics:
  total time:        30.0026s
  total number of events: 703523

Latency (ms):
  min:                0.02
  avg:                0.04
  max:               12.04
  95th percentile:    0.15
  sum:              29830.73

Threads fairness:
  events (avg/stddev): 703523.0000/0.00
  execution time (avg/stddev): 29.8307/0.00
```

Scenario 3 Command :

- a. “sysbench --test=fileio --file-total-size=1G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct prepare”
- b. “sysbench --test=fileio --file-total-size=1G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct run”
- c. “sysbench --test=fileio --file-total-size=1G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct cleanup”

** at HOST terminal ran the below command before starting each test to clear the cache:
“sync && sudo purge”

Test 1(QEMU FILE I/O Test with 1GB)

```
Threads started!

File operations:
    reads/s:          6768.32
    writes/s:         4512.23
    fsyncs/s:        14440.93

Throughput:
    read, MiB/s:      105.76
    written, MiB/s:   70.50

General statistics:
    total time:       30.0031s
    total number of events: 771609

Latency (ms):
    min:              0.01
    avg:              0.04
    max:             36.74
    95th percentile:  0.14
    sum:            29824.81

Threads fairness:
    events (avg/stddev): 771609.0000/0.00
    execution time (avg/stddev): 29.8248/0.00
```

Test 2(QEMU FILE I/O Test with 1GB)

Threads started!

File operations:

reads/s:	9365.07
writes/s:	6243.38
fsyncs/s:	19980.96

Throughput:

read, MiB/s:	146.33
written, MiB/s:	97.55

General statistics:

total time:	30.0025s
total number of events:	1067660

Latency (ms):

min:	0.02
avg:	0.03
max:	4.08
95th percentile:	0.03
sum:	29783.25

Threads fairness:

events (avg/stddev):	1067660.0000/0.00
execution time (avg/stddev):	29.7833/0.00

Test 3(QEMU FILE I/O Test with 1GB)

Threads started!

File operations:

reads/s:	9369.69
writes/s:	6246.46
fsyncs/s:	19991.23

Throughput:

read, MiB/s:	146.40
written, MiB/s:	97.60

General statistics:

total time:	30.0030s
total number of events:	1068220

Latency (ms):

min:	0.02
avg:	0.03
max:	4.35
95th percentile:	0.03
sum:	29783.77

Threads fairness:

events (avg/stddev):	1068220.0000/0.00
execution time (avg/stddev):	29.7838/0.00

Test 4(QEMU FILE I/O Test with 1GB)

Threads started!

File operations:

reads/s:	9383.37
writes/s:	6255.58
fsyncs/s:	20021.27

Throughput:

read, MiB/s:	146.62
written, MiB/s:	97.74

General statistics:

total time:	30.0028s
total number of events:	1069796

Latency (ms):

min:	0.02
avg:	0.03
max:	32.27
95th percentile:	0.03
sum:	29792.46

Threads fairness:

events (avg/stddev):	1069796.0000/0.00
execution time (avg/stddev):	29.7925/0.00

Test 5(QEMU FILE I/O Test with 1GB)

Threads started!

File operations:

reads/s:	9041.05
writes/s:	6027.37
fsyncs/s:	19289.21

Throughput:

read, MiB/s:	141.27
written, MiB/s:	94.18

General statistics:

total time:	30.0025s
total number of events:	1030709

Latency (ms):

min:	0.02
avg:	0.03
max:	16.05
95th percentile:	0.03
sum:	29776.85

Threads fairness:

events (avg/stddev):	1030709.0000/0.00
execution time (avg/stddev):	29.7768/0.00

Docker Results:

Scenario 1 Command :

- a. “sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct prepare”
- b. “sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct run”
- c. “sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct cleanup”

** at HOST terminal ran the below command before starting each test to clear the cache:
“sync && sudo purge”

Test-1(Docker FILE I/O Test with 3GB)

```
Extra file open flags: directio
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:          3725.31
  writes/s:         2483.54
  fsyncs/s:        7950.98

Throughput:
  read, MiB/s:      58.21
  written, MiB/s:   38.81

General statistics:
  total time:       30.0051s
  total number of events: 424746

Latency (ms):
  min:              0.03
  avg:              0.07
  max:              3.91
  95th percentile: 0.14
  sum:             29902.51

Threads fairness:
  events (avg/stddev): 424746.0000/0.00
  execution time (avg/stddev): 29.9025/0.00
```

Test2(Docker FILE I/O Test with 3GB)

Threads started!

File operations:

reads/s:	4167.94
writes/s:	2778.63
fsyncs/s:	8894.17

Throughput:

read, MiB/s:	65.12
written, MiB/s:	43.42

General statistics:

total time:	30.0056s
total number of events:	475192

Latency (ms):

min:	0.03
avg:	0.06
max:	2.18
95th percentile:	0.14
sum:	29895.76

Threads fairness:

events (avg/stddev):	475192.0000/0.00
execution time (avg/stddev):	29.8958/0.00

Test 3(Docker FILE I/O Test with 3GB)

Threads started!

File operations:

reads/s:	4018.19
writes/s:	2678.80
fsyncs/s:	8574.19

Throughput:

read, MiB/s:	62.78
written, MiB/s:	41.86

General statistics:

total time:	30.0057s
total number of events:	458104

Latency (ms):

min:	0.03
avg:	0.07
max:	48.64
95th percentile:	0.14
sum:	29897.32

Threads fairness:

events (avg/stddev):	458104.0000/0.00
execution time (avg/stddev):	29.8973/0.00

