

CSEN-241 Cloud Computing

HW1: System Vs OS Virtualization Report

Submitted by: Rishabh Agrawal
(W1651177)

Host System Configuration:

1	Chip	Apple M1 (Apple Silicon)
2	CPU	8 (4 performance and 4 efficiency)
3	Memory	8 GB
4	Free disk space	90 GB
5	Operating System	Mac OS Sonoma

QEMU Installation and creation of QEMU image

In this section, we'll guide you through the process of QEMU installation and the creation of a QEMU image that will be utilized for all our experiments.

1. Downloading Ubuntu Guest Virtual Machine

Initially, we will obtain the necessary ISO image tailored to our system's architecture:



```
Last login: Wed Jan 31 03:25:17 on ttys000
[(base) sumit@Rishabh ~ % arch
arm64
(base) sumit@Rishabh ~ %
```

Therefore, we will use Arm64v8 architecture from below link:

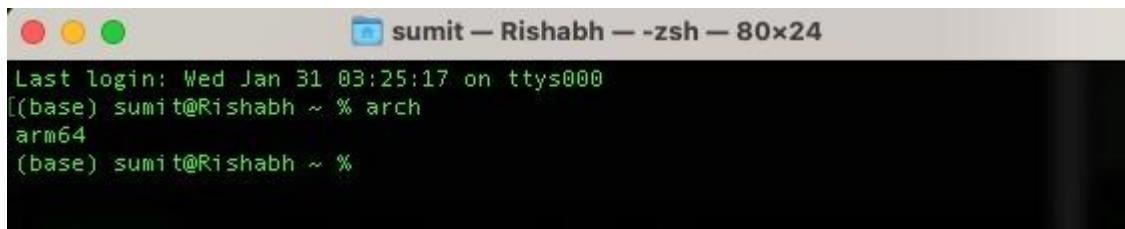
ARM (Apple Silicon) - [Ubuntu 20.04 Server for ARM](#)

Create one folder to store the ubuntu image and a ISO file

2. Install QEMU

To install qemu use below homebrew command:

```
brew install qemu
```



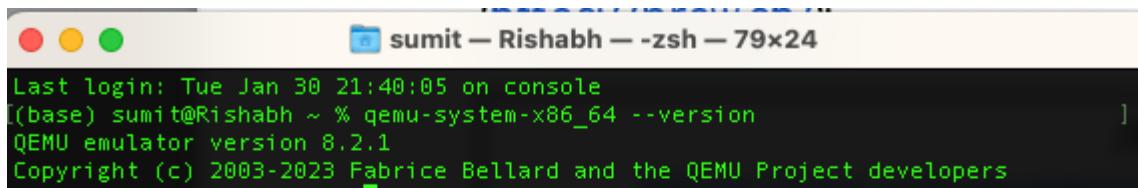
```
Last login: Wed Jan 31 03:25:17 on ttys000
[(base) sumit@Rishabh ~ % arch
arm64
(base) sumit@Rishabh ~ %
```

3. Check QEMU version

We have to make sure that the QEMU version should be 8.2.0+.

```
qemu-system-x86_64 --version
```

We have a QEMU 8.2.1 version.



```
Last login: Tue Jan 30 21:40:05 on console
[(base) sumit@Rishabh ~ % qemu-system-x86_64 --version
QEMU emulator version 8.2.1
Copyright (c) 2003-2023 Fabrice Bellard and the QEMU Project developers
```

4. Create QEMU image

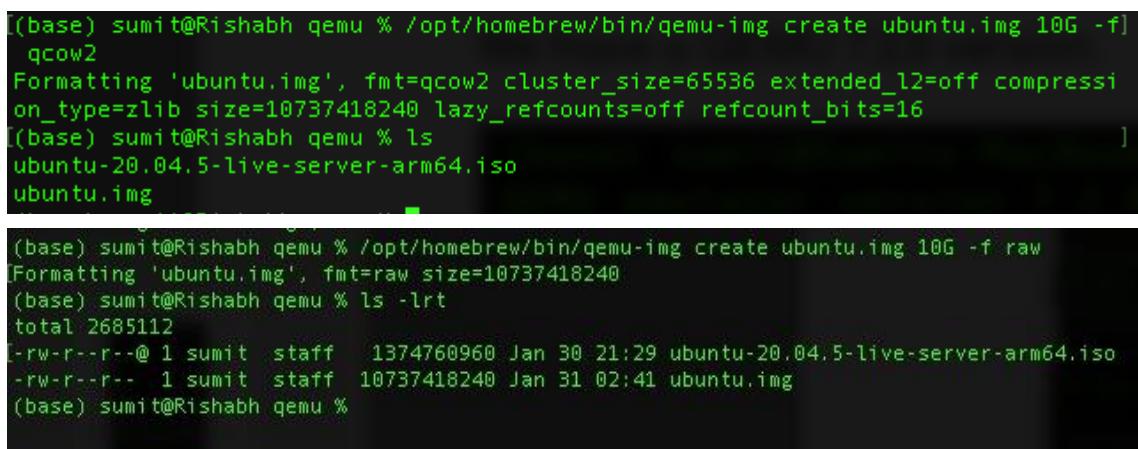
Then we'll make a QEMU image with a disk image as qcow2 and raw . This image will be created in the same folder where the Ubuntu guest virtual machine ISO file was copied. In the terminal, type the following command:

For disk image qcow2:

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

For disk image raw:

```
qemu-img create ubuntu.img 10G -f raw
```



```
[(base) sumit@Rishabh qemu % /opt/homebrew/bin/qemu-img create ubuntu.img 10G -f]
qcow2
Formatting 'ubuntu.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=10737418240 lazy_refcounts=off refcount_bits=16
[(base) sumit@Rishabh qemu % ls
ubuntu-20.04.5-live-server-arm64.iso
ubuntu.img
[...]
(base) sumit@Rishabh qemu % /opt/homebrew/bin/qemu-img create ubuntu.img 10G -f raw
Formatting 'ubuntu.img', fmt=raw size=10737418240
(base) sumit@Rishabh qemu % ls -lrt
total 2685112
-rw-r--r--@ 1 sumit  staff  1374760960 Jan 30 21:29 ubuntu-20.04.5-live-server-arm64.iso
-rw-r--r--  1 sumit  staff   10737418240 Jan 31 02:41 ubuntu.img
(base) sumit@Rishabh qemu %
```

5. Install Ubuntu VM

Run the below command to install Ubuntu VM:

For disk image qcow2:

```
/opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 2 \
-drive
file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=pflash,fo
rmat=raw,readonly=on \
-drive if=none,file=ubuntu.img,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 ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

For disk image raw:

```
/opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 2 \
-drive
file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=pflash,fo
rmat=raw,readonly=on \
-drive if=none,file=ubuntu.img,format=raw,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 ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

```
(base) sumit@Rishabh: ~ % /opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 2 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=pflash,format=r
aw,readonly=on \
-drive if=none,file=ubuntu.img,format=raw,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 ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

While installing it, you must provide a server name, username, and password which will be used to login. Then Select reboot option.



We can try logging in with the following command (we must remove the cdrom argument). It will prompt you for your username and password. Disk image name will change i.e either qcow2 or raw

```
/opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 2 \
-drive
file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=pflash,fo
rmat=raw,readonly=on \
-drive if=none,file=ubuntu.img,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 \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```



```

* Management:      https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Wed Jan 31 06:02:41 UTC 2024

System load:          0.08
Usage of /:           49.0% of 7.50GB
Memory usage:         10%
Swap usage:           0%
Processes:            104
Users logged in:     0
IPv4 address for eth0: 10.0.2.15
IPv6 address for eth0: fec0::5054:ff:fe12:3456

68 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

rishabh@rishabh:~$ 

```

Below is the explanation of arguments used in the command.

-accel	This option enables the use of a hardware accelerator to improve performance. The available options include Kvm, xen, hax, hvf, nvmm, whpx, or tcg, which vary based on the target architecture.
-m	This flag specifies the amount of memory to allocate to the virtual machine (VM). In this case, "2G" indicates that 2GB of RAM memory is being allocated.
-smp	This parameter sets the number of CPU cores assigned to the VM. The value "2" means that two cores are being allocated.
-cdrom	This argument is used to specify a CD-ROM image file. It cannot be used simultaneously with `'-hdc'`. To use the host's CD-ROM, you can specify `/dev/cdrom` as the filename.
-device	This is used to add a device driver, with properties specified in the form of `prop=value`, allowing for the configuration of the driver's properties

When running for different configurations, we would update the -m and -smp values.

Docker Installation, Building Docker Image, and Enabling Docker Containers

1. Installation

Next, we'll go over the installation steps for Docker, a tool used for container creation and management, allowing us to run and oversee various images.

We'll utilize the <https://docs.docker.com/desktop/mac/apple-silicon/> link to install Docker Desktop, which comes with both the Docker CLI and Docker Engine by default. To ensure a successful installation, please execute the following commands in your terminal:

```
docker pull hello-world  
docker images  
docker run hello-world
```

2. Basic docker image creation and running docker image:

We will make a docker image from dockerfile to run experiments. Below is basic step to create a docker image:

Terminal 1: docker images sudo docker run -it --entrypoint "bin/bash" arm64v8/ubuntu:20.04 apt update apt install <package>	Terminal 2: docker ps docker commit <container_id> <image_name> docker images docker history <image_name>
--	---

Following are descriptions of the docker commands:

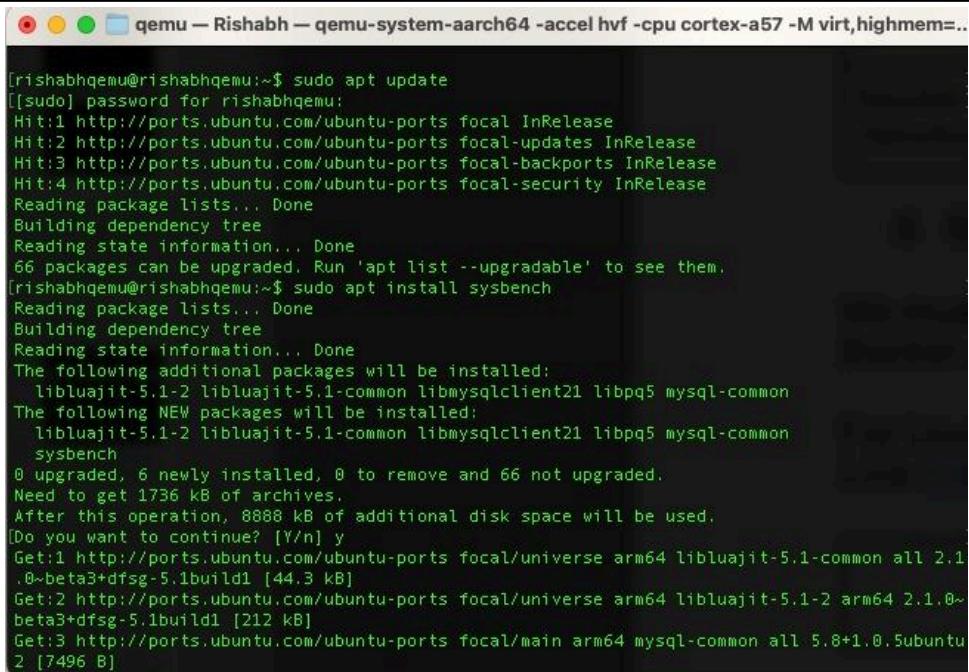
docker pull image-name:tag	To pull a docker image
docker images	To list out all docker images
docker run image-name	To run a docker image
docker run [container] image [command]	To run a container
docker ps -a	To list out all containers; running as well as exited.
docker commit <container_id> <image_name>	To commit a container's file changes or settings into a new image
docker history <image_name>	To show history of an image
docker inspect container id	To check all details about containers
docker system prune -a	To delete all docker images
Docker exec -it containerID /bin/bash	To go inside the running container

3. Sysbench Installation.

We must make sure that we have a same version of sysbench in QEMU and in Docker. To install use the below command

For Docker, we are using the sysbench library package:
Link: <https://github.com/akopytov/sysbench>

```
apt update;  
apt install sysbench;
```



```
[rishabhqemu@rishabhqemu:~$ sudo apt update  
[sudo] password for rishabhqemu:  
Hit:1 http://ports.ubuntu.com/ubuntu-ports focal InRelease  
Hit:2 http://ports.ubuntu.com/ubuntu-ports focal-updates InRelease  
Hit:3 http://ports.ubuntu.com/ubuntu-ports focal-backports InRelease  
Hit:4 http://ports.ubuntu.com/ubuntu-ports focal-security InRelease  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
66 packages can be upgraded. Run 'apt list --upgradable' to see them.  
[rishabhqemu@rishabhqemu:~$ sudo apt install sysbench  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
libluajit-5.1-2 libluajit-5.1-common libmysqlclient21 libpq5 mysql-common  
The following NEW packages will be installed:  
libluajit-5.1-2 libluajit-5.1-common libmysqlclient21 libpq5 mysql-common  
sysbench  
0 upgraded, 6 newly installed, 0 to remove and 66 not upgraded.  
Need to get 1736 kB of archives.  
After this operation, 8888 kB of additional disk space will be used.  
[Do you want to continue? [Y/n] y  
Get:1 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libluajit-5.1-common all 2.1  
.0~beta3+dfsg-5.1build1 [44.3 kB]  
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libluajit-5.1-2 arm64 2.1.0~  
beta3+dfsg-5.1build1 [212 kB]  
Get:3 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 mysql-common all 5.8+1.0.5ubuntu  
2 [7496 B]
```

Test Conditions: To conduct the experiment with the VM, we will consider four test conditions showed in the table below. To ensure consistency between results, the same test conditions will be used for Docker as well as for QEMU

Cores	Memory allocation
2	2 GB
3	3 GB
3	2 GB
6	3 GB

Proof of experiment - Reports and Findings

In this section, we will look at test cases that are specifically related to sysbench CPU and File I/O commands. We will also test different QEMU VM configurations to see if we get different results by changing the VM configurations.

1. CPU Testing

We will use the following three test cases to evaluate CPU performance between QEMU and Docker. For our testing, we will use the sysbench command and the test cases listed below:

```
sysbench cpu --threads={Value} --cpu-max-prime={Value} --time={Value} run  
1. max-prime = 1000 and time = 30 seconds  
2. max-prime = 10,000 and time = 30 seconds  
3. max-prime = 100,000 and time = 30 seconds
```

2. FILEIO Testing

For File I/O testing we will be using random read & sequential write). Also, for our testing purposes, we will use the following sysbench commands, and changing the values for different configuration.

```
sysbench --test=fileio --file-total-size={Value} --file-test-mode={Value} prepare  
sysbench --test=fileio --file-total-size={Value} --file-test-mode={Value} run  
sysbench --test=fileio --file-total-size={Value} cleanup
```

Here test mode value will be **seqwr** for sequential write & **rndrd** for random read

3. Memory Testing

For memory testing we will be using random & sequential access. Also, for our testing purposes, we will use the following sysbench commands, and changing the values for different configuration.

```
sysbench --test=memory --memory-block-size={Value} --memory-total-size={Value}  
--memory-access-mode={Value} run
```

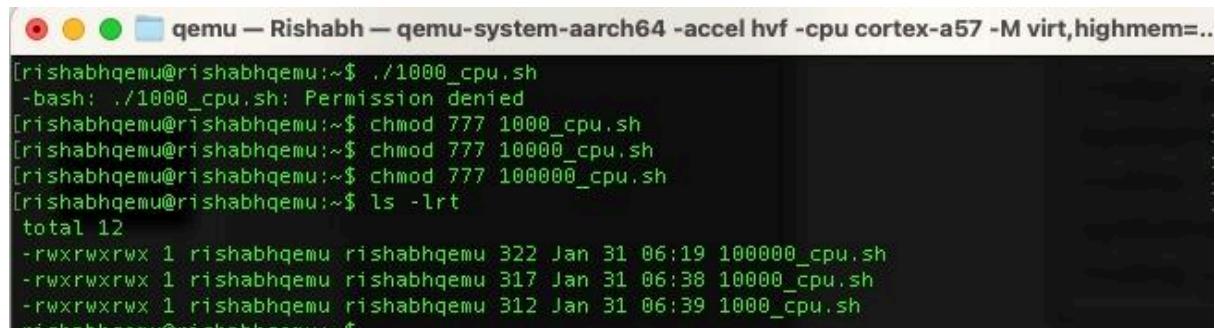
Here memory access mode value will be **seq** for sequential & **rnd** for random access

4. Results

1. Configuration 1: 2 GB RAM with 2 Cores

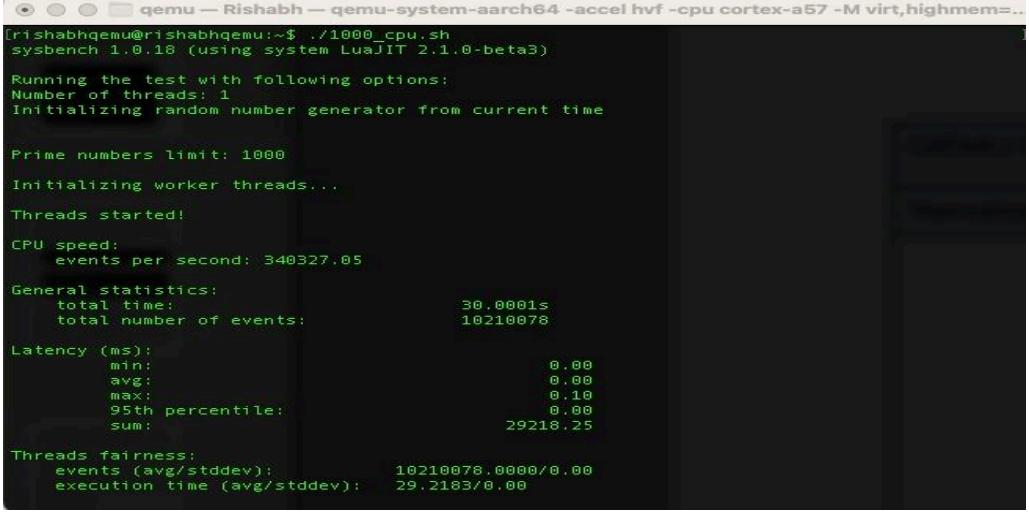
1.1 CPU Testing

To conduct these experiments, I wrote three bash scripts, each with a different max-prime value and five iterations:



```
[rishabhqemu@rishabhqemu:~$ ./1000_cpu.sh
-bash: ./1000_cpu.sh: Permission denied
[rishabhqemu@rishabhqemu:~$ chmod 777 1000_cpu.sh
[rishabhqemu@rishabhqemu:~$ chmod 777 10000_cpu.sh
[rishabhqemu@rishabhqemu:~$ chmod 777 100000_cpu.sh
[rishabhqemu@rishabhqemu:~$ ls -lrt
total 12
-rwxrwxrwx 1 rishabhqemu rishabhqemu 322 Jan 31 06:19 100000_cpu.sh
-rwxrwxrwx 1 rishabhqemu rishabhqemu 317 Jan 31 06:38 10000_cpu.sh
-rwxrwxrwx 1 rishabhqemu rishabhqemu 312 Jan 31 06:39 1000_cpu.sh
-rw-r--r-- 1 rishabhqemu rishabhqemu 10210078 Jan 31 06:39 sysbench.lua
```