```
Extra file open flags: directio
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:          4049.92
  writes/s:         2699.95
  fsyncs/s:         8642.17

Throughput:
  read, MiB/s:      63.28
  written, MiB/s:   42.19

General statistics:
  total time:        30.0067s
  total number of events: 461745

Latency (ms):
  min:                0.03
  avg:                0.06
  max:               28.83
  95th percentile:    0.14
  sum:              29898.36

Threads fairness:
  events (avg/stddev): 461745.0000/0.00
  execution time (avg/stddev): 29.8984/0.00
```

Threads started!

File operations:

reads/s:	4280.35
writes/s:	2853.59
fsyncs/s:	9132.98

Throughput:

read, MiB/s:	66.88
written, MiB/s:	44.59

General statistics:

total time:	30.0058s
total number of events:	487984

Latency (ms):

min:	0.03
avg:	0.06
max:	6.14
95th percentile:	0.14
sum:	29894.01

Threads fairness:

events (avg/stddev):	487984.0000/0.00
execution time (avg/stddev):	29.8940/0.00

Scenario 2 Command :

- a. “sysbench --test=fileio --file-total-size=2G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct prepare”
- b. “sysbench --test=fileio --file-total-size=2G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct run”
- c. “sysbench --test=fileio --file-total-size=2G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct cleanup”

** at HOST terminal ran the below command before starting each test to clear the cache:
“sync && sudo purge”

Test 1(Docker FILE I/O Test with 2GB)

Threads started!

File operations:

reads/s:	4375.13
writes/s:	2916.75
fsyncs/s:	9333.65

Throughput:

read, MiB/s:	68.36
written, MiB/s:	45.57

General statistics:

total time:	30.0053s
total number of events:	498737

Latency (ms):

min:	0.03
avg:	0.06
max:	40.68
95th percentile:	0.14
sum:	29893.53

Threads fairness:

events (avg/stddev):	498737.0000/0.00
execution time (avg/stddev):	29.8935/0.00

Test 2(Docker FILE I/O Test with 2GB)

Threads started!

File operations:

reads/s:	4444.72
writes/s:	2963.15
fsyncs/s:	9482.71

Throughput:

read, MiB/s:	69.45
written, MiB/s:	46.30

General statistics:

total time:	30.0060s
total number of events:	506701

Latency (ms):

min:	0.03
avg:	0.06
max:	56.71
95th percentile:	0.14
sum:	29888.73

Threads fairness:

events (avg/stddev):	506701.0000/0.00
execution time (avg/stddev):	29.8887/0.00

2GiB 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:	4876.78
writes/s:	3251.16
fsyncs/s:	10404.15

Throughput:

read, MiB/s:	76.20
written, MiB/s:	50.80

General statistics:

total time:	30.0059s
total number of events:	555955

Latency (ms):

min:	0.03
avg:	0.05
max:	2.11
95th percentile:	0.14
sum:	29881.24

Threads fairness:

events (avg/stddev):	555955.0000/0.00
execution time (avg/stddev):	29.8812/0.00

Test 4(Docker FILE I/O Test with 2GB)

2GiB 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:	4668.62
writes/s:	3112.39
fsyncs/s:	9960.49

Throughput:

read, MiB/s:	72.95
written, MiB/s:	48.63

General statistics:

total time:	30.0059s
total number of events:	532233

Latency (ms):

min:	0.03
avg:	0.06
max:	9.55
95th percentile:	0.14
sum:	29885.53

Threads fairness:

events (avg/stddev):	532233.0000/0.00
execution time (avg/stddev):	29.8855/0.00

Test 5(Docker FILE I/O Test with 2GB)

2GiB 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 request
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:	4954.75
writes/s:	3303.14
fsyncs/s:	10570.49

Throughput:

read, MiB/s:	77.42
written, MiB/s:	51.61

General statistics:

total time:	30.0059s
total number of events:	564847

Latency (ms):

min:	0.03
avg:	0.05
max:	2.24
95th percentile:	0.14
sum:	29878.72

Threads fairness:

events (avg/stddev):	564847.0000/0.00
execution time (avg/stddev):	29.8787/0.00

Scenario 3 Command :

- a. “sysbench --test=fileio --file-total-size=1G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct prepare”
- b. “sysbench --test=fileio --file-total-size=1G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct run”
- c. “sysbench --test=fileio --file-total-size=1G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct cleanup”

** at HOST terminal ran the below command before starting each test to clear the cache:
“sync && sudo purge”

Test 1(Docker FILE I/O Test with 1GB)

```
1GiB 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:          5034.34
  writes/s:         3356.21
  fsyncs/s:        10741.14

Throughput:
  read, MiB/s:      78.66
  written, MiB/s:   52.44

General statistics:
  total time:       30.0058s
  total number of events: 573947

Latency (ms):
  min:              0.03
  avg:              0.05
  max:             120.58
  95th percentile:  0.14
  sum:            29874.71

Threads fairness:
  events (avg/stddev): 573947.0000/0.00
  execution time (avg/stddev): 29.8747/0.00
```

Test 2(Docker FILE I/O Test with 1GB)

```
1GiB 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:	6094.86
writes/s:	4063.24
fsyncs/s:	13003.66

Throughput:

read, MiB/s:	95.23
written, MiB/s:	63.49

General statistics:

total time:	30.0050s
total number of events:	694855

Latency (ms):

min:	0.03
avg:	0.04
max:	4.15
95th percentile:	0.04
sum:	29856.15

Threads fairness:

events (avg/stddev):	694855.0000/0.00
execution time (avg/stddev):	29.8562/0.00

Test 3(Docker FILE I/O Test with 1GB)

```
1GiB 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:	5650.96
writes/s:	3767.31
fsyncs/s:	12057.81

Throughput:

read, MiB/s:	88.30
written, MiB/s:	58.86

General statistics:

total time:	30.0050s
total number of events:	644273

Latency (ms):

min:	0.03
avg:	0.05
max:	27.75
95th percentile:	0.05
sum:	29863.62

Threads fairness:

events (avg/stddev):	644273.0000/0.00
execution time (avg/stddev):	29.8636/0.00

Test 4(Docker FILE I/O Test with 1GB)

Threads started!

File operations:

reads/s:	6144.94
writes/s:	4096.64
fsyncs/s:	13113.19

Throughput:

read, MiB/s:	96.01
written, MiB/s:	64.01

General statistics:

total time:	30.0053s
total number of events:	700651

Latency (ms):

min:	0.03
avg:	0.04
max:	25.58
95th percentile:	0.04
sum:	29856.51

Threads fairness:

events (avg/stddev):	700651.0000/0.00
execution time (avg/stddev):	29.8565/0.00

1GiB 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:	6152.89
writes/s:	4101.93
fsyncs/s:	13129.80

Throughput:

read, MiB/s:	96.14
written, MiB/s:	64.09

General statistics:

total time:	30.0049s
total number of events:	701537

Latency (ms):

min:	0.03
avg:	0.04
max:	2.18
95th percentile:	0.04
sum:	29856.56

Threads fairness:

events (avg/stddev):	701537.0000/0.00
execution time (avg/stddev):	29.8566/0.00

Measurements in three different scenarios for each virtualization technology:

CPU Test with LIMIT = 30000 (QEMU Results)

Test #	CPU Speed (events/s)	Total Time (s)	Avg Latency (ms)
1	2465.95	10.0002	0.41
2	2464.41	10.0005	0.41
3	2466.02	10.0001	0.41
4	2464.61	10.0003	0.41
5	2467.39	10.0003	0.41
Min	2464.41	10.0001	0.41
Max	2467.39	10.0005	0.41
Avg	2465.676	10.00028	0.41
Std Dev	1.21143716	0.00014832	0

CPU Test with LIMIT = 50000 (QEMU Results)

Test #	CPU Speed (events/s)	Total Time (s)	Avg Latency (ms)
1	1220.64	10.0008	0.82
2	1213.35	10.0006	0.82
3	1218.16	10.0006	0.82
4	1221.77	10.0005	0.82
5	1223.75	10.0007	0.82
Min	1213.35	10.0005	0.82
Max	1223.75	10.0008	0.82
Avg	1219.534	10.00064	0.82
Std Dev	4.00397677	0.00011402	0

CPU Test with LIMIT = 10000 (QEMU Results)

Test #	CPU Speed (events/s)	Total Time (s)	Avg Latency (ms)
1	10969.42	10.0001	0.09
2	10969.42	10.0001	0.09
3	10955.57	10.0001	0.09
4	10934.59	10.0001	0.09
5	10969.87	10.0001	0.09
Min	10934.59	10.0001	0.09
Max	10969.87	10.0001	0.09
Avg	10959.774	10.0001	0.09
Std Dev	15.3291171	0	0

CPU Test with LIMIT = 30000 (Docker Results)

Test #	CPU Speed (events/s)	Total Time (s)	Avg Latency (ms)