QEMU Results, Configuration: 2 GB 2 cores for max-prime = 1000 and time = 30 seconds, disk image = qcow2

Iteration	Screenshot
1	 <pre>[rishabhqemu@rishabhqemu:~\$./1000_cpu.sh 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: 1000 Initializing worker threads... Threads started! CPU speed: events per second: 340327.05 General statistics: total time: 30.00015 total number of events: 10210078 Latency (ms): min: 0.00 avg: 0.00 max: 0.10 95th percentile: 0.00 sum: 29218.25 Threads fairness: events (avg/stddev): 10210078.0000/0.00 execution time (avg/stddev): 29.2183/0.00</pre>

```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=..

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: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 343245.09

General statistics:
total time: 30.0001s
total number of events: 10297716

Latency (ms):
min: 0.00
avg: 0.00
max: 0.13
95th percentile: 0.00
sum: 29214.58

Threads fairness:
events (avg/stddev): 10297716.0000/0.00
execution time (avg/stddev): 29.2146/0.00

```

2


```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=..

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: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 340526.17

General statistics:
total time: 30.0001s
total number of events: 10216136

Latency (ms):
min: 0.00
avg: 0.00
max: 0.09
95th percentile: 0.00
sum: 29217.53

Threads fairness:
events (avg/stddev): 10216136.0000/0.00
execution time (avg/stddev): 29.2175/0.00

```

3


```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=..

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: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 339094.61

General statistics:
total time: 30.0001s
total number of events: 10173112

Latency (ms):
min: 0.00
avg: 0.00
max: 0.09
95th percentile: 0.00
sum: 29218.61

Threads fairness:
events (avg/stddev): 10173112.0000/0.00
execution time (avg/stddev): 29.2186/0.00

```

4

5

```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=...
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: 1000
Initializing worker threads...
Threads started!
CPU speed:
  events per second: 338034.03

General statistics:
  total time:          30.0001s
  total number of events: 10141356

Latency (ms):
  min:                  0.00
  avg:                  0.00
  max:                  0.00
  95th percentile:      0.00
  sum:                 29221.90

Threads fairness:
  events (avg/stddev): 10141356.0000/0.00
  execution time (avg/stddev): 29.2219/0.00

```

Detailed report:

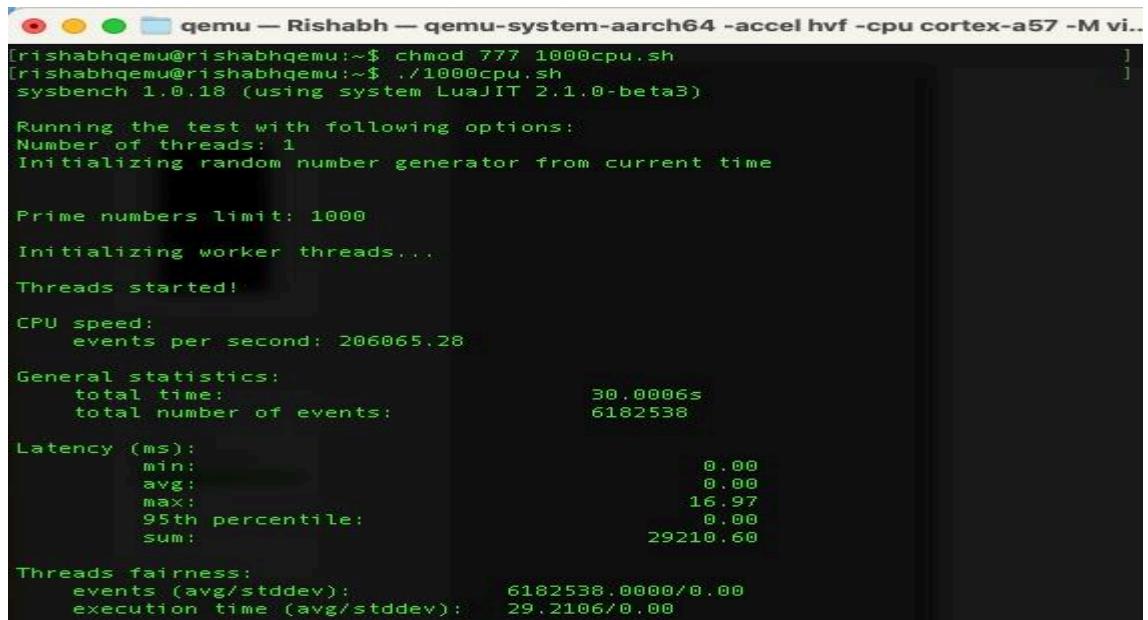
Iteration No	Events/second
1	340327.05
2	343245.09
3	340526.17
4	339094.61
5	330034.03

Observations:

Average events per second	338845.99
Minimum Number of events per second recorded	343245.09
Maximum Number of events per second recorded	330034.03

QEMU Results, Configuration: 2 GB 2 cores for max-prime = 1000 and time = 30 seconds, disk image = raw

Screenshot



```
[rishabh@qemu:~$ chmod 777 1000cpu.sh
[rishabh@qemu:~$ ./1000cpu.sh
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: 1000
Initializing worker threads...

Threads started!
CPU speed:
events per second: 206065.28

General statistics:
total time: 30.00006s
total number of events: 6182538

Latency (ms):
min: 0.00
avg: 0.00
max: 16.97
95th percentile: 0.00
sum: 29210.60

Threads fairness:
events (avg/stddev): 6182538.0000/0.00
execution time (avg/stddev): 29.2106/0.00
```

Detailed report:

Iteration No	Events/second
1	206065.28
2	201245.89
3	214102.59
4	212065.28
5	205034.03

Observations:

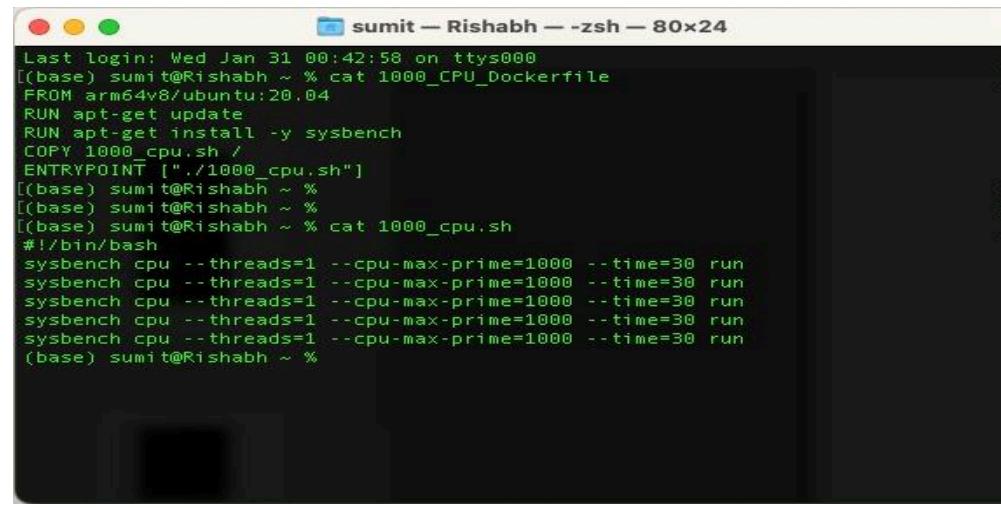
Average events per second	207702.61
Minimum Number of events per second recorded	201245.89
Maximum Number of events per second recorded	214102.59

Docker Results, Configuration: 2 GB 2 cores for max-prime = 1000 and time = 30 seconds

Setting CPU count and memory in docker desktop:



Dockerfile, sysbench command, building docker image and running it:



```
sumit@Rishabh: ~ zsh - 80x24
Last login: Wed Jan 31 00:42:58 on ttys000
[(base) sumit@Rishabh: ~ % cat 1000_CPU_Dockerfile
FROM arm64v8/ubuntu:20.04
RUN apt-get update
RUN apt-get install -y sysbench
COPY 1000_cpu.sh /
ENTRYPOINT ["/1000_cpu.sh"]
[(base) sumit@Rishabh: ~ %
[(base) sumit@Rishabh: ~ %
[(base) sumit@Rishabh: ~ % cat 1000_cpu.sh
#!/bin/bash
sysbench cpu --threads=1 --cpu-max-prime=1000 --time=30 run
(base) sumit@Rishabh: ~ %
```

```

[[base] sumit@Rishabh ~ % docker build -t 100cpudockerimage . -f 1000_CPU_Dockerfile
[*] Building 19.0s (10/10) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from 1000_CPU_Dockerfile
=> => transferring dockerfile: 171B
=> [internal] load metadata for docker.io/arm64v8/ubuntu:20.04
=> [auth] arm64v8/ubuntu:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/arm64v8/ubuntu:20.04@sha256:080169816683e6f063d3903434565624287828ecfd
=> => resolve docker.io/arm64v8/ubuntu:20.04@sha256:080169816683e6f063d3903434565624287828ecfd
=> => sha256:080169816683e6f063d3903434565624287828ecfd 2.3s
=> => sha256:fde9c12d7d3f936756d545cc36391de434bbe311fd9d60f98e496c57cf58f21 0.0s
=> => sha256:d519a3a2a796a075e4e40e5c4a1513aa8db8f8fd009662b76858f0149143b2 2.32kB / 2.32kB
=> => sha256:d519a3a2a796a075e4e40e5c4a1513aa8db8f8fd009662b76858f0149143b2 0.0s
=> => extracting sha256:d519a3a2a796a075e4e40e5c4a1513aa8db8f8fd009662b76858f0149143b28 0.6s
=> [internal] load build context
=> => transferring context: 352B
=> [2/4] RUN apt-get update
=> [3/4] RUN apt-get install -y sysbench
=> [4/4] COPY 1000_cpu.sh /
=> exporting to image
=> => exporting layers
=> => writing image sha256:bb8d0ba7f26fd3f4f0b2a804e8e627adcbf4d7576596bbd1980531b61274223d
=> => naming to docker.io/library/100cpudockerimage
[[base] sumit@Rishabh ~ % docker inspect b453c3c23cfaec3981277fc69150905adea4a78fd7fa7e761fd3b37c09808855
[*] What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
[[base] sumit@Rishabh ~ %

```

```

[[base] sumit@Rishabh ~ % docker run -it --memory="2g" --cpuset-cpus="0-1" -d 100cpudockerimage
b453c3c23cfaec3981277fc69150905adea4a78fd7fa7e761fd3b37c09808855
[[base] sumit@Rishabh ~ %

```

Docker inspect shows us the memory and cores set for running container:

```

[[base] sumit@Rishabh ~ % docker inspect b453c3c23cfaec3981277fc69150905adea4a78fd7fa7e761fd3b37c09808855
[{"Id": "b453c3c23cfaec3981277fc69150905adea4a78fd7fa7e761fd3b37c09808855",
 "Created": "2024-01-31T09:12:30.951881012Z",
 "Path": "./1000_cpu.sh",
 "Args": [],
 "State": {
     "Status": "exited",
     "Running": false,
     "Paused": false,
     "Restarting": false,
     "OOMKilled": false,
     "Dead": false,
     "Pid": 0,
     "ExitCode": 0,
     "Error": "",
     "StartedAt": "2024-01-31T09:12:31.264691887Z",
     "FinishedAt": "2024-01-31T09:15:01.355017304Z"
 },
 "Image": "sha256:bb8d0ba7f26fd3f4f0b2a804e8e627adcbf4d7576596bbd1980531b61274223d",
 "ResolveConfPath": "/var/lib/docker/containers/b453c3c23cfaec3981277fc69150905adea4a78fd7fa7e761fd3b37c09808855",
 "HostnamePath": "/var/lib/docker/containers/b453c3c23cfaec3981277fc69150905adea4a78fd7fa7e761fd3b37c09808855",
 "HostsPath": "/var/lib/docker/containers/b453c3c23cfaec3981277fc69150905adea4a78fd7fa7e761fd3b37c09808855",
 "LogPath": "/var/lib/docker/containers/b453c3c23cfaec3981277fc69150905adea4a78fd7fa7e761fd3b37c09808855",
 "Name": "/serene_wu",
 "RestartCount": 0,
 "Driver": "overlay2",
 "Platform": "linux",
 "MountLabel": "",
 "ProcessLabel": ""
},

```

```

"Links": null,
"OomScoreAdj": 0,
"PidMode": "",
"Privileged": false,
"PublishAllPorts": false,
"ReadonlyRootfs": false,
"SecurityOpt": null,
"UTSMode": "",
"UsernsMode": "",
"ShmSize": 67108864,
"Runtime": "runc",
"Isolation": "",
"CpuShares": 0,
"Memory": 2147483648,
"NanoCpus": 0,
"GroupParent": "",
"BlkioWeight": 0,
"BlkioWeightDevice": [],
"BlkioDeviceReadBps": [],
"BlkioDeviceWriteBps": [],
"BlkioDeviceReadIOps": [],
"BlkioDeviceWriteIOps": [],
"CpuPeriod": 0,
"CpuQuota": 0,
"CpuRealtimePeriod": 0,
"CpuRealtimeRuntime": 0,
"CpusetCpus": "0-1",
"CpusetMems": "",
"Devices": [],
"DeviceGroupRules": null,
"DeviceRequests": null,
"MemoryReservation": 0,

```

Iteration	Screenshot
1	<pre> [(base) sumit@Rishabh ~ % docker logs -f serene_wu 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: 1000 Initializing worker threads... Threads started! CPU speed: events per second: 204399.41 General statistics: total time: 30.00003s total number of events: 6132235 Latency (ms): min: 0.00 avg: 0.00 max: 10.20 95th percentile: 0.00 sum: 29181.50 Threads fairness: events (avg/stddev): 6132235.0000/0.00 execution time (avg/stddev): 29.1815/0.00 </pre>
2	<pre> 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: 1000 Initializing worker threads... Threads started! CPU speed: events per second: 206078.96 General statistics: total time: 30.0001s total number of events: 6182571 Latency (ms): min: 0.00 avg: 0.00 max: 3.79 95th percentile: 0.00 sum: 29210.43 Threads fairness: events (avg/stddev): 6182571.0000/0.00 execution time (avg/stddev): 29.2104/0.00 </pre>

3	<pre> 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: 1000 Initializing worker threads... Threads started! CPU speed: events per second: 205914.97 General statistics: total time: 30.0005s total number of events: 6177797 Latency (ms): min: 0.00 avg: 0.00 max: 5.88 95th percentile: 0.00 sum: 29192.11 Threads fairness: events (avg/stddev): 6177797.0000/0.00 execution time (avg/stddev): 29.1921/0.00 </pre>
4	<pre> 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: 1000 Initializing worker threads... Threads started! CPU speed: events per second: 203152.59 General statistics: total time: 30.0005s total number of events: 6094952 Latency (ms): min: 0.00 avg: 0.00 max: 13.47 95th percentile: 0.00 sum: 29186.57 Threads fairness: events (avg/stddev): 6094952.0000/0.00 execution time (avg/stddev): 29.1866/0.00 </pre>
5	<pre> 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: 1000 Initializing worker threads... Threads started! CPU speed: events per second: 204358.80 General statistics: total time: 30.0001s total number of events: 6131033 Latency (ms): min: 0.00 avg: 0.00 max: 11.32 95th percentile: 0.00 sum: 29202.50 Threads fairness: events (avg/stddev): 6131033.0000/0.00 execution time (avg/stddev): 29.2025/0.00 (base) sumit@Rishabh ~ % </pre>

Detailed report:

Iteration No	Events per second
1	204399.41
2	206078.96
3	205914.97
4	203152.59
5	204358.80

Observations:

Average events per second	204780.94
Minimum Number of events per second recorded	203152.59
Maximum Number of events per second recorded	206078.96

Conclusion: The experimentation across different configurations and environments—QEMU with qcow2 and raw disk images, and Docker—reveals distinct performance outcomes. Specifically, QEMU with a qcow2 disk image configuration showcases the highest performance, achieving an average of 338,845.99 events per second. In contrast, QEMU with a raw disk image and Docker configurations exhibit lower performance, with averages of 207,702.61 and 204,780.94 events per second, respectively. This indicates that the choice of disk image format and environment significantly impacts the computational performance, with qcow2 in a QEMU setup offering superior efficiency for the given task.

QEMU Results for 2 GB 2 cores for max-prime = 10000 and time = 30 seconds, disk image = qcow2

1

```
[rishabh@rishabh:~$ ./10000_cpu.sh
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: 11012.97

General statistics:
total time: 30.0002s
total number of events: 330398

Latency (ms):
min: 0.09
avg: 0.09
max: 0.36
95th percentile: 0.10
sum: 29967.73

Threads fairness:
events (avg/stddev): 330398.0000/0.00
execution time (avg/stddev): 29.9677/0.00
```

2

```
[rishabh@rishabh:~$ ./10000_cpu.sh
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: 10956.62

General statistics:
total time: 30.0002s
total number of events: 328707

Latency (ms):
min: 0.09
avg: 0.09
max: 7.04
95th percentile: 0.10
sum: 29962.81

Threads fairness:
events (avg/stddev): 328707.0000/0.00
execution time (avg/stddev): 29.9628/0.00
```

3

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=..  
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: 10938.99  
  
General statistics:  
total time: 30.0002s  
total number of events: 328180  
  
Latency (ms):  
min: 0.09  
avg: 0.09  
max: 2.47  
95th percentile: 0.10  
sum: 29958.45  
  
Threads fairness:  
events (avg/stddev): 328180.0000/0.00  
execution time (avg/stddev): 29.9585/0.00
```

4

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=..  
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: 10991.52  
  
General statistics:  
total time: 30.0001s  
total number of events: 329754  
  
Latency (ms):  
min: 0.09  
avg: 0.09  
max: 3.02  
95th percentile: 0.10  
sum: 29962.81  
  
Threads fairness:  
events (avg/stddev): 329754.0000/0.00  
execution time (avg/stddev): 29.9628/0.00
```

5

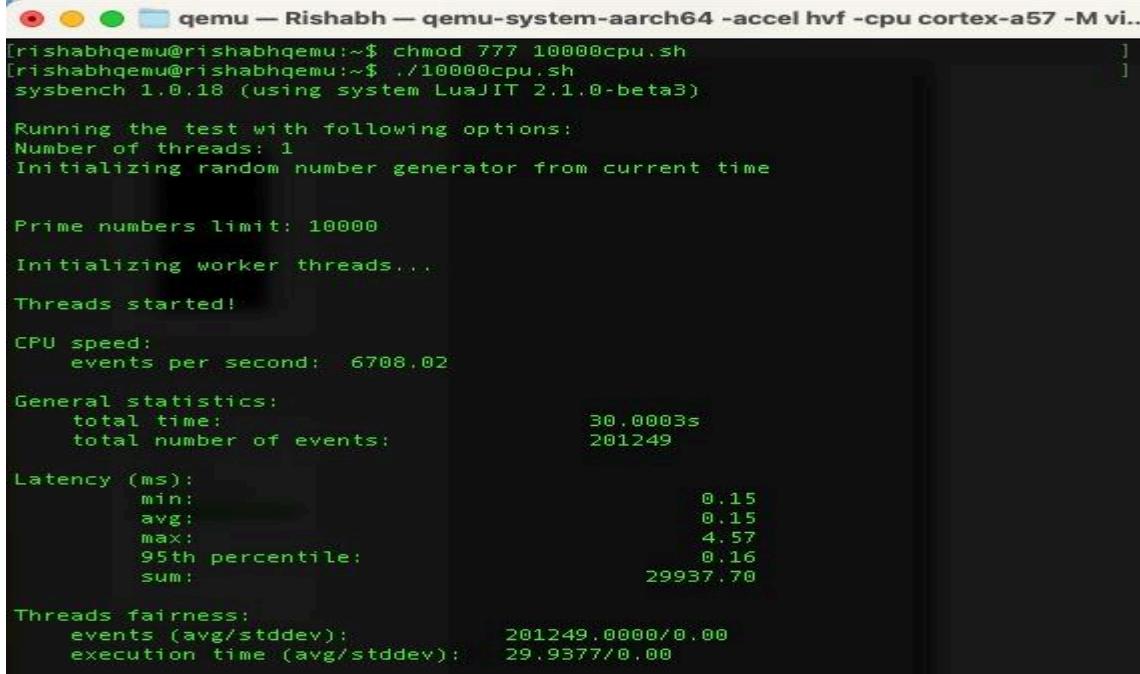
```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=..  
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)  
Running the test with following options:  
Number of threads: 1  
Initializing random number generator from current time  
  
Prime numbers limit: 10000  
Initializing worker threads...  
Threads started!  
  
CPU speed:  
events per second: 10938.99  
  
General statistics:  
total time: 30.0002s  
total number of events: 327937  
  
Latency (ms):  
min: 0.09  
avg: 0.09  
max: 3.96  
95th percentile: 0.10  
sum: 29958.62  
  
Threads fairness:  
events (avg/stddev): 327937.0000/0.00  
execution time (avg/stddev): 29.9586/0.00
```

Detailed report:

Iteration No	Events per second
1	11012.97
2	10956.62
3	10938.99
4	10991.52
5	10930.90

Observations:

Average events per second	10966.40
Minimum Number of events per second recorded	10930.90
Maximum Number of events per second recorded	11012.97

QEMU Results, Configuration: 2 GB 2 cores for max-prime = 10000 and time = 30 seconds, disk image = raw	
Screenshot	
 <pre>[rishabhqemu@rishabhqemu:~\$ chmod 777 10000cpu.sh [rishabhqemu@rishabhqemu:~\$./10000cpu.sh 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: 6708.02 General statistics: total time: 30.00003s total number of events: 201249 Latency (ms): min: 0.15 avg: 0.15 max: 4.57 95th percentile: 0.16 sum: 29937.70 Threads fairness: events (avg/stddev): 201249.0000/0.00 execution time (avg/stddev): 29.9377/0.00</pre>	

Detailed report:

Iteration No	Events/second
1	6708.02
2	6610.78
3	6725.68
4	6798.56
5	6649.98

Observations:

Average events per second	6698.60
Minimum Number of events per second recorded	6610.78
Maximum Number of events per second recorded	6798.56

Docker results for 2 GB 2 cores for max-prime = 10000 and time = 30 seconds

Dockerfile, sysbench command, building docker image and running it:

```
(base) sumit@Rishabh ~ % cat 10000_cpu.sh
#!/bin/bash
sysbench cpu --threads=1 --cpu-max-prime=10000 --time=30 run
(sysbench cpu --threads=1 --cpu-max-prime=10000 --time=30 run
(base) sumit@Rishabh ~ % cat 10000_CPU_Dockerfile
FROM arm64v8/ubuntu:20.04
RUN apt-get update
RUN apt-get install -y sysbench
COPY 10000_cpu.sh /
ENTRYPOINT ["/10000_cpu.sh"]
(base) sumit@Rishabh ~ %

(base) sumit@Rishabh ~ % docker build -t 10000cpudockerimage . -f 10000_CPU_Dockerfile
[*] Building 1.2s (10/10) FINISHED
docker:desktop-linux      0.0s
=> [internal] load build definition from 10000_CPU_Dockerfile          0.0s
=> transferring dockerfile: 174B                                     0.0s
=> [internal] load .dockerignore                                    0.0s
=> => transferring context: 28                                      0.0s
=> [internal] load metadata for docker.io/arm64v8/ubuntu:20.04       1.1s
=> [auth] arm64v8/ubuntu:pull token for registry-1.docker.io          0.0s
=> [1/4] FROM docker.io/arm64v8/ubuntu:20.04@sha256:080169816683eef063d390343456562428782ecfd06bd2f813b30325e8bleca 0.0s
=> [internal] load build context                                     0.0s
=> => transferring context: 350B                                     0.0s
=> => CACHED [2/4] RUN apt-get update                                0.0s
=> => CACHED [3/4] RUN apt-get install -y sysbench                  0.0s
=> => [4/4] COPY 10000_cpu.sh /                                       0.0s
=> exporting to image                                              0.0s
=> => exporting layers                                             0.0s
=> => writing image sha256:a575739f4bd76d63b9be62ad318ce279c3914d94d0eb3b708ccb82403af37a4 0.0s
=> => naming to docker.io/library/10000cpudockerimage               0.0s

What's Next?
View a summary of image vulnerabilities and recommendations - docker scout quickview
```

```
(base) sumit@Rishabh ~ % docker run -it --memory="2g" --cpuset-cpus="0-1" -d 10000cpudockerimage
965415d3e40f648ea3f32c29d3c0e6a35e3bd5a43b797dd5506ba2e3c15ad58e
(base) sumit@Rishabh ~ %
```

Iteration	Screenshot
1	<pre>(base) sumit@Rishabh ~ % docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS 965415d3e40f 10000cpudockerimage "/10000_cpu.sh" 36 seconds ago Up 36 seconds 8e4f0b8275f2 kindest/node:v1.27.3 "/usr/local/bin/entr..." 3 months ago Up 16 minutes 0.0.0.0:30000->30000/tcp, 127.0.0.1:65182->6443/tcp kind-control-plane (base) sumit@Rishabh ~ % docker logs -f naughty_bell 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: 6613.42 General statistics: total time: 30.0001s total number of events: 198410 Latency (ms): min: 0.15 avg: 0.15 max: 13.09 95th percentile: 0.16 sum: 29918.07 Threads fairness: events (avg/stddev): 198410.0000/0.00 execution time (avg/stddev): 29.9181/0.00</pre>
2	<pre>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: 6646.92 General statistics: total time: 30.0003s total number of events: 199418 Latency (ms): min: 0.15 avg: 0.15 max: 6.39 95th percentile: 0.16 sum: 29929.73 Threads fairness: events (avg/stddev): 199418.0000/0.00 execution time (avg/stddev): 29.9297/0.00</pre>
3	<pre>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: 6666.08 General statistics: total time: 30.0002s total number of events: 199990 Latency (ms): min: 0.15 avg: 0.15 max: 7.18 95th percentile: 0.16 sum: 29935.34 Threads fairness: events (avg/stddev): 199990.0000/0.00 execution time (avg/stddev): 29.9353/0.00</pre>

4	<pre> 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: 6599.67 General statistics: total time: 30.0001s total number of events: 197998 Latency (ms): min: 0.15 avg: 0.15 max: 9.51 95th percentile: 0.16 sum: 29909.47 Threads fairness: events (avg/stddev): 197998.0000/0.00 execution time (avg/stddev): 29.9095/0.00 </pre>
5	<pre> 95th percentile: 0.16 sum: 29909.47 Threads fairness: events (avg/stddev): 197998.0000/0.00 execution time (avg/stddev): 29.9095/0.00 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: 6635.00 General statistics: total time: 30.0004s total number of events: 199059 Latency (ms): min: 0.15 avg: 0.15 max: 7.69 95th percentile: 0.16 sum: 29920.39 Threads fairness: events (avg/stddev): 199059.0000/0.00 execution time (avg/stddev): 29.9204/0.00 (base) sumit@Rishabh ~ % </pre>

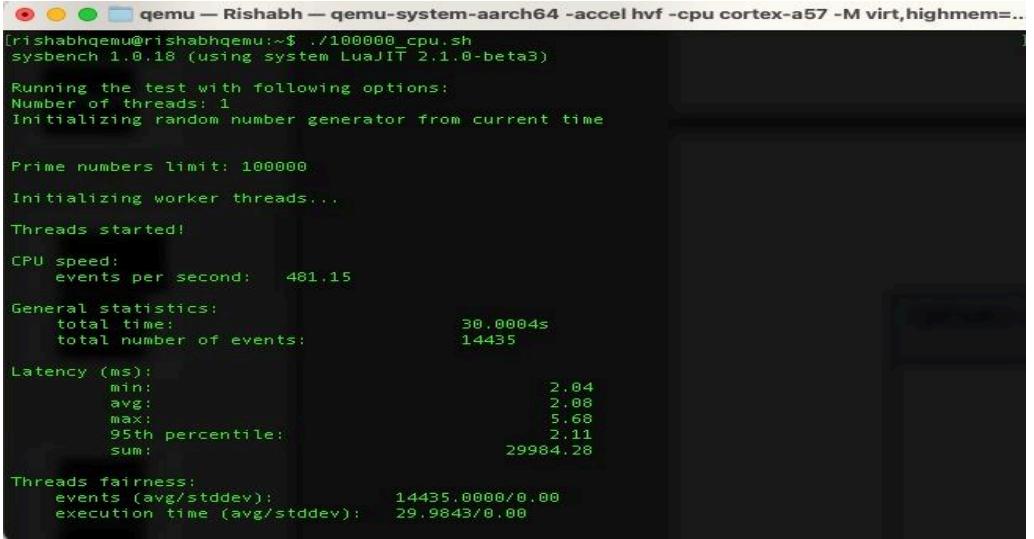
Detailed report:

Iteration No	Events per second
1	6613.42
2	6646.92
3	6666.08
4	6599.67
5	6635.00

Observations:

Average events per second	6632.22
Minimum Number of events per second recorded	6599.67
Maximum Number of events per second recorded	6666.08

Conclusion: Analysing the QEMU and Docker performances with a configuration of 2 GB RAM and 2 cores for a computational task (max-prime = 10000) over 30 seconds, distinct performance trends emerge. QEMU with a qcow2 disk image significantly outperforms its raw disk image counterpart and Docker in terms of events processed per second. Specifically, QEMU with qcow2 achieves an average of 10,966.40 events per second, whereas QEMU with a raw disk image and Docker report lower averages of 6,698.60 and 6,632.22 events per second, respectively. This highlights the efficiency of the qcow2 disk image in a QEMU environment for handling intensive computational tasks, compared to raw disk images and containerized applications in Docker.

QEMU Results for 2 GB 2 cores for max-prime = 100000 and time = 30 seconds, disk image = qcow2	
Iteration	Screenshot
1	 <pre> qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=.. [rishabh@rishabh:~/100000_cpu.sh] 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: 100000 Initializing worker threads... Threads started! CPU speed: events per second: 481.15 General statistics: total time: 30.0004s total number of events: 14435 Latency (ms): min: 2.04 avg: 2.08 max: 5.68 95th percentile: 2.11 sum: 29984.28 Threads fairness: events (avg/stddev): 14435.0000/0.00 execution time (avg/stddev): 29.9843/0.00 </pre>

2	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=.. 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: 100000 Initializing worker threads... Threads started! CPU speed: events per second: 476.61 General statistics: total time: 30.0022s total number of events: 14300 Latency (ms): min: 2.04 avg: 2.10 max: 59.97 95th percentile: 2.14 sum: 29980.91 Threads fairness: events (avg/stddev): 14300.0000/0.00 execution time (avg/stddev): 29.9809/0.00</pre>
3	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=.. 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: 100000 Initializing worker threads... Threads started! CPU speed: events per second: 475.76 General statistics: total time: 30.0018s total number of events: 14274 Latency (ms): min: 2.04 avg: 2.10 max: 4.01 95th percentile: 2.14 sum: 29982.16 Threads fairness: events (avg/stddev): 14274.0000/0.00 execution time (avg/stddev): 29.9822/0.00</pre>
4	

5

```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=...
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: 100000
Initializing worker threads...
Threads started!
CPU speed:
  events per second: 477.95

General statistics:
  total time:           30.0021s
  total number of events: 14340

Latency (ms):
  min:                  2.04
  avg:                  2.09
  max:                 12.34
  95th percentile:      2.14
  sum:                29980.44

Threads fairness:
  events (avg/stddev): 14340.0000/0.00
  execution time (avg/stddev): 29.9804/0.00

```

Detailed report:

Iteration No	Events per second
1	481.15
2	480.35
3	476.61
4	475.76
5	477.95

Observations:

Average events per second	478.36
Minimum Number of events per second recorded	475.76
Maximum Number of events per second recorded	481.15

QEMU Results, Configuration: 2 GB 2 cores for max-prime = 100000 and time = 30 seconds, disk image = raw

Screenshot

```
[rishabh@rishabhqemu:~$ ./100000cpu.sh
[rishabh@rishabhqemu:~$ chmod 777 100000cpu.sh
[rishabh@rishabhqemu:~$ ./100000cpu.sh
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: 100000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 291.74

General statistics:
total time: 30.00009s
total number of events: 8753

Latency (ms):
min: 3.35
avg: 3.42
max: 15.43
95th percentile: 3.49
sum: 29965.25

Threads fairness:
events (avg/stddev): 8753.0000/0.00
execution time (avg/stddev): 29.9652/0.00
```

Detailed report:

Iteration No	Events/second
1	291.74
2	290.78
3	291.56
4	290.47
5	291.85

Observations:

Average events per second	291.28
Minimum Number of events per second recorded	290.47
Maximum Number of events per second recorded	291.85

Docker results for 2 GB 2 cores for max-prime = 100000 and time = 30 seconds

Dockerfile, sysbench command, building docker image and running it:

```
(base) sumit@Rishabh ~ % cat 100000_cpu.sh
#!/bin/bash
sysbench cpu --threads=1 --cpu-max-prime=100000 --time=30 run
(base) sumit@Rishabh ~ %
(base) sumit@Rishabh ~ % cat 100000_CPU_Dockerfile
FROM arm64v8/ubuntu:20.04
RUN apt-get update
RUN apt-get install -y sysbench
COPY 100000_cpu.sh /
ENTRYPOINT ["/100000_cpu.sh"]
(base) sumit@Rishabh ~ %

(base) sumit@Rishabh ~ % docker build -t 100000cpudockerimage . -f 100000_CPU_Dockerfile
[+] Building 2.2s (10/10) FINISHED
=> [internal] load build definition from 100000_CPU_Dockerfile
=> transferring dockerfile: 179B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/arm64v8/ubuntu:20.04
=> [auth] arm64v8/ubuntu:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/arm64v8/ubuntu:20.04@sha256:088169816683e6f063d3903434565624287828ecfd06bd2f81
=> [internal] load build context
=> => transferring context: 364B
=> CACHED [2/4] RUN apt-get update
=> CACHED [3/4] RUN apt-get install -y sysbench
=> [4/4] COPY 100000_cpu.sh /
=> exporting to image
=> => exporting layers
=> => writing image sha256:00c53d96254f756398f735a6049f642fe326f3547f3c3efbf10e1a3c5b1ff874
=> => naming to docker.io/library/100000cpudockerimage
What's Next?
View a summary of image vulnerabilities and recommendations - docker scout quickview
(base) sumit@Rishabh ~ %

(base) sumit@Rishabh ~ % docker run -it --memory="2g" --cpuset-cpus="0-1" -d 100000cpudockerimage
0f70985dcc49c944b4435a98a086907b7878c7faf822ab36b943546053e6bd36
(base) sumit@Rishabh ~ % docker ps
CONTAINER ID IMAGE COMMAND NAMES CREATED STATUS PORTS
0f70985dcc49 100000cpudockerimage "./100000_cpu.sh" nostalgic_lewin 4 seconds ago Up 4 seconds
8e4f0b8275f2 kindest/node:v1.27.3 "/usr/local/bin/entr" kind-control-plane 3 months ago Up 21 minutes 0.0.0.0:30080->30080/tcp, 127.0.0.1:65182->6443/tcp
(base) sumit@Rishabh ~ %
```

Iteration

Screenshot

1	<pre>(base) sumit@Rishabh ~ % docker logs -f nostalgic_lewin 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: 100000 Initializing worker threads... Threads started! CPU speed: events per second: 289.65 General statistics: total time: 30.0038s total number of events: 8691 Latency (ms): min: 3.36 avg: 3.45 max: 12.38 95th percentile: 3.49 sum: 29960.24 Threads fairness: events (avg/stddev): 8691.0000/0.00 execution time (avg/stddev): 29.9602/0.00</pre>
2	<pre>sumit@Rishabh ~ % docker logs -f nostalgic_lewin 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: 100000 Initializing worker threads... Threads started! CPU speed: events per second: 290.96 General statistics: total time: 30.0027s total number of events: 8730 Latency (ms): min: 3.33 avg: 3.43 max: 8.27 95th percentile: 3.49 sum: 29958.19 Threads fairness: events (avg/stddev): 8730.0000/0.00 execution time (avg/stddev): 29.9582/0.00</pre>
3	<pre>sumit@Rishabh ~ % docker logs -f nostalgic_lewin 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: 100000 Initializing worker threads... Threads started! CPU speed: events per second: 290.08 General statistics: total time: 30.0009s total number of events: 8703 Latency (ms): min: 3.37 avg: 3.44 max: 10.71 95th percentile: 3.49 sum: 29958.33 Threads fairness: events (avg/stddev): 8703.0000/0.00 execution time (avg/stddev): 29.9583/0.00</pre>

4

```

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: 100000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 289.22

General statistics:
total time: 30.0003s
total number of events: 8677

Latency (ms):
min: 3.37
avg: 3.45
max: 11.15
95th percentile: 3.58
sum: 29963.93

Threads fairness:
events (avg/stddev): 8677.0000/0.00
execution time (avg/stddev): 29.9639/0.00