1	2450.93	10.0005	0.41
2	2445.76	10.0005	0.41
3	2461.29	10.0003	0.41
4	2444.06	10.0003	0.41
5	2453.43	10.0002	0.41
Min	2444.06	10.0002	0.41
Max	2461.29	10.0005	0.41
Avg	2451.094	10.00036	0.41
Std Dev	6.8441749	0.00013416	0

CPU Test with LIMIT = 50000 (Docker Results)

Test #	CPU Speed (events/s)	Total Time (s)	Avg Latency (ms)
1	1219.53	10.0007	0.82
2	1219.76	10.0004	0.82
3	1220.05	10.0003	0.82
4	1218.06	10.0003	0.82
5	1222.21	10.0006	0.82
Min	1218.06	10.0003	0.82
Max	1222.21	10.0007	0.82
Avg	1219.922	10.00046	0.82
Std Dev	1.49149925	0.00018166	0

CPU Test with LIMIT = 10000 (Docker Results)

Test #	CPU Speed (events/s)	Total Time (s)	Avg Latency (ms)
1	10952.52	10.0002	0.09
2	10919.33	10.0002	0.09
3	10938.22	10.0002	0.09
4	10825.39	10.0002	0.09
5	10877.65	10.0002	0.09
Min	10825.39	10.0002	0.09
Max	10952.52	10.0002	0.09
Avg	10902.622	10.0002	0.09
Std Dev	51.5377723	0	0

File I/O Test (QEMU Results with 3GB)

Test #	Read Throughput MiB/s	Write Throughput MibB/s	Total Time (s)
1	92.03	61.35	30.0031
2	85.69	57.13	30.0037
3	84.21	56.14	30.0027
4	83.72	55.81	30.0030
5	83.72	55.81	30.0030
Min	83.72	55.81	30.0027
Max	92.03	61.35	30.0037
Avg	85.874	57.248	30.0031
Std Dev	3.53477439	2.35603905	0.00036742

File I/O Test (QEMU Results with 2GB)

Test #	Read Throughput MiB/s	Write Throughput MibB/s	Total Time (s)
1	89.46	59.64	30.0034
2	85.02	56.68	30.0025
3	92.03	61.35	30.0031
4	104.52	69.68	30.0026
5	96.43	64.29	30.0026
Min	85.02	56.68	30.0025
Max	104.52	69.68	30.0034
Avg	93.492	62.328	30.00284
Std Dev	7.42323851	4.94932016	0.00039115

File I/O Test (QEMU Results with 1GB)

Test #	Read Throughput MiB/s	Write Throughput MibB/s	Total Time (s)
1	105.76	70.50	30.0031
2	146.33	97.55	30.0025
3	146.40	97.60	30.0030
4	146.62	97.74	30.0028
5	141.27	94.18	30.0025
Min	105.76	70.5	30.0025
Max	146.62	97.74	30.0031
Avg	137.276	91.514	30.00278
Std Dev	17.7605104	11.8419965	0.00027749

File I/O Test (Docker Results with 3GB)

Test #	Read Throughput MiB/s	Write Throughput MiB/s	Total Time (s)
1	58.21	38.81	30.0051
2	65.12	43.42	30.0056
3	62.78	41.86	30.0057
4	63.28	42.19	30.0067
5	66.88	44.59	30.0058
Min	58.21	38.81	30.0051
Max	66.88	44.59	30.0067
Avg	63.254	42.174	30.00578
Std Dev	3.2518118	2.168232	0.00058052

File I/O Test (Docker Results with 2GB)

Test #	Read Throughput MiB/s	Write Throughput MiB/s	Total Time (s)
1	68.36	45.57	30.0053
2	69.45	46.30	30.0060
3	76.20	50.80	30.0059
4	72.95	48.63	30.0059
5	77.42	51.61	30.0059
Min	68.36	45.57	30.0053
Max	77.42	51.61	30.006
Avg	72.876	48.582	30.0058
Std Dev	3.99481289	2.66318794	0.00028284

File I/O Test (Docker Results with 1GB)

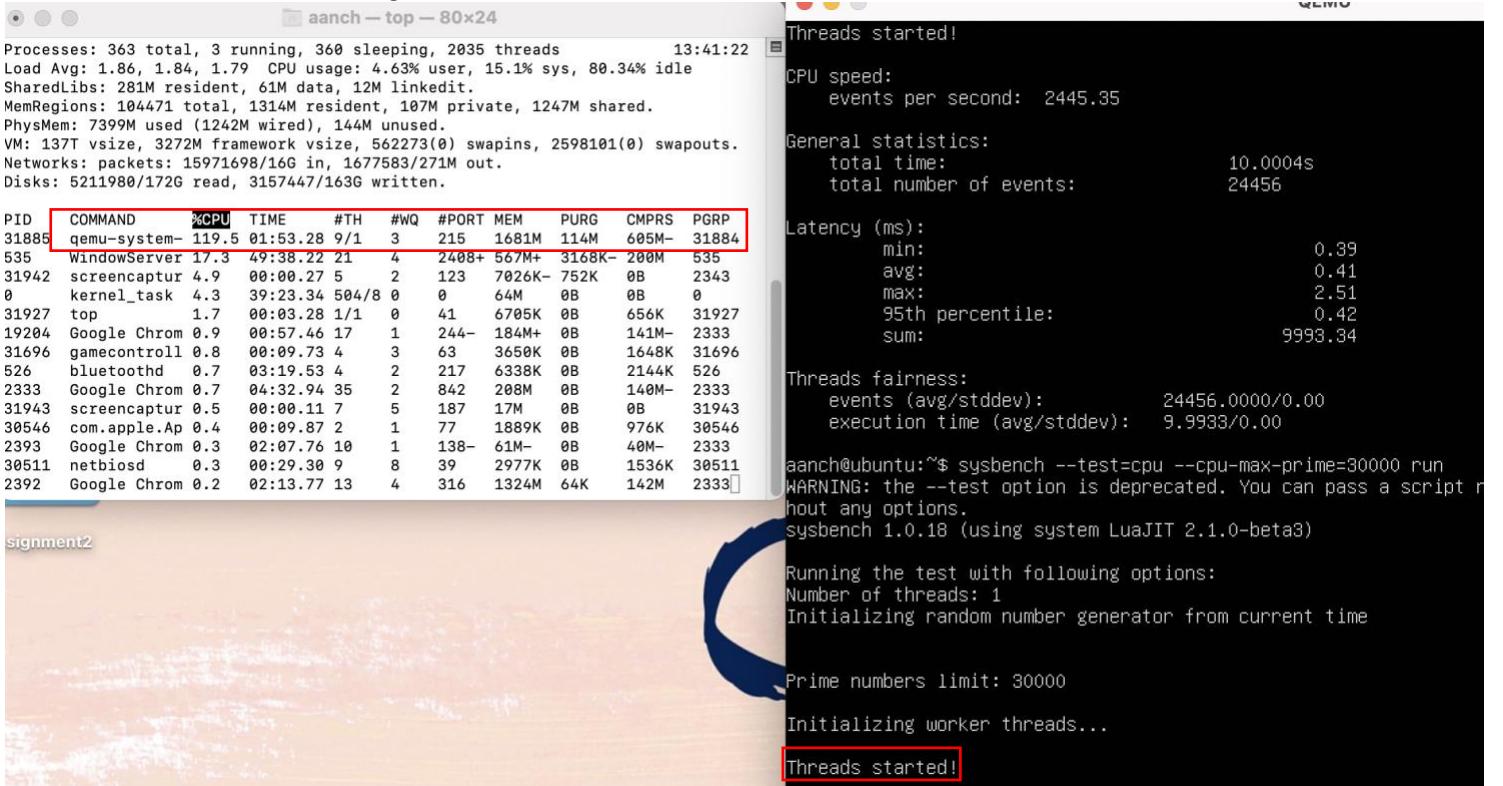
Test #	Read Throughput MiB/s	Write Throughput MiB/s	Total Time (s)
1	78.66	52.44	30.0058
2	95.23	63.49	30.0050
3	88.30	58.86	30.0050
4	96.01	64.01	30.0053
5	96.14	64.09	30.0049
Min	78.66	52.44	30.0049
Max	96.14	64.09	30.0058
Avg	90.868	60.578	30.0052
Std Dev	7.56457335	5.04408267	0.00036742

Performance Data

Ran the command “top” at the Host System terminal while running the experiments in QEMU and Docker. The top command provides the CPU Performance data.

As highlighted in the images below, the CPU Utilization % and Memory utilized increases as the experiments start running. The memory used however seems to depend primarily on the type of virtualization.

CPU Max Prime Test with QEMU



File I/O Test with QEMU