```

Detailed report:

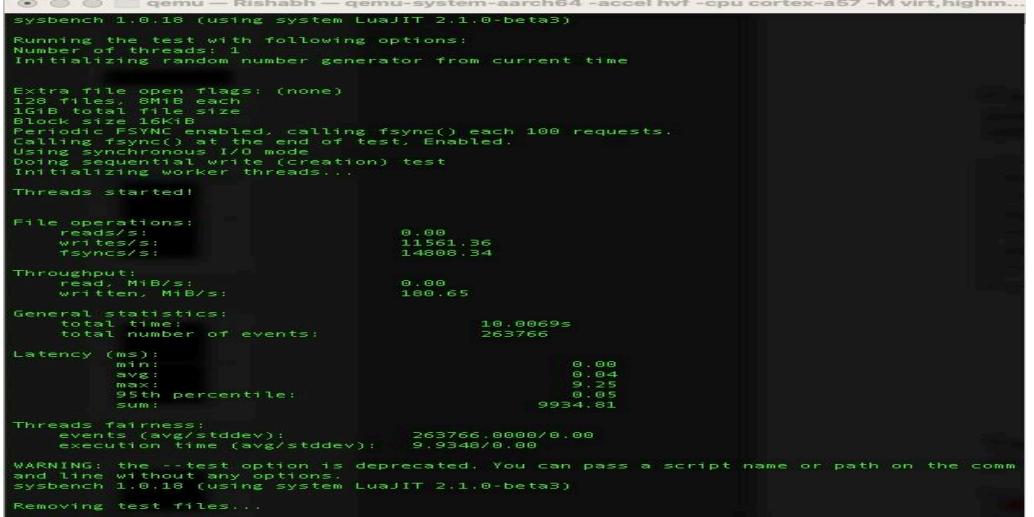
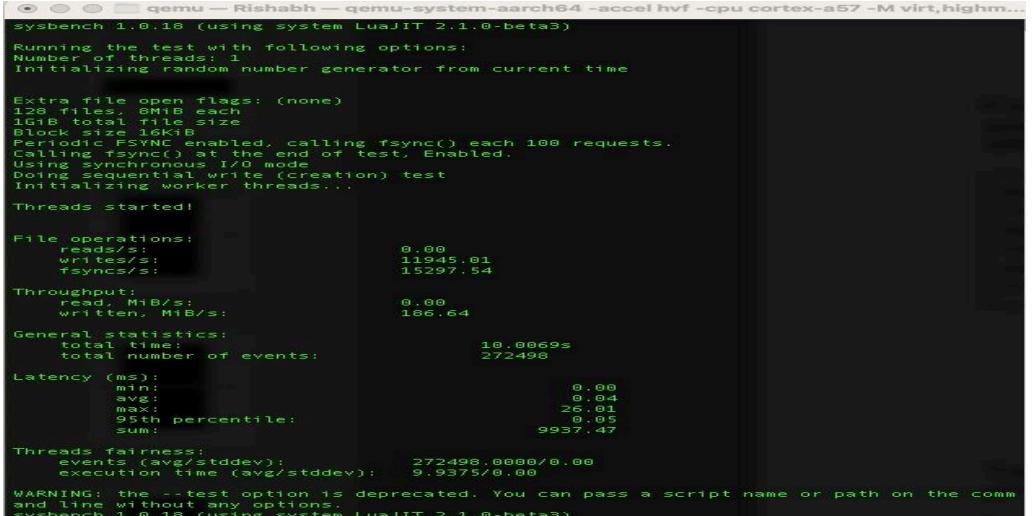
Iteration No	Events per second
1	289.65
2	290.96
3	290.08
4	289.22
5	289.94

Observations:

Average events per second	289.97
Minimum Number of events per second recorded	289.22
Maximum Number of events per second recorded	290.96

Conclusion: When examining the results for a computational task set to find the maximum prime numbers up to 100,000 within a 30-second timeframe on a setup with 2 GB RAM and 2 cores, it's evident that the performance varies significantly across different environments and disk image formats. QEMU with a qcow2 disk image leads in performance, achieving an average of 478.36 events per second. This is substantially higher compared to both QEMU with a raw disk image and Docker, which show averages of 291.28 and 289.97 events per second, respectively. The qcow2 disk image's efficiency in processing intensive computational tasks is highlighted, showing its advantage over raw disk images and Docker environments for this specific benchmark.

1.2 File I/O Testing

QEMU Results for 2 GB 2 cores for sequential write, disk image = qcow2	
Iteration	Results
1	 <pre> qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem... 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 Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size: 4KiB Periodic FSync enabled, calling fsync() each 100 requests. Calling fsync() at the end of test. Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 11561.36 fsyncs/s: 14806.34 Throughput: read, MiB/s: 0.00 written, MiB/s: 180.65 General statistics: total time: 10.0069s total number of events: 263766 Latency (ms): min: 0.00 avg: 0.00 max: 9.55 95th percentile: 0.05 sum: 9934.81 Threads fairness: events (avg/stddev): 263766.0000/0.00 execution time (avg/stddev): 9.9348/0.00 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) Removing test files... </pre>
2	 <pre> qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem... 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 Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size: 4KiB Periodic FSync enabled, calling fsync() each 100 requests. Calling fsync() at the end of test. Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 11945.01 fsyncs/s: 15297.54 Throughput: read, MiB/s: 0.00 written, MiB/s: 186.64 General statistics: total time: 10.0069s total number of events: 272498 Latency (ms): min: 0.00 avg: 0.04 max: 26.01 95th percentile: 0.05 sum: 9937.47 Threads fairness: events (avg/stddev): 272498.0000/0.00 execution time (avg/stddev): 9.9375/0.00 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) Removing test files... </pre>

3	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem 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 Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 12279.04 fsyncs/s: 15718.18 Throughput: read, MiB/s: 0.00 written, MiB/s: 191.86 General statistics: total time: 10.00082s total number of events: 280094 Latency (ms): min: 0.00 avg: 0.04 max: 11.93 95th percentile: 0.06 sum: 9934.08 Threads fairness: events (avg/stddev): 280094.0000/0.00 execution time (avg/stddev): 9.9341/0.00 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) Removing test files...</pre>
4	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem 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 Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 12098.64 fsyncs/s: 15495.95 Throughput: read, MiB/s: 0.00 written, MiB/s: 189.04 General statistics: total time: 10.00087s total number of events: 276077 Latency (ms): min: 0.00 avg: 0.04 max: 7.79 95th percentile: 0.05 sum: 9935.18 Threads fairness: events (avg/stddev): 276077.0000/0.00 execution time (avg/stddev): 9.9352/0.00 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) Removing test files...</pre>
5	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem 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 Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 12468.73 fsyncs/s: 15967.77 Throughput: read, MiB/s: 0.00 written, MiB/s: 194.82 General statistics: total time: 10.00084s total number of events: 284494 Latency (ms): min: 0.00 avg: 0.03 max: 17.94 95th percentile: 0.05 sum: 9932.12 Threads fairness: events (avg/stddev): 284494.0000/0.00 execution time (avg/stddev): 9.9324/0.00 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) Removing test files...</pre>

QEMU Results for 2 GB 2 cores for random read, disk image = qcow2

Iteration	Results
1	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Number of I/O requests: 8 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 read test Initializing worker threads... Threads started! File operations: reads/s: 461847.82 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 7283.86 written, MiB/s: 0.00 General statistics: total time: 19.000028 total number of events: 4610833 Latency (ms): min: 0.00 avg: 0.00 max: 0.10 95th percentile: 0.00 sum: 9339.26 Threads fairness: events (avg/stddev): 4610833.0000/0.00 execution time (avg/stddev): 9.339370.00 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) Removing test files...</pre>
2	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Number of I/O requests: 8 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 read test Initializing worker threads... Threads started! File operations: reads/s: 467245.78 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 7300.72 written, MiB/s: 0.00 General statistics: total time: 19.000018 total number of events: 4672771 Latency (ms): min: 0.00 avg: 0.00 max: 0.08 95th percentile: 0.00 sum: 9346.91 Threads fairness: events (avg/stddev): 4672771.0000/0.00 execution time (avg/stddev): 9.346970.00 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) Removing test files...</pre>

```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Number of IO requests: 8
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 read test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          461335.61
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     7208.37
  written, MiB/s:  0.00

General statistics:
  total time:      10.0001s
  total number of events: 4613686

Latency (ms):
  min:              0.00
  avg:              0.00
  max:              0.00
  95th percentile: 0.00
  sum:              9345.00

Threads fairness:
  events (avg/stddev): 4613686.0000/0.00
  execution time (avg/stddev): 9.3458/0.00

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)

Removing test files...

```

3


```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Number of IO requests: 8
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 read test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          463191.84
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     7237.37
  written, MiB/s:  0.00

General statistics:
  total time:      10.0001s
  total number of events: 4632182

Latency (ms):
  min:              0.00
  avg:              0.00
  max:              0.00
  95th percentile: 0.00
  sum:              9354.15

Threads fairness:
  events (avg/stddev): 4632182.0000/0.00
  execution time (avg/stddev): 9.3541/0.00

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)

Removing test files...

```

4


```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Number of IO requests: 8
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 read test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          455978.15
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     7124.66
  written, MiB/s:  0.00

General statistics:
  total time:      10.0001s
  total number of events: 4560101

Latency (ms):
  min:              0.00
  avg:              0.00
  max:              0.1
  95th percentile: 0.00
  sum:              9345.98

Threads fairness:
  events (avg/stddev): 4560101.0000/0.00
  execution time (avg/stddev): 9.3460/0.00

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)

Removing test files...
File I/O tests completed.
rishabh@rishabh-qemu:~$ 

```

5

**QEMU Results for 2 GB 2 cores for sequential write & random read,
disk image = qcow2**

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 11561.36 fsyncs/s: 14808.34	reads/s: 461047.02 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 11945.01 fsyncs/s: 15297.54	reads/s: 467245.78 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 12279.04 fsyncs/s: 15718.18	reads/s: 461335.61 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 12098.64 fsyncs/s: 15495.95	reads/s: 463191.84 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 12468.73 fsyncs/s: 15967.77	reads/s: 455978.15 writes/s: 0.00 fsyncs/s: 0.00

QEMU Results for 2 GB 2 cores for sequential write & random read,
disk image = raw

Screenshot

```
qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi..
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        7834.46
  fsyncs/s:       10030.71

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   122.41

General statistics:
  total time:           10.0061s
  total number of events: 178650

Latency (ms):
  min:                  0.00
  avg:                  0.05
  max:                 7.95
  95th percentile:     0.10
  sum:                9798.38

Threads fairness:
  events (avg/stddev): 178650.0000/0.00
  execution time (avg/stddev): 9.7984/0.00

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)

Removing test files...
```

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 7834.46 fsyncs/s: 10030.71	reads/s: 431234.58 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 7947.40 fsyncs/s: 11059.39	reads/s: 441236.96 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 8103.39 fsyncs/s: 11305.40	reads/s: 421589.04 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 78123.93 fsyncs/s: 10945.63	reads/s: 439128.44 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 79482.73 fsyncs/s: 11456.96	reads/s: 444950.40 writes/s: 0.00 fsyncs/s: 0.00

Docker Results for 2 GB 2 cores for sequential write

Iteration	Results
1	<pre> sumit — Rishabh — com.docker.cli - docker logs -f frosty_mcnulty — 87x48 Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 11930.49 fsyncs/s: 15281.52 Throughput: read, MiB/s: 0.00 written, MiB/s: 186.41 General statistics: total time: 10.0070s total number of events: 272209 Latency (ms): min: 0.01 avg: 0.04 max: 41.84 95th percentile: 0.05 sum: 9848.38 Threads fairness: events (avg/stddev): 272209.0000/0.00 execution time (avg/stddev): 9.8484/0.00 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) Removing test files... </pre>
2	<pre> sumit — Rishabh — -zsh — 87x48 Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 9711.84 fsyncs/s: 12431.66 Throughput: read, MiB/s: 0.00 written, MiB/s: 151.75 General statistics: total time: 10.0074s total number of events: 221493 Latency (ms): min: 0.01 avg: 0.04 max: 42.77 95th percentile: 0.06 sum: 9894.92 Threads fairness: events (avg/stddev): 221493.0000/0.00 execution time (avg/stddev): 9.8949/0.00 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) Removing test files... </pre>

```
sumit--Rishabh--zsh--87x48

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

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size: 64KiB
File periodic FSync enabled, calling fSync() each 100 requests.
Calling fSync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:                      0.00
  writes/s:                     9626.79
  fsyncs/s:                      12326.59

Throughput:
  read, MiB/s:                  0.00
  written, MiB/s:                150.42

General statistics:
  total time:                   10.0127s
  total number of events:        219707

Latency (ms):
  min:                           0.01
  avg:                           0.05
  max:                           7.00
  95th percentile:               0.06
  sum:                          9896.84

Threads fairness:
  events (avg/stddev):          219707.0000/0.00
  execution time (avg/stddev):   9.8968/0.00

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)

Removing test files...
```

3

```
[sumit — Rishabh — zsh — 87x48]
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
16GiB total file size
Block size: 16KiB
File periodic FSync enabled, calling fSync() each 100 requests.
Calling fSync() at the end of test: Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        11367.01
  fsyncs/s:       14550.87

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   177.61

General statistics:
  total time:           19.01065
  total number of events: 259347

Latency (ms):
  min:                  0.01
  avg:                  0.04
  max:                 21.07
  95th percentile:      0.05
  sum:                9851.35

Threads fairness:
  events (avg/stddev): 259347.0000/0.00
  execution time (avg/stddev): 9.8514/0.00

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)

Removing test files...
```

4

```
[sumit — Rishabh — zsh — 87x48]
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:                      0.00
  writes/s:                     18932.98
  fsyncs/s:                      14002.81

Throughput:
  read, MiB/s:                  0.00
  written, MiB/s:                170.83

General statistics:
  total time:                   10.0056s
  total number of events:        249390

Latency (ms):
  min:                           0.01
  avg:                           0.04
  max:                           13.39
  95th percentile:               0.05
  sum:                           9852.91

Threads fairness:
  events (avg/stddev):          249390.0000/0.00
  execution time (avg/stddev):   9.8529/0.00

WARNING: the -test option is deprecated. You can pass a script name or path on the command line without any options. (use ./bsbebench 1.0.18 (using system libltt 2.1 @-beta))
```

5

Docker Results for 2 GB 2 cores for random read

Iteration	Results
1	<pre>sumit — Rishabh — -zsh — 87x48 Number of threads: 1 Initializing random number generator from current time Extra file open flags: (none) 128 files, 8MB each 1GB total file size Block size 16KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Per-thread FSync enabled, calling fsync() each 100 requests. Calling fSync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations: reads/s: 11673.76 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 182.40 written, MiB/s: 0.00 General statistics: total time: 18.0005s total number of events: 116754 Latency (ms): min: 0.00 avg: 0.09 max: 6.90 95th percentile: 0.29 sum: 9946.00 Threads fairness: events (avg/stddev): 116754.0000/0.00 execution time (avg/stddev): 9.9468/0.00 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) Removing test files...</pre>
2	<pre>sumit — Rishabh — -zsh — 87x48 Number of threads: 1 Initializing random number generator from current time Extra file open flags: (none) 128 files, 8MB each 1GB total file size Block size 16KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Per-thread FSync enabled, calling fsync() each 100 requests. Calling fSync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations: reads/s: 14093.78 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 228.22 written, MiB/s: 0.00 General statistics: total time: 18.0008s total number of events: 140964 Latency (ms): min: 0.00 avg: 0.07 max: 8.25 95th percentile: 0.27 sum: 9937.26 Threads fairness: events (avg/stddev): 140964.0000/0.00 execution time (avg/stddev): 9.9373/0.00 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) Removing test files...</pre>

3

```

sumit@Rishabh: ~ - zsh - 87x48
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
16GiB 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 read test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          15400.25
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     248.63
  written, MiB/s:  0.00

General statistics:
  total time:      18.00095s
  total number of events: 154035

Latency (ms):
  min:              0.00
  avg:              0.06
  max:              1.03
  95th percentile:  0.27
  sum:              9920.54

Threads fairness:
  events (avg/stddev): 154035.0000/0.00
  execution time (avg/stddev): 9.9205/0.00

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)
Removing test files...

```

4

```

sumit@Rishabh: ~ - zsh - 87x48
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
16GiB 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 read test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          22329.15
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     348.89
  written, MiB/s:  0.00

General statistics:
  total time:      18.00010s
  total number of events: 223338

Latency (ms):
  min:              0.00
  avg:              0.04
  max:              28.09
  95th percentile:  0.23
  sum:              9900.66

Threads fairness:
  events (avg/stddev): 223338.0000/0.00
  execution time (avg/stddev): 9.90007/0.00

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)
Removing test files...

```

5

```

sumit@Rishabh: ~ - zsh - 87x48
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
16GiB 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 read test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          24967.68
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     390.12
  written, MiB/s:  0.00

General statistics:
  total time:      18.00060s
  total number of events: 249857

Latency (ms):
  min:              0.00
  avg:              0.04
  max:              42.76
  95th percentile:  0.23
  sum:              9885.22

Threads fairness:
  events (avg/stddev): 249857.0000/0.00
  execution time (avg/stddev): 9.8852/0.00

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)
Removing test files...
File I/O tests completed.
(base) sumit@Rishabh: ~ %

```

Docker Results for 2 GB 2 cores for sequential write & random read		
Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 11930.49 fsyncs/s: 15281.52	reads/s: 11673.76 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 9711.84 fsyncs/s: 12431.66	reads/s: 14093.78 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 9626.79 fsyncs/s: 12326.59	reads/s: 15400.25 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 11367.01 fsyncs/s: 14550.07	reads/s: 22329.15 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 10932.98 fsyncs/s: 14002.81	reads/s: 24967.68 writes/s: 0.00 fsyncs/s: 0.00

Conclusion: The performance for sequential write and random read operations is notably superior in the QEMU environment with qcow2 disk images compared to raw images and Docker, especially in random read operations. The observed data suggests that for environments requiring high I/O performance, QEMU with qcow2 might offer the best efficiency. However, Docker's performance, while lower in random reads, might still be considered for applications with lower I/O demands or where containerization benefits outweigh raw performance metrics. The anomaly in the raw disk image data suggests a need for review to ensure accuracy.

1.3 Memory Testing

QEMU Results for 2 GB 2 cores for Sequential Access, disk image = qcow2	
Iteration	Results
1	<pre>[rishabhqemu@rishabhqemu:~\$ chmod 777 memorytest.sh [rishabhqemu@rishabhqemu:~\$./memorytest.sh Running First Memory Test: Sequential Access Iteration 1 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: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 86085608 (8607197.97 per second) 84067.98 MiB transferred (8405.47 MiB/sec) General statistics: total time: 10.0002s total number of events: 86085608 Latency (ms): min: 0.00 avg: 0.00 max: 0.06 95th percentile: 0.00 sum: 4191.76 Threads fairness: events (avg/stddev): 86085608.0000/0.00 execution time (avg/stddev): 4.1918/0.00</pre>
2	<pre>Iteration 2 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 86743655 (8673170.97 per second) 84710.60 MiB transferred (8469.89 MiB/sec) General statistics: total time: 10.0001s total number of events: 86743655 Latency (ms): min: 0.00 avg: 0.00 max: 0.08 95th percentile: 0.00 sum: 4192.50 Threads fairness: events (avg/stddev): 86743655.0000/0.00 execution time (avg/stddev): 4.1925/0.00</pre>

3

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem
Iteration 3
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 85979129 (8597112.10 per second)
83963.99 MiB transferred (8395.62 MiB/sec)

General statistics:
    total time:                      10.0001s
    total number of events:          85979129

Latency (ms):
    min:                            0.00
    avg:                            0.00
    max:                            0.00
    95th percentile:                0.00
    sum:                           4195.14

Threads fairness:
    events (avg/stddev):          85979129.0000/0.00
    execution time (avg/stddev):   4.1951/0.00
```

4

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem
Iteration 4
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 86293905 (8628581.17 per second)
84271.39 MiB transferred (8426.35 MiB/sec)

General statistics:
    total time:                      10.0001s
    total number of events:          86293905

Latency (ms):
    min:                            0.00
    avg:                            0.00
    max:                            0.13
    95th percentile:                0.00
    sum:                           4193.52

Threads fairness:
    events (avg/stddev):          86293905.0000/0.00
    execution time (avg/stddev):   4.1935/0.00
```

5

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem
Iteration 5
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

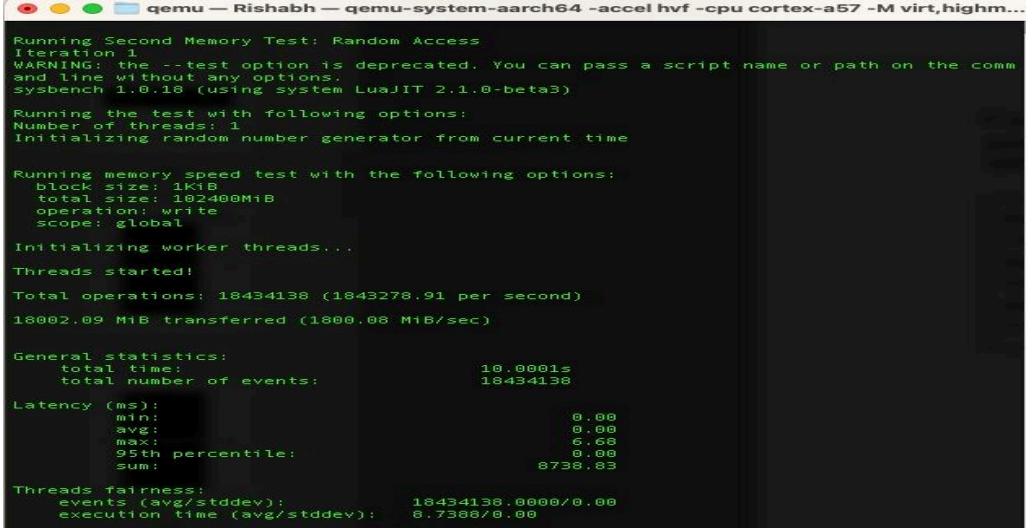
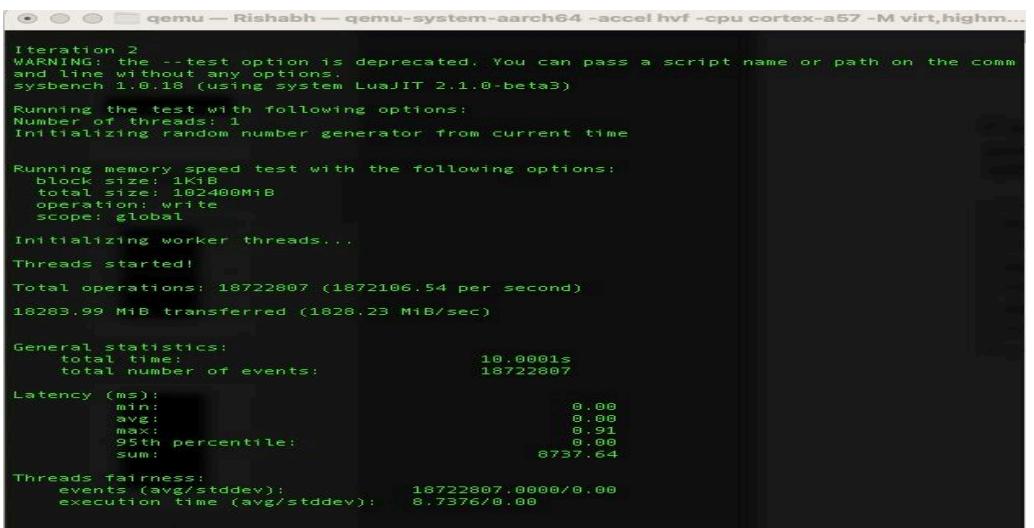
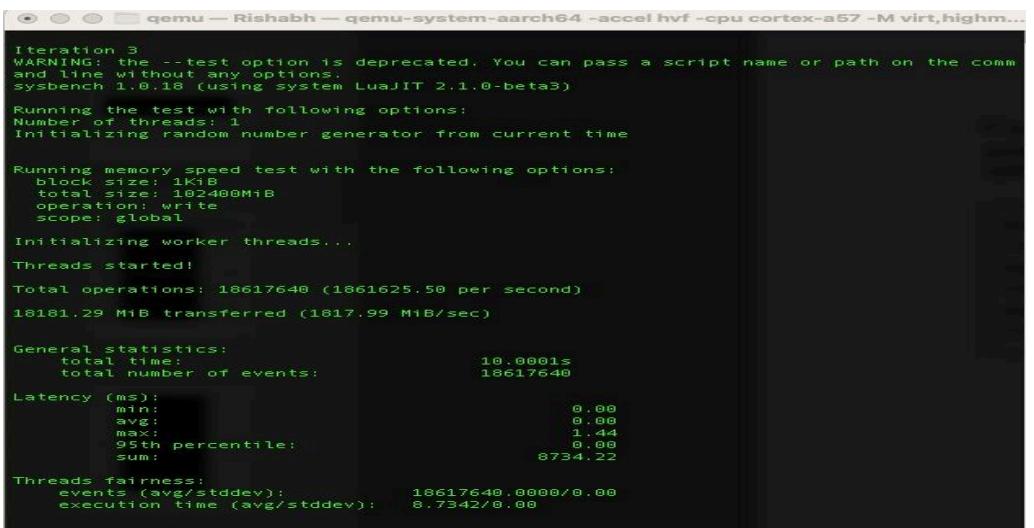
Initializing worker threads...
Threads started!
Total operations: 86788175 (8677874.81 per second)
84754.08 MiB transferred (8474.49 MiB/sec)

General statistics:
    total time:                      10.0002s
    total number of events:          86788175

Latency (ms):
    min:                            0.00
    avg:                            0.00
    max:                            0.05
    95th percentile:                0.00
    sum:                           4192.15

Threads fairness:
    events (avg/stddev):          86788175.0000/0.00
    execution time (avg/stddev):   4.1922/0.00
```

QEMU Results for 2 GB 2 cores for Random Access, disk image = qcow2

Iteration	Results
1	 <pre> qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=2G Running Second Memory Test: Random Access Iteration 1 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 Running memory speed test with the following options: block size: 1KiB total size: 1024000MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18434138 (1843278.91 per second) 18002.69 MiB transferred (1800.08 MiB/sec) General statistics: total time: 10.00001s total number of events: 18434138 Latency (ms): min: 0.00 avg: 0.00 max: 6.68 95th percentile: 0.00 sum: 8738.83 Threads fairness: events (avg/stddev): 18434138.0000/0.00 execution time (avg/stddev): 8.7388/0.00 </pre>
2	 <pre> qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=2G Iteration 2 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 Running memory speed test with the following options: block size: 1KiB total size: 1024000MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18722807 (1872106.54 per second) 18283.99 MiB transferred (1828.23 MiB/sec) General statistics: total time: 10.00001s total number of events: 18722807 Latency (ms): min: 0.00 avg: 0.00 max: 0.91 95th percentile: 0.00 sum: 8737.64 Threads fairness: events (avg/stddev): 18722807.0000/0.00 execution time (avg/stddev): 8.7376/0.00 </pre>
3	 <pre> qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem=2G Iteration 3 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 Running memory speed test with the following options: block size: 1KiB total size: 1024000MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18617640 (1861625.50 per second) 18181.29 MiB transferred (1817.99 MiB/sec) General statistics: total time: 10.00001s total number of events: 18617640 Latency (ms): min: 0.00 avg: 0.00 max: 1.44 95th percentile: 0.00 sum: 8734.22 Threads fairness: events (avg/stddev): 18617640.0000/0.00 execution time (avg/stddev): 8.7342/0.00 </pre>

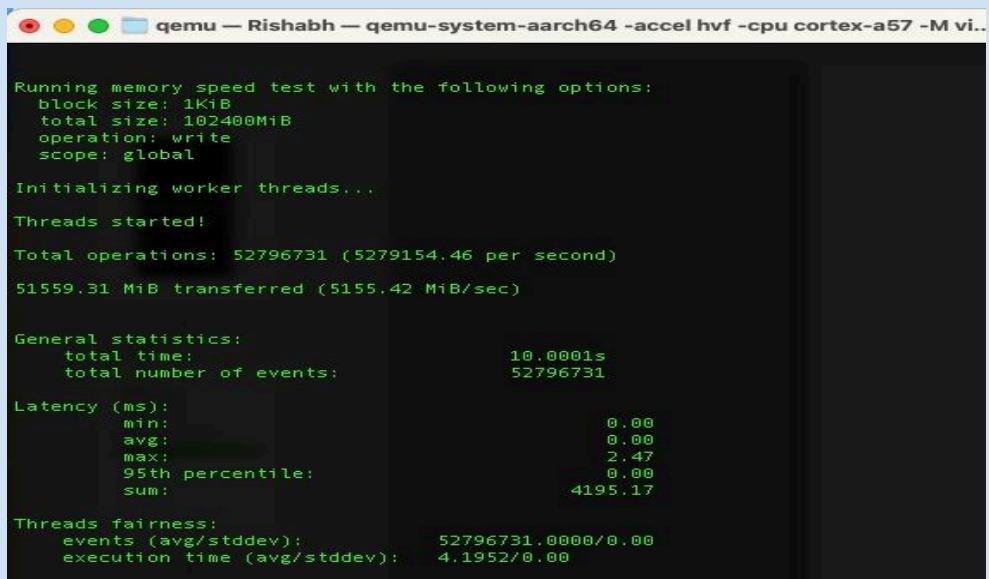
4	<pre> qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem Iteration 4 WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options. sysbench 1.0.18 (using system LuajIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 1024000MiB operations: write scope: global Initializing worker threads... Threads started! Total operations: 18668957 (1866709.29 per second) 18231.40 MiB transferred (1822.96 MiB/sec) General statistics: total time: 10.0002s total number of events: 18668957 Latency (ms): min: 0.00 avg: 0.00 max: 0.00 95th percentile: 0.00 sum: 8736.86 Threads fairness: events (avg/stddev): 18668957.0000/0.00 execution time (avg/stddev): 8.7389/0.00 </pre>
5	<pre> qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,highmem Iteration 5 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 Running memory speed test with the following options: block size: 1KiB total size: 1024000MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18634070 (1863256.21 per second) 18197.33 MiB transferred (1819.59 MiB/sec) General statistics: total time: 10.0001s total number of events: 18634070 Latency (ms): min: 0.00 avg: 0.00 max: 4.00 95th percentile: 0.00 sum: 8735.56 Threads fairness: events (avg/stddev): 18634070.0000/0.00 execution time (avg/stddev): 8.7356/0.00 Memory tests completed. </pre>

QEMU Results for 2 GB 2 cores for sequential & random access, disk image = qcow2

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	8607197.97	1843278.91
2	8673170.97	1872106.54
3	8597112.10	1861625.50
4	8628581.17	1866709.29
5	8677874.81	1863256.21

QEMU Results for 2 GB 2 cores for sequential & random access, disk image = raw

Screenshot



The screenshot shows a terminal window titled "qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi..". The window displays the output of a memory speed test. The text in the terminal is as follows:

```
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 52796731 (5279154.46 per second)
51559.31 MiB transferred (5155.42 MiB/sec)

General statistics:
  total time: 10.00001s
  total number of events: 52796731

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 2.47
  95th percentile: 0.00
  sum: 4195.17

Threads fairness:
  events (avg/stddev): 52796731.0000/0.00
  execution time (avg/stddev): 4.1952/0.00
```

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	5279154.46	1145006.45
2	5397439.29	1138561.78
3	5434983.49	1175682.47
4	5245939.34	1126844.85
5	5434572.35	1135773.56

Docker Results for 2 GB 2 cores for Sequential Access

```

sumit — Rishabh — zsh — 87x48
[[base] sumit@Rishabh ~ % vim memorytestdockerfile
[[base] sumit@Rishabh ~ % chmod 777 memorytest.sh
[[base] sumit@Rishabh ~ % cat memorytestdockerfile
FROM arm64v8/ubuntu:20.04
RUN apt-get update
RUN apt-get install -y sysbench
COPY memorytest.sh /
ENTRYPOINT ["./memorytest.sh"]

(base) sumit@Rishabh ~ %

(base) sumit@Rishabh ~ % docker build -t memorytestdockerimage . -f memorytestdockerfile
[+] Building 1.2s (10/10) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from memorytestdockerfile
=> => transferring dockerfile: 179B
=> [internal] load metadata for docker.io/arm64v8/ubuntu:20.04
=> [auth] arm64v8/ubuntu:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/arm64v8/ubuntu:20.04@sha256:080169816683e6f063d39034345
=> [internal] load build context
=> => transferring context: 617B
=> CACHED [2/4] RUN apt-get update
=> CACHED [3/4] RUN apt-get install -y sysbench
=> [4/4] COPY memorytest.sh /
=> exporting to image
=> => exporting layers
=> => writing image sha256:93b21fe3d8a3137bbc55b07ed80c4a0313ef9604efce2faa179
=> => naming to docker.io/library/memorytestdockerimage

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
(base) sumit@Rishabh ~ %

(base) sumit@Rishabh ~ % docker run -it --memory="2g" --cpuset-cpus="0-1" -d memorytestdockerimage
c4b29fe2022ca84271c82babf191058a54256defd0040bc4f1d7699be60b6f94
(base) sumit@Rishabh ~ % docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
c4b29fe2022c memorytestdockerimage "/.memorytest.sh" 4 seconds ago Up 2 seconds
8e4f0b8275f2 kindest/node:v1.27.3 "/usr/local/bin/entr..." 3 months ago Up 48 minutes 0.0.0
.0:30080->30080/tcp, 127.0.0.1:65182->6443/tcp kind-control-plane
(base) sumit@Rishabh ~ %

```

Iteration	Results
1	<pre> sumit — Rishabh — com.docker.cli - docker logs -f trusting_hertz — 101x48 (base) sumit@Rishabh ~ % docker logs -f trusting_hertz Running First Memory Test: Sequential Access Iteration 1 WARNING: the --test option is deprecated. You can pass a script name or path on the command line with out 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 51701114 (5169532.54 per second) 50489.37 MiB transferred (5048.37 MiB/sec) General statistics: total time: 10.00002s total number of events: 51701114 Latency (ms): min: 0.00 avg: 0.00 max: 0.56 95th percentile: 0.00 sum: 4266.11 Threads fairness: events (avg/stddev): 51701114.0000/0.00 execution time (avg/stddev): 4.2661/0.00 </pre>

2	<pre>sumit — Rishabh — com.docker.cli - docker logs -f trusting_hertz — 101x48 Iteration 2 WARNING: the --test option is deprecated. You can pass a script name or path on the command line with out 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 51215170 (5120980.21 per second) 50014.81 MiB transferred (5000.96 MiB/sec) General statistics: total time: 10.0001s total number of events: 51215170 Latency (ms): min: 0.00 avg: 0.00 max: 4.15 95th percentile: 0.00 sum: 4260.56 Threads fairness: events (avg/stddev): 51215170.0000/0.00 execution time (avg/stddev): 4.2606/0.00</pre>
3	<pre>sumit — Rishabh — com.docker.cli - docker logs -f trusting_hertz — 101x48 Iteration 3 WARNING: the --test option is deprecated. You can pass a script name or path on the command line with out 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 51868274 (5186215.14 per second) 50652.61 MiB transferred (5064.66 MiB/sec) General statistics: total time: 10.0002s total number of events: 51868274 Latency (ms): min: 0.00 avg: 0.00 max: 1.22 95th percentile: 0.00 sum: 4262.04 Threads fairness: events (avg/stddev): 51868274.0000/0.00 execution time (avg/stddev): 4.2620/0.00</pre>
4	<pre>sumit — Rishabh — com.docker.cli - docker logs -f trusting_hertz — 101x48 Iteration 4 WARNING: the --test option is deprecated. You can pass a script name or path on the command line with out 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 52129816 (5212455.49 per second) 50908.02 MiB transferred (5090.29 MiB/sec) General statistics: total time: 10.0001s total number of events: 52129816 Latency (ms): min: 0.00 avg: 0.00 max: 0.16 95th percentile: 0.00 sum: 4274.79 Threads fairness: events (avg/stddev): 52129816.0000/0.00 execution time (avg/stddev): 4.2748/0.00</pre>

5

```
sumit — Rishabh — com.docker.cli - docker logs -f trusting_hertz — 101x48

Iteration 5
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 52022702 (5201740.66 per second)
50803.42 MiB transferred (5079.82 MiB/sec)

General statistics:
    total time:                      10.0001s
    total number of events:           52022702

Latency (ms):
    min:                            0.00
    avg:                            0.00
    max:                            0.50
    95th percentile:                0.00
    sum:                           4272.70

Threads fairness:
    events (avg/stddev):          52022702.0000/0.00
    execution time (avg/stddev):   4.2727/0.00
```

Docker Results for 2 GB 2 cores for Random Access

Iteration	Results
1	<pre>sumit — RISHABH — ~zsn — 101x48 Running Second Memory Test: Random Access Iteration 1 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 11054691 (1105232.20 per second) 10795.60 MiB transferred (1079.33 MiB/sec) General statistics: total time: 10.0005s total number of events: 11054691 Latency (ms): min: 0.00 avg: 0.00 max: 10.97 95th percentile: 0.00 sum: 8721.05 Threads fairness: events (avg/stddev): 11054691.0000/0.00 execution time (avg/stddev): 8.7211/0.00</pre>