```
Processes: 344 total, 3 running, 341 sleeping, 1716 threads 17:55:4
Load Avg: 1.70, 1.51, 1.46 CPU usage: 7.94% user, 13.44% sys, 78.60% idle
SharedLibs: 167M resident, 41M data, 9168K linkedit.
MemRegions: 87328 total, 1512M resident, 135M private, 813M shared.
PhysMem: 6890M used (1210M wired), 652M unused.
/M: 130T vszie, 3272M framework vszie, 593998(0) swapins, 2615797(0) swapouts.
Networks: packets: 17883909/16G in, 2021336/319M out.
Disks: 8199898/230G read, 7311561/279G written.
```

The figure shows two terminal windows side-by-side. The left window displays the output of the 'top' command, showing system statistics and a detailed list of processes. The right window shows the output of a file I/O test. A red box highlights the 'Threads started!' message in both windows.

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP
33196	qemu-system-	124.3	23:08.66	11/2	2	220	1394M+	114M	351M	33195
535	WindowServer	17.3	72:36.00	21	5	2525+	660M+	8000K-	235M	535
0	kernel_task	6.4	52:11.13	506/8	0	0	64M	0B	0B	0
33470	screencaptur	5.3	00:00.24	5	2	124+	6946K+	752K	0B	2343
33469	top	1.7	00:00.29	1/1	0	28	4161K	0B	0B	33469
32775	gamecontroll	1.0	00:22.57	4	3	63	3810K	0B	1552K	32775
772	mds_stores	1.0	08:48.15	4	2	93	57M+	16K	44M-	772
526	bluetoothd	0.9	04:54.42	5	3	217	6514K	0B	1904K	526
30546	com.apple.Ap	0.6	00:32.82	2	1	77	1889K	0B	976K	30546
33471	screencaptur	0.6	00:00.16	7	5	184	16M	0B	0B	33471
32432	netbiosd	0.5	00:45.13	7	6	33	3153K	0B	1296K	32432
33462	mdworker_sha	0.4	00:00.07	3	1	48	2929K+	0B	0B	33462
2392	Google Chrom	0.2	02:34.65	11	2	169	151M	64K	97M	2333
2332	Terminal	0.1	06:56.62	9	3	452	149M+	48K-	70M	2332

CPU Max Prime Test with Docker

Processes: 329 total, 3 running, 326 sleeping, 1778 threads
 Load Avg: 1.32, 1.31, 1.42 CPU usage: 1.57% user, 14.64% sys, 83.77% idle
 SharedLibs: 336M resident, 82M data, 19M linkedit.
 MemRegions: 85196 total, 1527M resident, 156M private, 1329M shared.
 PhysMem: 7247M used (1118M wired), 294M unused.
 VM: 124T vsize, 3272M framework vsize, 593443(0) swapins, 2615797(0) swapouts.
 Networks: packets: 16845338/16G in, 1871884/299M out.
 Disks: 5693032/185G read, 3367587/169G written.

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP	PPID	STATE
32508	qemu-system-	102.3	05:54.43	8/1	0	23	2795M	0B	726M	32471	32506	running
535	WindowServer	9.4	00:21.48	21	5	2580	678M-	40M+	222M	535	1	sleeping
0	kernel_task	4.3	46:29.00	586/8	0	0	7792K-	0B	0B	0	0	running
32876	screencaptur	3.9	00:00.60	3	1	120	6450K	752K	0B	2343	2343	sleeping
32858	top	2.4	00:03.93	1/1	0	35	4785K	0B	656K	32858	32590	running
526	bluetoothd	0.7	04:13.02	4	2	217	6450K	0B	2096K	526	1	sleeping
32775	gamecontroll	0.6	00:02.68	3	2	63	3634K	0B	1488K	32775	1	sleeping
2332	Terminal	0.6	05:36.51	9	3	428	124M	21M	31M	2332	1	sleeping