```

2
SUMIT — KISHADH — -ZSN — 101X48
Iteration 2
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 11036374 (1103522.48 per second)
10777.71 MiB transferred (1077.66 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 11036374

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 6.02
  95th percentile: 0.00
  sum: 8693.11

Threads fairness:
  events (avg/stddev): 11036374.0000/0.00
  execution time (avg/stddev): 8.6931/0.00

3
SUMIT — KISHADH — -ZSN — 101X48
Iteration 3
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 11257698 (1125657.15 per second)
10993.85 MiB transferred (1099.27 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 11257698

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 1.95
  95th percentile: 0.00
  sum: 8792.56

Threads fairness:
  events (avg/stddev): 11257698.0000/0.00
  execution time (avg/stddev): 8.7324/0.00

4
SUMIT — KISHADH — -ZSN — 101X48
Iteration 4
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 11171044 (1116990.80 per second)
10909.22 MiB transferred (1090.81 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 11171044

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 3.22
  95th percentile: 0.00
  sum: 8712.47

Threads fairness:
  events (avg/stddev): 11171044.0000/0.00
  execution time (avg/stddev): 8.7125/0.00

```

```

Iteration 5
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 11161572 (1116045.19 per second)
10899.97 MiB transferred (1089.89 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 11161572

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 6.02
  95th percentile: 0.00
  sum: 8723.56

Threads fairness:
  events (avg/stddev): 11161572.0000/0.00
  execution time (avg/stddev): 8.7236/0.00

Memory tests completed.
(base) sumit@Rishabh ~ %

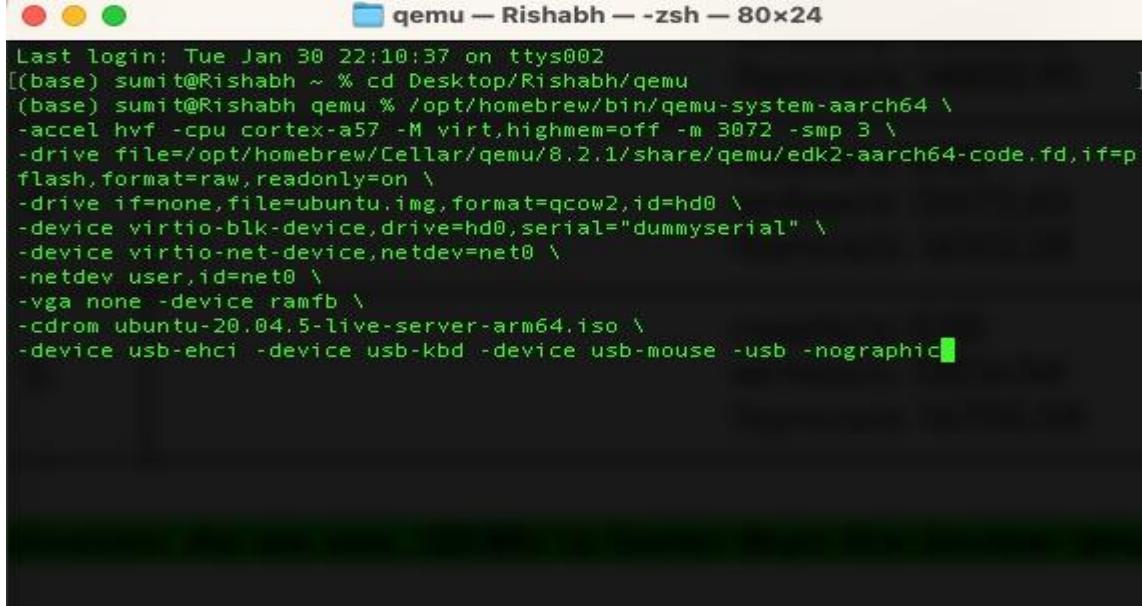
```

Docker Results for 2 GB 2 cores for sequential & random access		
Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	5169532.54	1105232.20
2	5120980.21	1103522.48
3	5186215.14	1125657.15
4	5212455.49	1116990.80
5	5201740.66	1116045.19

Conclusion: The QEMU environment with a qcow2 disk image configuration significantly outperforms both the QEMU with a raw disk image and Docker environments in terms of both sequential and random access. This indicates that the qcow2 disk image format is more efficient for high-intensity I/O operations. While the QEMU raw disk image and Docker environments offer lower performance, they still provide substantial throughput for many applications. Docker's performance, while the lowest in the comparison, may still be adequate for a wide range of applications, especially considering the benefits of containerization in terms of ease of deployment and scalability. The choice between these environments should be guided by specific application requirements, including performance needs, deployment flexibility, and resource utilization efficiency.

2. Configuration 1: 3 GB RAM with 3 Cores

QEMU Setup (disk image: qcow2):



```
Last login: Tue Jan 30 22:10:37 on ttys002
[(base) sumit@Rishabh ~ % cd Desktop/Rishabh/qemu
(base) sumit@Rishabh qemu % /opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 3072 -smp 3 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=ps
flash,format=raw,readonly=on \
-drive if=none,file=ubuntu.img,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 ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

QEMU Setup (disk image: raw):



```
Last login: Wed Jan 31 03:12:21 on ttys000
[(base) sumit@Rishabh ~ % cd Desktop/Rishabh/qemu
(base) sumit@Rishabh qemu % /opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 3072 -smp 3 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=ps
flash,format=raw,readonly=on \
-drive if=none,file=ubuntu.img,format=raw,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 ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

Docker Setup:

Settings [Give feedback](#)

General Resources Advanced

Resources

CPUs: 3

Memory: 3 GB

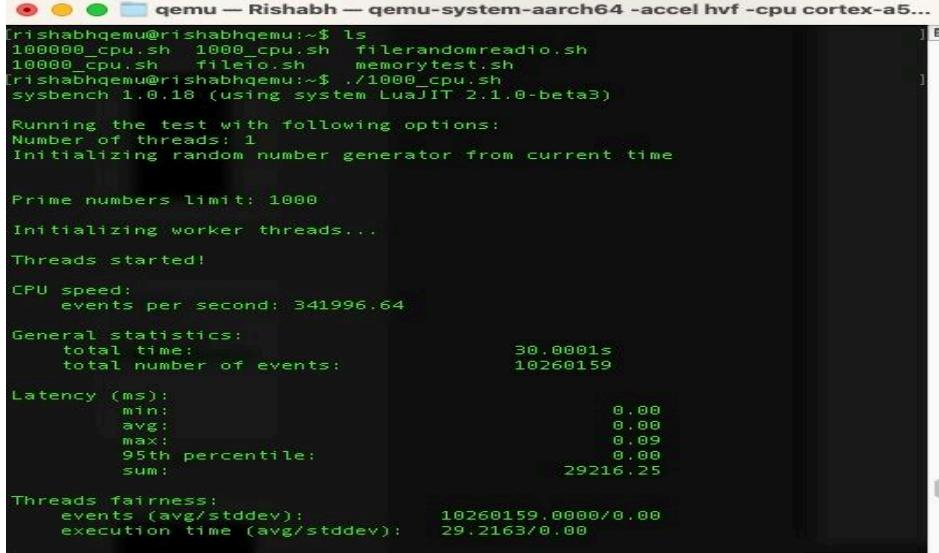
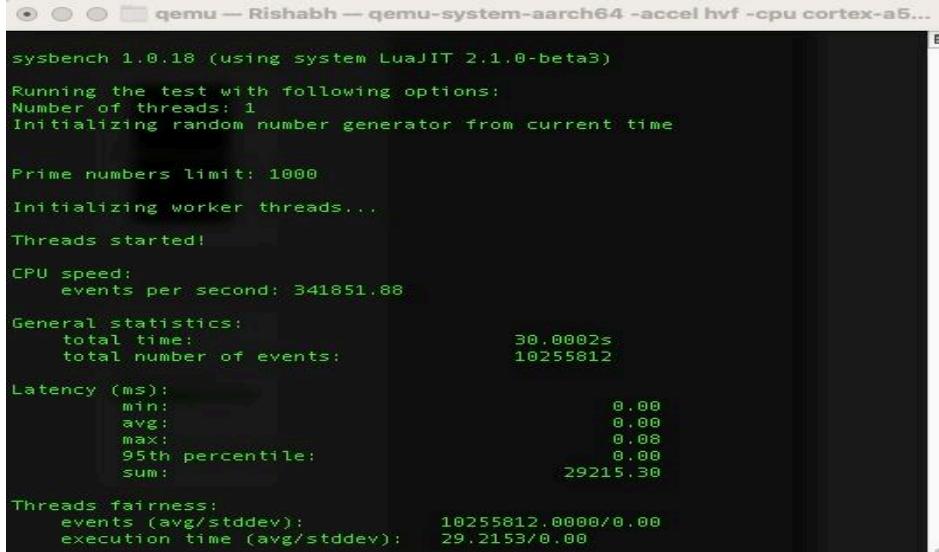
Swap: 1 GB

Virtual disk limit: 64 GB
Due to filesystem overhead, the real available space might be less.

```
[base] sumit@Rishabh ~ % docker run -it --memory="3g" --cpuset-cpus="0-2" -d 100cpudockerimage 1c89392db6a25fb859939ea150fddd61ace53d12d1450b6ea5c6dc5469176566
[base] sumit@Rishabh ~ % docker ps
CONTAINER ID IMAGE COMMAND NAMES CREATED STATUS PORTS
1c89392db6a2 100cpudockerimage "/./1000_cpu.sh" lucid_elion 3 seconds ago Up 3 seconds
8e4f0b8275f2 kindest/node:v1.27.3 "/usr/local/bin/entr..." 3 months ago Up 2 minutes 0.0.0.0:30080->30080/tcp, 127.0.0.1:65182->6443/tcp kind-control-plane
[base] sumit@Rishabh ~ % docker inspect lucid_elion
[
  {
    "Id": "1c89392db6a25fb859939ea150fddd61ace53d12d1450b6ea5c6dc5469176566",
    "Created": "2024-01-31T10:02:09.402561589Z",
    "Path": "./1000_cpu.sh",
    "Args": [],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 3852,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2024-01-31T10:02:09.622762714Z",
      "FinishedAt": "2001-01-01T00:00:00Z"
    },
    "Config": {
      "Image": "100cpudockerimage",
      "Cmd": [
        "/./1000_cpu.sh"
      ],
      "ExposedPorts": {},
      "Labels": {},
      "Env": [
        "PATH=/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin"
      ],
      "Mounts": [
        {
          "Type": "bind",
          "Source": "/var/run/docker.sock",
          "Destination": "/var/run/docker.sock",
          "Mode": "ro"
        }
      ],
      "HostConfig": {
        "Binds": [
          "/var/run/docker.sock:/var/run/docker.sock"
        ],
        "CpuShares": 48,
        "CpuPeriod": 101,
        "CpuQuota": null,
        "CpuRealtimePeriod": 0,
        "CpuRealtimeRuntime": 0,
        "CpusetCpus": "0-2",
        "CpusetMems": "",
        "Devices": [],
        "DeviceGroupRules": null,
        "DeviceRequests": null,
        "Dns": [],
        "DnsOptions": [],
        "DnsSearch": [],
        "ExtraHosts": null,
        "GroupAdd": null,
        "IpcMode": "private",
        "Cgroup": "",
        "Links": null,
        "LogConfig": {
          "Type": "json-file",
          "Config": {}
        },
        "PidMode": "host",
        "Privileged": false,
        "PublishAllPorts": false,
        "ReadonlyRootfs": false,
        "SecurityOpt": null,
        "UTSMode": "",
        "UserNamespaceMode": ""
      }
    },
    "NetworkSettings": {
      "Bridge": "bridge",
      "ContainerId": "1c89392db6a25fb859939ea150fddd61ace53d12d1450b6ea5c6dc5469176566",
      "GlobalIPv6": false,
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:22:25:47:22",
      "NetworkMode": "bridge",
      "Ports": {
        "30080/tcp": {
          "HostIp": "127.0.0.1",
          "HostPort": 30080
        }
      },
      "SecondaryIPs": [],
      "SecondaryIPv6s": []
    }
  }
]
```

2.1 CPU Testing

QEMU Results for 3 GB 3 cores for max-prime = 1000 and time = 30 seconds, disk image = qcow2

Iteration No	Screenshot
1	 <pre> rishabhqemu@rishabhqemu:~\$ ls 100000_cpu.sh 1000_cpu.sh filerandomradio.sh 10000_cpu.sh fileIo.sh memorytest.sh rishabhqemu@rishabhqemu:~\$./1000_cpu.sh 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: 1000 Initializing worker threads... Threads started! CPU speed: events per second: 341996.64 General statistics: total time: 30.0001s total number of events: 10260159 Latency (ms): min: 0.00 avg: 0.00 max: 0.09 95th percentile: 0.00 sum: 29216.25 Threads fairness: events (avg/stddev): 10260159.0000/0.00 execution time (avg/stddev): 29.2163/0.00 </pre>
2	 <pre> 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: 1000 Initializing worker threads... Threads started! CPU speed: events per second: 341851.88 General statistics: total time: 30.0002s total number of events: 10255812 Latency (ms): min: 0.00 avg: 0.00 max: 0.08 95th percentile: 0.00 sum: 29215.30 Threads fairness: events (avg/stddev): 10255812.0000/0.00 execution time (avg/stddev): 29.2153/0.00 </pre>

3

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5...
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: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 342890.10

General statistics:
total time: 30.0001s
total number of events: 10286943

Latency (ms):
min: 0.00
avg: 0.00
max: 0.16
95th percentile: 0.00
sum: 29215.82

Threads fairness:
events (avg/stddev): 10286943.0000/0.00
execution time (avg/stddev): 29.2158/0.00
```

4

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5...
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: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 340542.47

General statistics:
total time: 30.0001s
total number of events: 10216505

Latency (ms):
min: 0.00
avg: 0.00
max: 0.09
95th percentile: 0.00
sum: 29222.35

Threads fairness:
events (avg/stddev): 10216505.0000/0.00
execution time (avg/stddev): 29.2224/0.00
```

5

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5...
events (avg/stddev): 10216505.0000/0.00
execution time (avg/stddev): 29.2224/0.00

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: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 340605.49

General statistics:
total time: 30.0001s
total number of events: 10218393

Latency (ms):
min: 0.00
avg: 0.00
max: 0.08
95th percentile: 0.00
sum: 29217.40

Threads fairness:
events (avg/stddev): 10218393.0000/0.00
execution time (avg/stddev): 29.2174/0.00

rishabh@qemu:~$ []
```

Detailed report:

Iteration No	Events per second
1	341996.64
2	341851.88
3	342890.10
4	340542.47
5	340605.49

Observations:

Average events per second	341577.32
Minimum Number of events per second recorded	340542.47
Maximum Number of events per second recorded	342890.10

QEMU Results, Configuration: 3 GB 3 cores for max-prime = 1000 and time = 30 seconds, disk image = raw

Screenshot

```
[rishabh@rishabhqemu:~/]$ ./1000cpu.sh
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: 1000

Initializing worker threads...
Threads started!

CPU speed:
events per second: 207631.67

General statistics:
total time: 30.0001s
total number of events: 6229150

Latency (ms):
min: 0.00
avg: 0.00
max: 2.25
95th percentile: 0.00
sum: 29207.62

Threads fairness:
events (avg/stddev): 6229150.0000/0.00
execution time (avg/stddev): 29.2076/0.00
```

Detailed report:

Iteration No	Events per second
1	207631.67
2	208456.83
3	207344.92
4	207493.40
5	208945.25

Observations:

Average events per second	207974.41
Minimum Number of events per second recorded	207344.92
Maximum Number of events per second recorded	208945.25

Docker results for 3 GB 3 cores for max-prime = 1000 and time = 30 seconds

Screenshot

```
[(base) sumit@Rishabh ~ % docker logs -f lucid_elion
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: 1000
Initializing worker threads...
Threads started!
CPU speed:
  events per second: 206335.18

General statistics:
  total time:          30.00001s
  total number of events: 6190266

Latency (ms):
  min:                  0.00
  avg:                  0.00
  max:                  0.93
  95th percentile:      0.00
  sum:                 29210.71

Threads fairness:
  events (avg/stddev): 6190266.0000/0.00
  execution time (avg/stddev): 29.2107/0.00
```

Detailed report:

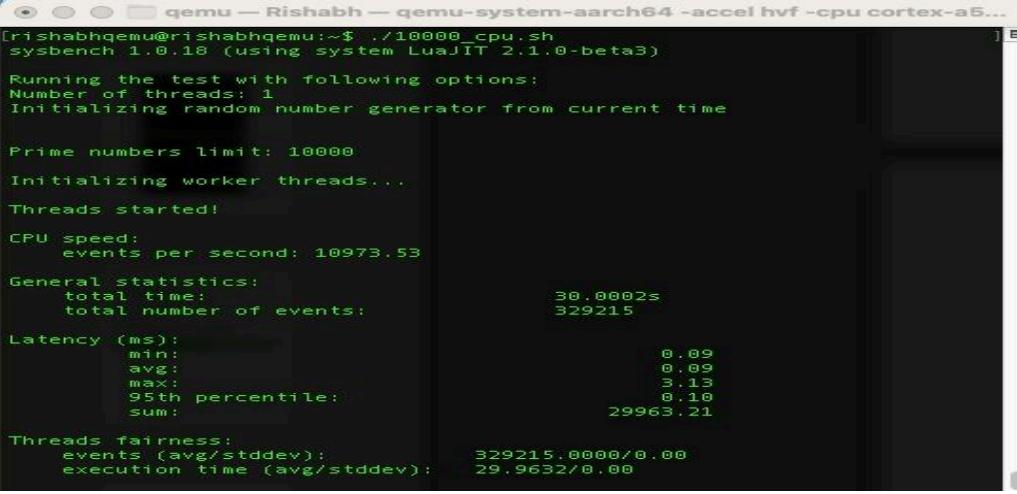
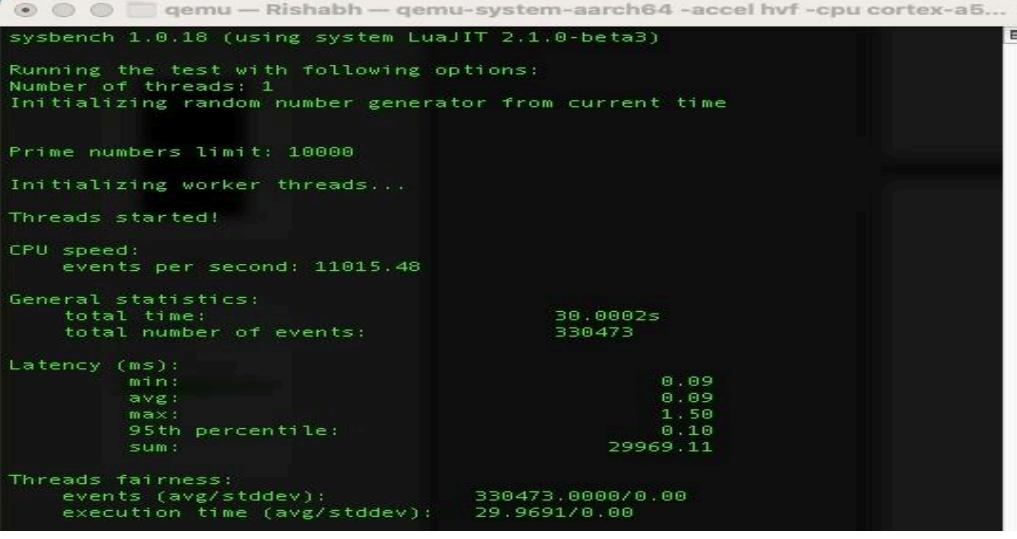
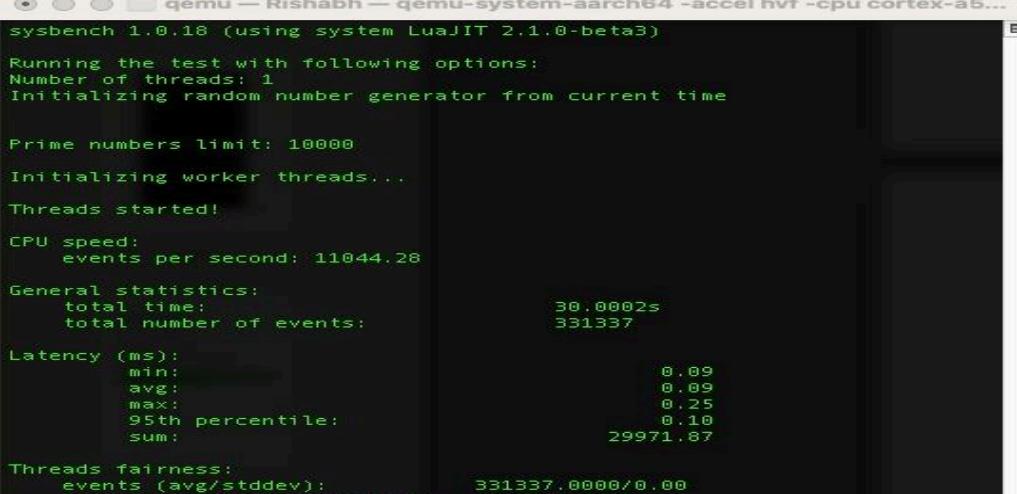
Iteration No	Events per second
1	206335.18
2	207103.78
3	206912.47
4	207028.30
5	206456.67

Observations:

Average events per second	206767.28
Minimum Number of events per second recorded	206335.18
Maximum Number of events per second recorded	207103.78

Conclusion: The comparison highlights the superior efficiency of QEMU with a qcow2 disk image for intensive computational tasks, such as calculating prime numbers (max-prime = 1000), within a specified timeframe (30 seconds). This setup outperforms both the raw disk image configuration and Docker in terms of events processed per second. While QEMU with a raw disk image and Docker provide substantial computational throughput, they are less optimized for this specific task compared to qcow2. The choice between these options should be informed by specific performance requirements, workload characteristics, and the operational benefits of containerization offered by Docker.

QEMU Results for 3 GB 3 cores for max-prime = 10000 and time = 30 seconds, disk image = qcow2

Iteration No	Screenshot
1	 <pre>[rishabhqemu@rishabhqemu:~/]\$./10000_cpu.sh 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: 10973.53 General statistics: total time: 30.0002s total number of events: 329215 Latency (ms): min: 0.09 avg: 0.09 max: 3.13 95th percentile: 0.10 sum: 29963.21 Threads fairness: events (avg/stddev): 329215.0000/0.00 execution time (avg/stddev): 29.9632/0.00</pre>
2	 <pre>[rishabhqemu@rishabhqemu:~/]\$./10000_cpu.sh 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: 11015.48 General statistics: total time: 30.0002s total number of events: 330473 Latency (ms): min: 0.09 avg: 0.09 max: 1.50 95th percentile: 0.10 sum: 29969.11 Threads fairness: events (avg/stddev): 330473.0000/0.00 execution time (avg/stddev): 29.9691/0.00</pre>
3	 <pre>[rishabhqemu@rishabhqemu:~/]\$./10000_cpu.sh 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: 11044.28 General statistics: total time: 30.0002s total number of events: 331337 Latency (ms): min: 0.09 avg: 0.09 max: 0.25 95th percentile: 0.10 sum: 29971.87 Threads fairness: events (avg/stddev): 331337.0000/0.00 execution time (avg/stddev): 29.9719/0.00</pre>

4

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5...
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: 11020.71

General statistics:
total time: 30.0002s
total number of events: 330629

Latency (ms):
min: 0.09
avg: 0.09
max: 0.60
95th percentile: 0.09
sum: 29963.83

Threads fairness:
events (avg/stddev): 330629.0000/0.00
execution time (avg/stddev): 29.9638/0.00
```

5

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5...
events (avg/stddev): 330629.0000/0.00
execution time (avg/stddev): 29.9638/0.00
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: 10952.94

General statistics:
total time: 30.0002s
total number of events: 328597

Latency (ms):
min: 0.09
avg: 0.09
max: 0.56
95th percentile: 0.10
sum: 29966.09

Threads fairness:
events (avg/stddev): 328597.0000/0.00
execution time (avg/stddev): 29.9661/0.00
rishabh@rishabh-qemu:~$
```

Detailed report:

Iteration No	Events per second
1	10973.53
2	11015.48
3	11044.28
4	11020.71
5	10952.94

Observations:

Average events per second	11001.39
Minimum Number of events per second recorded	10952.94
Maximum Number of events per second recorded	11044.28

QEMU Results, Configuration: 3 GB 3 cores for max-prime = 10000 and time = 30 seconds, disk image = raw

Screenshot

```
[rishabh@rishabh-qemu:~$ ./10000cpu.sh
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: 6724.72

General statistics:
total time: 30.0002s
total number of events: 201750

Latency (ms):
min: 0.15
avg: 0.15
max: 2.31
95th percentile: 0.15
sum: 29952.96

Threads fairness:
events (avg/stddev): 201750.0000/0.00
execution time (avg/stddev): 29.9530/0.00
```

Detailed report:

Iteration No	Events per second
1	6724.72
2	6983.43
3	6934.32
4	6745.83
5	6733.96

Observations:

Average events per second	6824.45
Minimum Number of events per second recorded	6724.72
Maximum Number of events per second recorded	6983.43

Docker Results for 3 GB 3 cores for max-prime = 10000 and time = 30 seconds

Screenshot

```
[base] sumit@Rishabh: ~ % docker logs -f kind_vaughan
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: 6675.03

General statistics:
total time: 30.00003s
total number of events: 200259

Latency (ms):
min: 0.15
avg: 0.15
max: 1.56
95th percentile: 0.16
sum: 29938.54

Threads fairness:
events (avg/stddev): 200259.0000/0.00
execution time (avg/stddev): 29.9385/0.00
```

Detailed report:

Iteration No	Events per second
1	6675.03
2	6633.63
3	6612.98
4	6069.18
5	5683.64

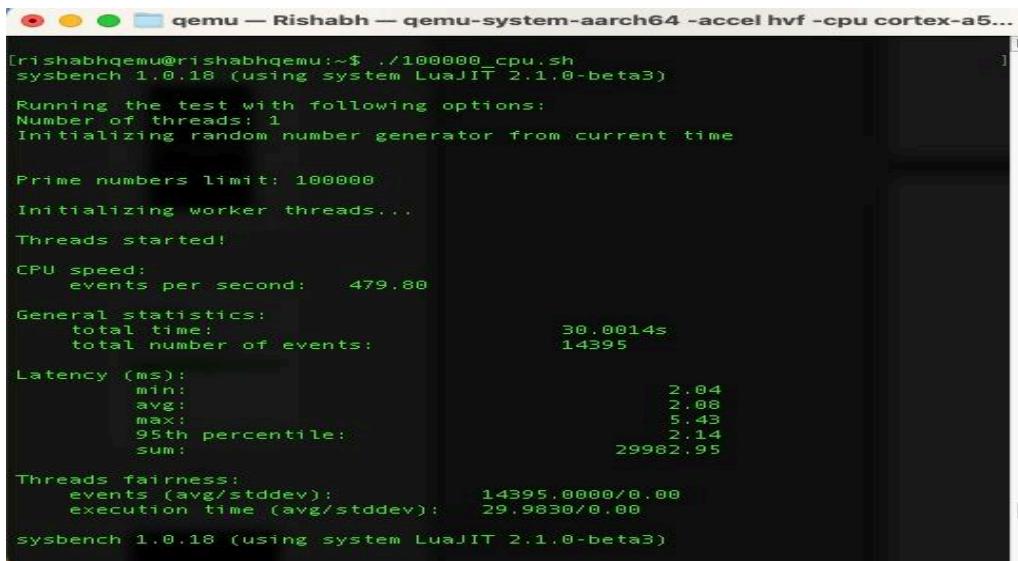
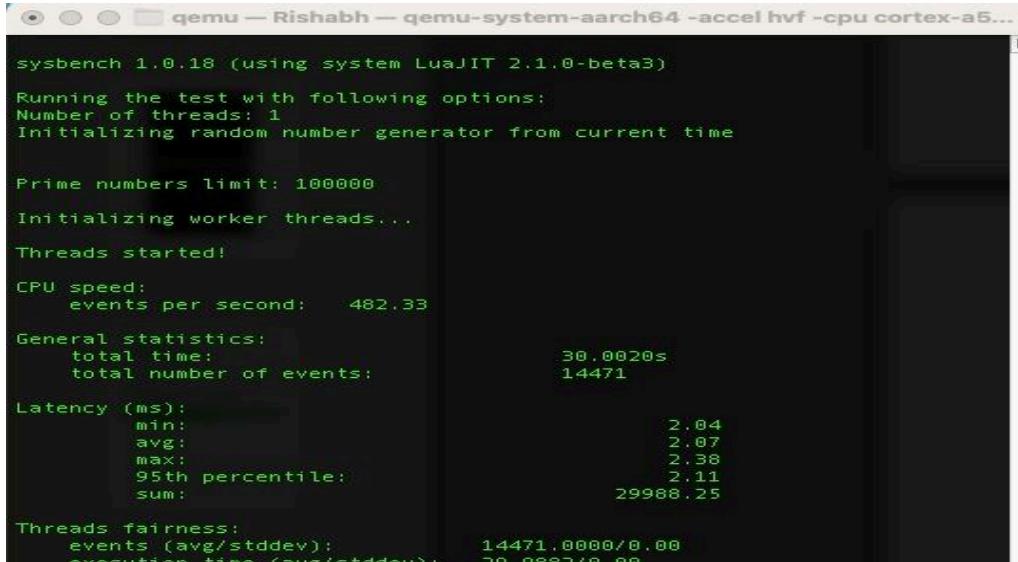
Observations:

Average events per second	6334.89
Minimum Number of events per second recorded	5683.64
Maximum Number of events per second recorded	6675.03

Conclusion: For the task of calculating prime numbers up to 10,000 within a 30-second timeframe on a 3 GB RAM and 3 core configuration, QEMU with a qcow2 disk image significantly outperforms both QEMU with a raw disk image and Docker. With an average of 11,001.39 events per second, qcow2 demonstrates superior computational efficiency. In contrast, QEMU with a raw disk image and Docker achieve lower averages of 6,824.45 and 6,334.89 events per second, respectively.

This performance gap underscores the effectiveness of the qcow2 format in optimizing resource utilization and computational throughput for intensive tasks.

QEMU Results for 3 GB 3 cores for max-prime = 100000 and time = 30 seconds, disk image = qcow2

Iteration No	Screenshot
1	 <pre>[rishabh@rishabh:~/]\$./100000_cpu.sh 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: 100000 Initializing worker threads... Threads started! CPU speed: events per second: 479.80 General statistics: total time: 30.0014s total number of events: 14395 Latency (ms): min: 2.04 avg: 2.08 max: 5.43 95th percentile: 2.14 sum: 29982.95 Threads fairness: events (avg/stddev): 14395.0000/0.00 execution time (avg/stddev): 29.9830/0.00 sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)</pre>
2	 <pre>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: 100000 Initializing worker threads... Threads started! CPU speed: events per second: 482.33 General statistics: total time: 30.0020s total number of events: 14471 Latency (ms): min: 2.04 avg: 2.07 max: 2.38 95th percentile: 2.11 sum: 29988.25 Threads fairness: events (avg/stddev): 14471.0000/0.00 execution time (avg/stddev): 29.9882/0.00</pre>

3

```
qemu -- Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5... E

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: 100000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 480.22
General statistics:
total time: 30.0021s
total number of events: 14408
Latency (ms):
min: 2.03
avg: 2.08
max: 2.45
95th percentile: 2.14
sum: 29988.29
Threads fairness:
events (avg/stddev): 14408.0000/0.00
execution time (avg/stddev): 29.9883/0.00
```

4

```
qemu -- Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5... E

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: 100000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 480.98
General statistics:
total time: 30.0004s
total number of events: 14430
Latency (ms):
min: 2.04
avg: 2.08
max: 2.40
95th percentile: 2.11
sum: 29983.90
Threads fairness:
events (avg/stddev): 14430.0000/0.00
execution time (avg/stddev): 29.9839/0.00
```

5

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5... E

events (avg/stddev): 14430.0000/0.00
execution time (avg/stddev): 29.9839/0.00
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: 100000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 479.70
General statistics:
total time: 30.0016s
total number of events: 14392
Latency (ms):
min: 2.04
avg: 2.08
max: 3.16
95th percentile: 2.14
sum: 29987.84
Threads fairness:
events (avg/stddev): 14392.0000/0.00
execution time (avg/stddev): 29.9878/0.00
rishabh@rishabh:~$
```

Detailed report:

Iteration No	Events per second
1	479.80
2	482.33
3	480.22
4	480.98
5	479.70

Observations:

Average events per second	480.61
Minimum Number of events per second recorded	479.70
Maximum Number of events per second recorded	482.33

QEMU Results, Configuration: 3 GB 3 cores for max-prime = 100000 and time = 30 seconds, disk image = raw

Screenshot

```
[rishabhqemu@rishabhqemu:~/]$ ./100000cpu.sh
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: 100000
Initializing worker threads...
Threads started!

CPU speed:
    events per second: 292.51

General statistics:
    total time:          30.00016s
    total number of events: 8776

Latency (ms):
    min:                  3.35
    avg:                  3.41
    max:                 13.12
    95th percentile:     3.43
    sum:                29966.54

Threads fairness:
    events (avg/stddev): 8776.0000/0.00
    execution time (avg/stddev): 29.9665/0.00

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

Detailed report:

Iteration No	Events per second
1	292.51
2	290.45
3	289.39
4	291.30
5	292.72

Observations:

Average events per second	291.27
Minimum Number of events per second recorded	289.39
Maximum Number of events per second recorded	292.72

Docker Results for 3 GB 3 cores for max-prime = 100000 and time = 30 seconds

Screenshot

```
[base] sumit@Rishabh ~ % docker logs -f intelligent_ritchie
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: 100000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 287.53

General statistics:
total time: 30.0027s
total number of events: 8627

Latency (ms):
min: 3.35
avg: 3.47
max: 14.98
95th percentile: 3.89
sum: 29973.53

Threads fairness:
events (avg/stddev): 8627.0000/0.00
execution time (avg/stddev): 29.9735/0.00
```

Detailed report:

Iteration No	Events per second
1	287.53
2	290.61
3	253.55
4	263.73
5	274.78

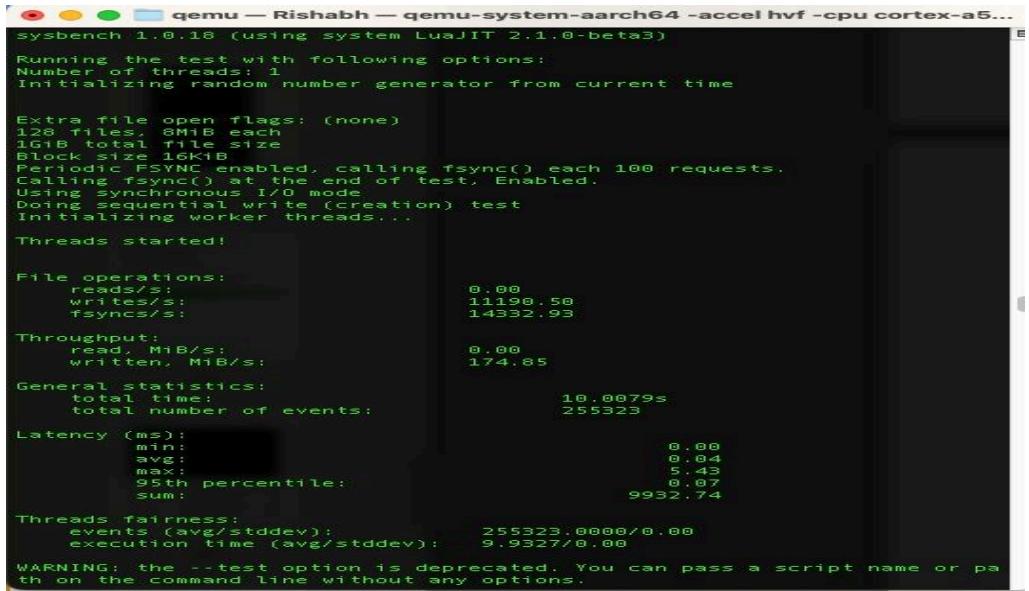
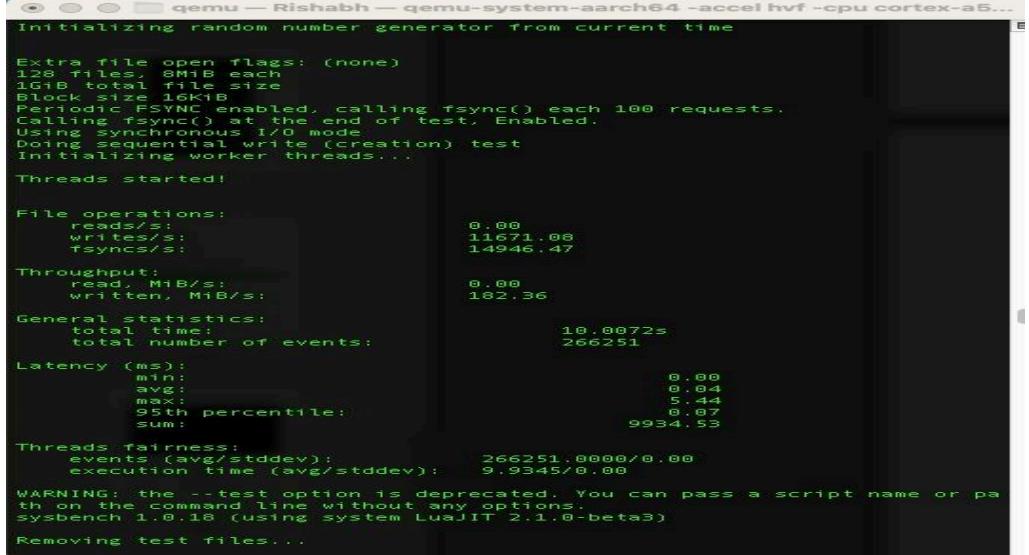
Observations:

Average events per second	274.04
Minimum Number of events per second recorded	253.55
Maximum Number of events per second recorded	290.61

Conclusion: When tasked with calculating prime numbers up to 100,000 within a 30-second interval on a system with 3 GB RAM and 3 cores, the performance across QEMU with qcow2, QEMU with a raw disk image, and Docker varied significantly. QEMU with a qcow2 disk image leads in efficiency, with an average event rate of 480.61 per second, showcasing its superior capability for handling intensive computational tasks. QEMU with a raw disk image follows, presenting a lower average of 291.27 events per second. Docker exhibits the least performance, with an average of 274.04 events per second, and also records the lowest minimum event rate among the configurations tested. This comparison highlights the qcow2 disk image's optimization and effectiveness for demanding computational operations within QEMU environments.