32874	screencaptur	0.4	00:00.25	4	2	177	18M	0B	0B	32874	1	sleeping
30546	com.apple.Ap	0.3	00:21.99	2	1	77	1889K	0B	976K	30546	1	sleeping
32506	com.docker.d	0.3	00:07.64	15	0	36	17M	0B	3920K	32471	32502	sleeping
32432	netbiosd	0.3	00:14.86	7	6	31	3137K	0B	1360K	32432	1	sleeping
529	corebrightne	0.2	01:54.80	4	3	132	6466K	0B	3984K	529	1	sleeping
31883	PerfPowerSer	0.2	00:31.18	4	2	642-	12M-	128K	2912K	31883	1	sleeping
29884	routined	0.2	00:23.39	4	3	177	22M	0B	17M-	29884	1	sleeping
511	locationd	0.2	03:39.96	6	4	190	11M	128K	5536K	511	1	sleeping
796	mysqld	0.1	01:16.61	39	0	71	367M	0B	1457M	796	1	sleeping
488	logd	0.1	01:11.17	4	3	1030	26M	0B	19M-	480	1	sleeping
856	UserEventAge	0.1	00:29.08	3	2	956	7713K	0B	4176K	856	1	sleeping
530	AirPlayXPCHe	0.1	00:22.40	9	5	293	5586K	0B	3552K	530	1	sleeping
32505	vpnkit-bridg	0.0	00:05.14	11	8	31	12M	0B	3184K	32471	32502	sleeping
641	TouchBarServ	0.0	04:13.35	3	1	368	19M	3712K	9632K	641	1	sleeping
30794	com.apple.ph	0.0	00:10.48	2	1	78	9698K	0B	10M	30794	1	sleeping
2392	Google Chrom	0.0	02:26.16	10	1	168	152M	64K	66M	2333	2333	sleeping
32503	com.docker.v	0.0	00:07.11	5	0	18	84M	0B	44M-	32471	32502	sleeping
1077	sharingd	0.0	01:59.95	6	2	320	28M	0B	25M	1077	1	sleeping
19047	Safari	0.0	06:50.62	5	1	1211-	204M-	0B	129M	19047	1	sleeping
1	launchd	0.0	03:30.56	2	1	4891	31M	0B	14M	1	0	sleeping
475	powerd	0.0	00:33.85	4	3	136	3777K	0B	1120K	475	1	sleeping
783	systemstats	0.0	00:00.79	4	2	82+	2337K+	0B	1232K-	471	471	sleeping
559	coreaudiod	0.0	01:13.98	6	2	422	22M	0B	8496K	559	1	sleeping
3287	storagekitd	0.0	00:14.10	7	3	79	11M	0B	9856K	3287	1	sleeping
2339	Microsoft Wo	0.0	17:00.98	22	9	1517	505M	35M	213M	2339	1	sleeping
31622	Microsoft Ex	0.0	00:57.54	25	7	564	185M	64K	147M	31622	1	sleeping
527	notifyd	0.0	02:12.56	2	1	503	3681K	0B	1360K	527	1	sleeping
499	thermalmonit	0.0	00:05.79	2	1	44	2241K	0B	1136K	499	1	sleeping
32471	com.docker.b	0.0	00:01.04	15	0	49	26M	0B	14M	32471	32470	sleeping
880	lockoutagent	0.0	00:06.50	3	2	84	3730K	0B	1808K	880	1	sleeping
2340	Messages	0.0	04:34.01	5	1	468-	102M-	0B	92M	2340	1	sleeping
2333	Google Chrom	0.0	05:08.44	33	1	774-	207M-	0B	111M	2333	1	sleeping
885	CalendarAgen	0.0	01:01.16	3	1	219-	31M-	0B	23M-	885	1	sleeping
993	knowledge-ag	0.0	00:39.24	2	1	359-	11M-	1280K	5376K	993	1	sleeping
32785	trustd	0.0	00:00.29	2	1	50-	4610K-	112K	2416K	32785	1	sleeping
463	syslogd	0.0	00:15.10	4	3	81	2113K	0B	896K	463	1	sleeping
2337	Notes	0.0	00:28.71	5	1	441-	93M-	0B	74M	2337	1	sleeping
467	fseventsds	0.0	04:18.98	10	1	252	8706K	0B	4400K	467	1	sleeping
2441	cloudpайд	0.0	00:11.71	2	1	128	5362K	0B	4256K	2441	1	sleeping
498	mds	0.0	10:14.45	6	3	361	57M	0B	45M	490	1	sleeping
1088	useractivity	0.0	00:12.10	2	1	133	4962K	0B	1872K	1088	1	sleeping
471	systemstats	0.0	00:44.14	5	4	124	5633K	0B	944K	471	1	sleeping

16:15:36

```
Initializing worker threads...
Threads started!
CPU speed:
events per second: 10825.3
General statistics:
total time:
total number of events:
Latency (ms):
min:
avg:
max:
95th percentile:
sum:
Threads fairness:
events (avg/stddev):
execution time (avg/stddev)
Test #: 5
WARNING: the --test option is
t name or path on the command
sysbench 1.0.18 (using system
Running the test with followin
Number of threads: 1
Initializing random number gen

Prime numbers limit: 10000
Initializing worker threads...
Threads started!
```

FILE I/O Test with Docker

```

0.00 - 0.01 test
Commands: prepare run cleanup help version
See 'sysbench --test=<name> help' for a list of options for each test.
[root@5e986695b075 ~]# sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct run
Unknown command: direct.
Usage:
  sysbench [general-options]... --test=<test-name> [test-options]... command

General options:
  --num-threads=N          number of threads to use [1]
  --max-requests=N         limit for total number of requests [10000]
  --max-time=N              limit for total execution time in seconds [0]
  --forced-shutdown=STRING amount of time to wait after --max-time before forcing shutdown [off]
  --thread-stack-size=SIZE size of stack per thread [32K]
  --init-rng=[on|off]       initialize random number generator [off]
  --test=STRING             test to run
  --debug=[on|off]          print more debugging info [off]
  --validate=[on|off]        perform validation checks where possible [off]
  --help=[on|off]            print help and exit
  --version=[on|off]         print version and exit

Compiled-in tests:
  fileio - File I/O test
  cpu - CPU performance test
  memory - Memory functions speed test
  threads - Threads subsystem performance test
  mutex - Mutex performance test
  olt - OLTP test

Commands: prepare run cleanup help version
See 'sysbench --test=<name> help' for a list of options for each test.
[root@5e986695b075 ~]# sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct prepare
sysbench 0.4.12: multi-threaded system evaluation benchmark

128 files, 24576K each, 3072M total
Creating files for the test...
[root@5e986695b075 ~]# sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw --max-time=30 --max-requests=0 --file-extra-flags=direct run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Extra file open flags: 16384
128 files, 24M each
3Gb total file size
Block size 16kb
Number of random requests for random IO: 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
Threads started!

```

PID	COMMAND	NCPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	GPR	PPID	STATE
32830	qemu-system-i686	66.5	01:22.90	13/1	0	28	633M	0B	433M	31989	32827	running
355	WindowsServer	16.9	01:25.34	22	0	23	720M-	7680K+	249M-	535	1	sleeping
8	kernel_task	7.1	04:28.32	584/8	0	0	63M	0B	0B	0	0	running
32138	screencaptur	6.9	08:08.15	5	2	123	6786K+	752K	0B	2343	2343	sleeping
31927	top	2.4	08:34.25	1/1	0	79	7281K	0B	2096K	31927	31922	running
31496	gamecontrolr	1.3	08:14.49	4	3	63	3746K	0B	1848K	31196	1	sleeping
526	bluetoothd	1.2	03:26.45	5	3	217	6354K	0B	2384K	526	1	sleeping
2332	Terminal	1.2	02:42.82	11	5	447+	131M	40M+	43M	2332	1	sleeping
30546	com.apple.Ap	0.7	08:12.13	2	1	77	1889K	0B	976K	30546	1	sleeping
32137	screencaptur	0.7	08:08.13	18	8	198	18M	48K	0B	32137	1	sleeping
529	corebrightne	0.6	01:32.36	7	6	139	6626K	0B	4896K	529	1	sleeping
31883	PerfPowerSer	0.2	08:09.87	6	4	633	11M	0B	3664K-	31883	1	sleeping
2393	Google Chrom	0.2	02:11.63	18	1	134	62M	0B	42M	2333	2333	sleeping
480	logd	0.2	01:31.36	4	3	1134	25M-	0B	22M-	480	1	sleeping
559	coreaudiod	0.1	01:02.59	8	1	436-	23M-	0B	12M	559	1	sleeping
32118	mdworker_sha	0.1	08:08.13	3	1	46	3297K+	0B	1200K	32118	1	sleeping
498	mds	0.1	08:08.13	6	3	366	59M	0B	49M	498	1	sleeping
2333	Google Chrom	0.1	04:42.62	34	1	843	289M	0B	152M	2333	1	sleeping
796	mysqld	0.1	01:05.15	39	0	71	367M	0B	1457M	796	1	sleeping
30511	netbiosd	0.1	00:32.51	7	6	37	3025K	0B	1568K	30511	1	sleeping
511	location	0.1	01:05.62	5	3	187	9842K	0B	5128K-	511	1	sleeping
2392	Google Chrom	0.1	02:16.99	11	2	302	1317M	0B	155M	2333	2333	sleeping
30794	com.apple.ph	0.05	08:05.63	2	1	77	9698K	0B	18M	30794	1	sleeping
772	mds_stores	0.05	08:35.01	5	3	91	58M+	0B	46M-	772	1	sleeping
982	Notification	0.01	08:21.60	4	2	585-	36M-	0B	26M-	982	1	sleeping
514	dasd	0.1	00:31.63	3	2	98	7026K	0B	2912K	514	1	sleeping
641	TouchBarServ	0.0	03:26.69	5	3	367	22M-	18M	641	1	sleeping	
475	powerd	0.0	00:24.03	3	2	138	3857K	0B	1728K	475	1	sleeping
856	UserEventAge	0.0	00:23.59	3	2	952-	7665K	0B	3888K-	856	1	sleeping
2337	Notes	0.0	00:18.18	7	3	447-	88M-	64K	68M-	2337	1	sleeping
19206	Google Chrom	0.0	00:09.87	8	1	111-	13M-	0B	11M-	2333	2333	sleeping
553	airportd	0.0	04:32.52	9	7	414	15M	0B	8488K	553	1	sleeping
463	syslogd	0.0	00:18.82	4	3	88+	2979K+	0B	784K	463	1	stuck
1077	sharingd	0.0	01:49.40	6	2	323	28M	0B	23M	1077	1	sleeping
19205	Google Chrom	0.0	00:37.05	16	1	225	180M	0B	74M	2333	2333	sleeping
500	contextstore	0.0	00:22.79	3	2	128	7922K	0B	3152K	500	1	sleeping
32116	Google Chrom	0.0	00:05.11	16	1	179	82M+	0B	22M	2333	2333	sleeping
31788	Google Chrom	0.0	00:05.41	15	1	278	72M	0B	52M	2333	2333	sleeping
878	CommCenter	0.0	00:02.87	6	2	129	7266K	0B	4168K	878	1	sleeping
527	notifyd	0.0	00:44.67	2	1	536	3649K	0B	1288K	527	1	sleeping
530	AirPlayXPCh	0.0	00:16.41	5	1	194	5522K	0B	2944K	530	1	sleeping
1	launchd	0.0	02:54.18	3	2	4895	38M	0B	15M-	1	0	sleeping
32024	com.docker.v	0.0	00:00.32	5	0	18	38M	0B	22M	31989	32823	sleeping
466	uninstallld	0.0	00:00.95	3	2	21+	1633K+	0B	688K-	466	1	sleeping
3287	storagekitd	0.0	00:12.45	7	3	78	11M	0B	18M	3287	1	sleeping
499	thermalmonitor	0.0	00:04.88	2	1	45	2273K	0B	1288K	499	1	sleeping
31622	Microsoft Ex	0.0	00:53.43	25	7	560	185M	0B	156M	31622	1	sleeping
2339	Microsoft Wo	0.0	14:30.32	21	8	1352	469M	0B	365M	2339	1	sleeping
880	lockoutagent	0.0	00:05.46	4	3	83	3762K	0B	2888K	880	1	sleeping
638	mDNSResponde	0.0	00:41.02	3	1	177	5906K	0B	2800K	638	1	sleeping
473	configd	0.0	00:39.66	8	2	468	7314K	0B	3648K	473	1	sleeping
1871	Spotlight	0.0	00:30.55	5	2	476-	85M-	16K	51M-	1871	1	sleeping
29697	com.apple.ge	0.0	00:03.48	3	1	183-	12M-	0B	9824K	29697	1	sleeping

Conclusion:

Ideally, the OS Virtualization with Docker should be more performant than the System Virtualization implemented through QEMU. This is because Docker containers are executed via a Docker Engine (instead of a Hypervisor) and they are not fully isolated. This makes docker lightweight with the ability to deploy faster. However, in the experiments performed above, the two virtualization techniques provided similar results when run with similar environment constraints. This may be because of internal system configurations that may have played a role.