2.2 File I/O Testing

QEMU Results for 3 GB 3 cores for sequential write, disk image = qcow2

Iteration No	Screenshots
1	 <pre>qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a5... 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 Extra file open flags: (none) 128 files, 8MB each 1GiB total file size Block size 16KiB Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 11190.50 fsyncs/s: 14532.93 Throughput: read, MiB/s: 0.00 written, MiB/s: 174.85 General statistics: total time: 18.00795 total number of events: 255323 Latency (ms): min: 0.00 avg: 0.04 max: 5.43 95th percentile: 0.07 sum: 9932.74 Threads fairness: events (avg/stddev): 255323.0000/0.00 execution time (avg/stddev): 9.9327/0.00 WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.</pre>
2	 <pre>qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a5... Initialzing random number generator from current time Extra file open flags: (none) 128 files, 8MB each 1GiB total file size Block size 16KiB Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initialzing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 11671.08 fsyncs/s: 14946.47 Throughput: read, MiB/s: 0.00 written, MiB/s: 182.36 General statistics: total time: 18.00725 total number of events: 266251 Latency (ms): min: 0.00 avg: 0.04 max: 5.44 95th percentile: 0.07 sum: 9934.53 Threads fairness: events (avg/stddev): 266251.0000/0.00 execution time (avg/stddev): 9.9345/0.00 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) Removing test files...</pre>

3

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a6...
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test. Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        12200.88
  fsyncs/s:        15619.02

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:  190.64

General statistics:
  total time:           10.0069s
  total number of events: 278279

Latency (ms):
  min:                  0.00
  avg:                 0.04
  max:                 5.97
  95th percentile:     0.06
  sum:                9935.57

Threads fairness:
  events (avg/stddev): 278279.0000/0.00
  execution time (avg/stddev): 9.9356/0.00

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)

Removing test files...
```

4

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a6...
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test. Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        11615.34
  fsyncs/s:        14877.93

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:  181.49

General statistics:
  total time:           10.0035s
  total number of events: 264911

Latency (ms):
  min:                  0.00
  avg:                 0.04
  max:                 5.97
  95th percentile:     0.06
  sum:                9935.52

Threads fairness:
  events (avg/stddev): 264911.0000/0.00
  execution time (avg/stddev): 9.9359/0.00

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)

Removing test files...
```

5

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a6...
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test. Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        11380.65
  fsyncs/s:        14570.33

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:  177.82

General statistics:
  total time:           10.0076s
  total number of events: 259595

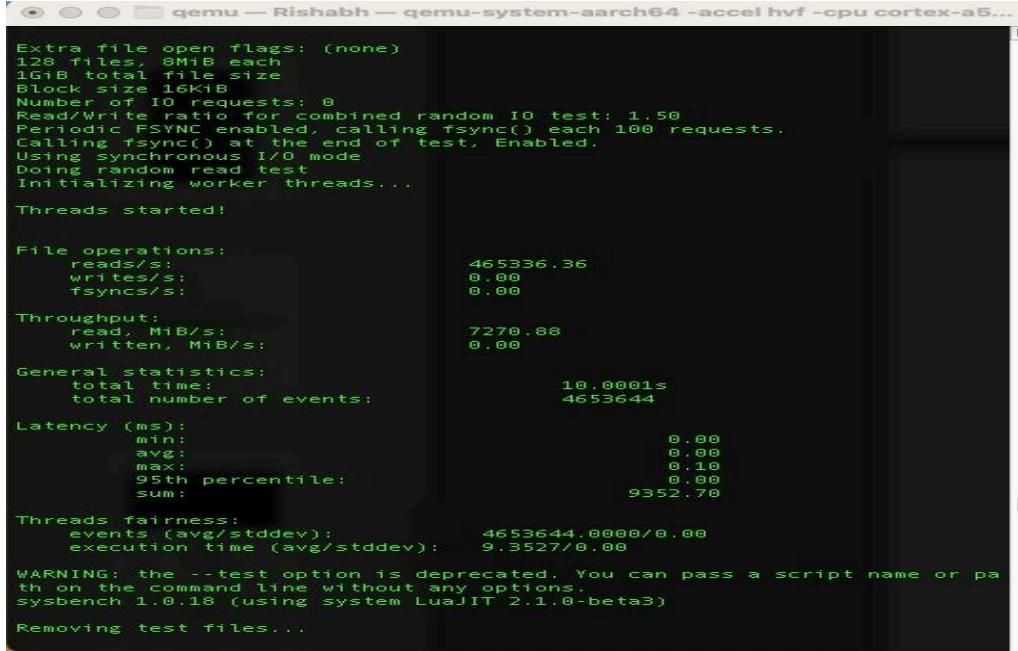
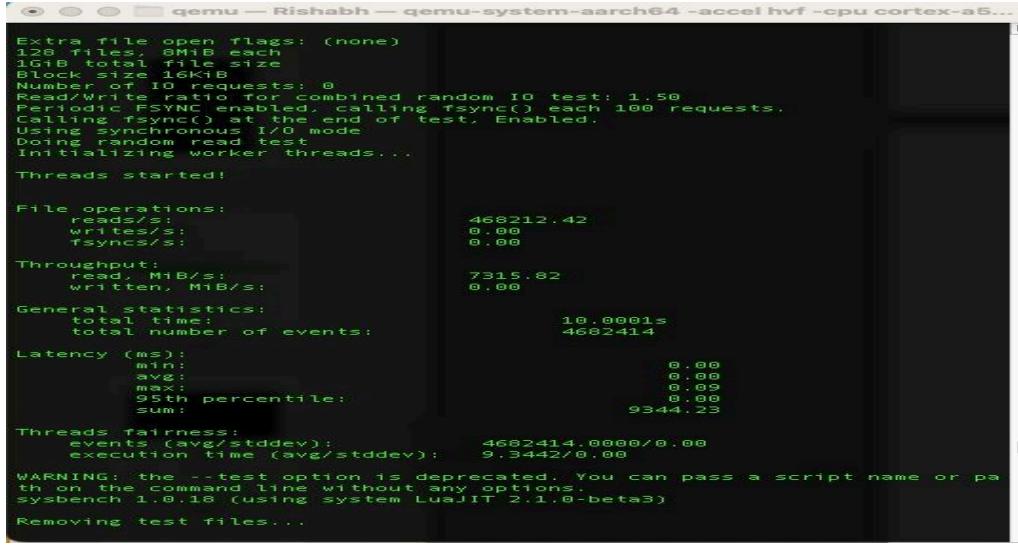
Latency (ms):
  min:                  0.00
  avg:                 0.04
  max:                 5.84
  95th percentile:     0.07
  sum:                9934.19

Threads fairness:
  events (avg/stddev): 259595.0000/0.00
  execution time (avg/stddev): 9.9342/0.00

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)

Removing test files...
```

QEMU Results for 3 GB 3 cores for random read, disk image = qcow2

Iteration No	Screenshots
1	 <pre> Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Number of IO requests: 8 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 read test Initializing worker threads... Threads started! File operations: reads/s: 465336.36 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 7270.88 written, MiB/s: 0.00 General statistics: total time: 10.0001s total number of events: 4653644 Latency (ms): min: 0.00 avg: 0.00 max: 0.10 95th percentile: 0.00 sum: 9352.70 Threads fairness: events (avg/stddev): 4653644.0000/0.00 execution time (avg/stddev): 9.3527/0.00 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) Removing test files... </pre>
2	 <pre> Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Number of IO requests: 8 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 read test Initializing worker threads... Threads started! File operations: reads/s: 468212.42 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 7315.82 written, MiB/s: 0.00 General statistics: total time: 10.0001s total number of events: 4682414 Latency (ms): min: 0.00 avg: 0.00 max: 0.09 95th percentile: 0.00 sum: 9344.23 Threads fairness: events (avg/stddev): 4682414.0000/0.00 execution time (avg/stddev): 9.3442/0.00 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) Removing test files... </pre>

3

```

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Number of I/O 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 read test
Initializing worker threads...

Threads started!

File operations:
  reads/s:                                470096.42
  writes/s:                               0.00
  fsyncs/s:                               0.00

Throughput:
  read, MiB/s:                            7345.26
  written, MiB/s:                           0.00

General statistics:
  total time:                             10.0001s
  total number of events:                 4701244

Latency (ms):
  min:                                     0.00
  avg:                                     0.00
  max:                                     0.09
  95th percentile:                         0.00
  sum:                                    9347.94

Threads fairness:
  events (avg/stddev):      4701244.0000/0.00
  execution time (avg/stddev): 9.3479/0.00

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)

Removing test files...

```

4

```

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Number of I/O 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 read test
Initializing worker threads...

Threads started!

File operations:
  reads/s:                                465217.64
  writes/s:                               0.00
  fsyncs/s:                               0.00

Throughput:
  read, MiB/s:                            7269.03
  written, MiB/s:                           0.00

General statistics:
  total time:                             10.0001s
  total number of events:                 4652454

Latency (ms):
  min:                                     0.00
  avg:                                     0.00
  max:                                     0.11
  95th percentile:                         0.00
  sum:                                    9353.36

Threads fairness:
  events (avg/stddev):      4652454.0000/0.00
  execution time (avg/stddev): 9.3534/0.00

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)

Removing test files...

```

5

```

128 files, 8MiB each
1GiB total file size
Block size 16KiB
Number of I/O 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 read test
Initializing worker threads...

Threads started!

File operations:
  reads/s:                                460277.85
  writes/s:                               0.00
  fsyncs/s:                               0.00

Throughput:
  read, MiB/s:                            7191.83
  written, MiB/s:                           0.00

General statistics:
  total time:                             10.0001s
  total number of events:                 4603036

Latency (ms):
  min:                                     0.00
  avg:                                     0.00
  max:                                     0.13
  95th percentile:                         0.00
  sum:                                    9342.35

Threads fairness:
  events (avg/stddev):      4603036.0000/0.00
  execution time (avg/stddev): 9.3423/0.00

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)

Removing test files...
File I/O tests completed.
rishabh@qemu:~$ []

```

**QEMU Results for 3 GB 3 cores for sequential write & random read,
disk image = qcow2**

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 11190.50 fsyncs/s: 14332.93	reads/s: 465336.36 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 11671.08 fsyncs/s: 14946.47	reads/s: 468212.42 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 12200.88 fsyncs/s: 15619.02	reads/s: 470096.42 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 11615.34 fsyncs/s: 14877.93	reads/s: 465217.64 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 11380.65 fsyncs/s: 14570.33	reads/s: 460277.85 writes/s: 0.00 fsyncs/s: 0.00

**QEMU Results for 3 GB 3 cores for sequential write & random read,
disk image = raw**

Screenshot

```

qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi...
Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test. Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        7170.39
  fsyncs/s:        9179.09

Throughput:
  read, MiB/s:      0.00
  written, MiB/s: 112.04

General statistics:
  total time:           10.0124s
  total number of events: 163586

Latency (ms):
  min:                  0.00
  avg:                  0.06
  max:                  5.65
  95th percentile:     0.10
  sum:                 9842.03

Threads fairness:
  events (avg/stddev): 163586.0000/0.00
  execution time (avg/stddev): 9.8420/0.00

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)

Removing test files...

```

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 7178.39 fsyncs/s: 9179.89	reads/s: 433256.87 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 7456.34 fsyncs/s: 9893.45	reads/s: 423849.94 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 7539.67 fsyncs/s: 9435.34	reads/s: 424539.32 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 7129.45 fsyncs/s: 9138.94	reads/s: 438732.34 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 7243.73 fsyncs/s: 9256.89	reads/s: 424324.56 writes/s: 0.00 fsyncs/s: 0.00

Docker Results for 3 GB 3 cores for sequential write & random read

```

sumit — Rishabh — zsh — 88x43
Extra file open flags: (none)
128 files, 8MB each
1GB total file size
Block size 16kB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test. Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          0.00
  writes/s:        11671.21
  fsyncs/s:       14942.75

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   182.36

General statistics:
  total time:           10.00065s
  total number of events: 266212

Latency (ms):
  min:                  0.00
  avg:                  0.04
  max:                35.78
  95th percentile:     0.05
  sum:                9878.66

Threads fairness:
  events (avg/stddev): 266212.0000/0.00
  execution time (avg/stddev): 9.8787/0.00

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)

Removing test files...

```

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 11671.21 fsyncs/s: 14942.75	reads/s: 232879.36 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 11020.46 fsyncs/s: 14109.68	reads/s: 245704.49 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00	reads/s: 278675.26

	writes/s: 13107.54 fsyncs/s: 16787.24	writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 11934.05 fsyncs/s: 15286.18	reads/s: 234659.03 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 12932.90 fsyncs/s: 16566.60	reads/s: 291983.36 writes/s: 0.00 fsyncs/s: 0.00

Conclusion: Upon comparing the performance metrics across different experiment configurations, we observe notable differences in performance between using QEMU with qcow2 and raw disk images, and Docker for sequential write and random read operations. Specifically, QEMU with qcow2 disk images shows higher write and fsyncs per second, indicating better sequential write performance compared to both QEMU with raw disk images and Docker. For random read operations, QEMU with either disk image significantly outperforms Docker, showcasing much higher reads per second. These results indicate that while the performance characteristics are distinct across different setups, QEMU with qcow2 disk images offers the most robust performance for the tested operations.

2.3 Memory Testing

QEMU Results for 3 GB 3 cores for Sequential Access, disk image = qcow2	
Iteration	Results
1	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5... Rishabh@qemu:~/Rishabh\$./memorytest.sh Running First Memory Test: Sequential Access Iteration 1 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 86332122 (8632484.20 per second) 84308.71 MiB transferred (8430.16 MiB/sec) General statistics: total time: 10.0001s total number of events: 86332122 Latency (ms): min: 0.00 avg: 0.00 max: 0.00 95th percentile: 0.00 sum: 4194.09 Threads fairness: events (avg/stddev): 86332122.0000/0.00 execution time (avg/stddev): 4.1941/0.00</pre>

2

```
Iteration 2
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 86553357 (8654611.23 per second)
84524.76 MiB transferred (8451.77 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 86553357

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 0.00
  95th percentile: 0.00
  sum: 4184.79

Threads fairness:
  events (avg/stddev): 86553357.0000/0.00
  execution time (avg/stddev): 4.1848/0.00
```

3

```
Iteration 3
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 86738077 (8673877.41 per second)
84705.15 MiB transferred (8469.88 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 86738077

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 0.00
  95th percentile: 0.00
  sum: 4192.65

Threads fairness:
  events (avg/stddev): 86738077.0000/0.00
  execution time (avg/stddev): 4.1926/0.00
```

4

```
Iteration 4
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 86729861 (8672196.83 per second)
84697.13 MiB transferred (8468.94 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 86729861

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 0.07
  95th percentile: 0.00
  sum: 4189.48

Threads fairness:
  events (avg/stddev): 86729861.0000/0.00
  execution time (avg/stddev): 4.1895/0.00
```

5

```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5...
Iteration 5
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 86025893 (8601892.26 per second)
84009.66 MiB transferred (8400.29 MiB/sec)

General statistics:
  total time: 10.00001s
  total number of events: 86025893

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 0.30
  95th percentile: 0.00
  sum: 4192.71

Threads fairness:
  events (avg/stddev): 86025893.0000/0.00
  execution time (avg/stddev): 4.1927/0.00

```

QEMU Results for 3 GB 3 cores for Random Access, disk image = qcow2

Iteration	Results
1	<pre> qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a5... Running Second Memory Test: Random Access Iteration 1 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18693349 (1869179.63 per second) 18255.22 MiB transferred (1825.37 MiB/sec) General statistics: total time: 10.00002s total number of events: 18693349 Latency (ms): min: 0.00 avg: 0.00 max: 0.10 95th percentile: 0.00 sum: 8735.94 Threads fairness: events (avg/stddev): 18693349.0000/0.00 execution time (avg/stddev): 8.7359/0.00 </pre>

2	<pre> Iteration 2 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18704196 (1870260.28 per second) 18265.82 MiB transferred (1826.43 MiB/sec) General statistics: total time: 10.0001s total number of events: 18704196 Latency (ms): min: 0.00 avg: 0.00 max: 0.00 95th percentile: 0.00 sum: 8739.36 Threads fairness: events (avg/stddev): 18704196.0000/0.00 execution time (avg/stddev): 8.7394/0.00 </pre>
3	<pre> Iteration 3 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18709605 (1870820.78 per second) 18271.10 MiB transferred (1826.97 MiB/sec) General statistics: total time: 10.0001s total number of events: 18709605 Latency (ms): min: 0.00 avg: 0.00 max: 0.07 95th percentile: 0.00 sum: 8739.20 Threads fairness: events (avg/stddev): 18709605.0000/0.00 execution time (avg/stddev): 8.7392/0.00 </pre>
4	<pre> Iteration 4 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18659698 (1865815.73 per second) 18222.36 MiB transferred (1822.09 MiB/sec) General statistics: total time: 10.0002s total number of events: 18659698 Latency (ms): min: 0.00 avg: 0.00 max: 0.24 95th percentile: 0.00 sum: 8739.14 Threads fairness: events (avg/stddev): 18659698.0000/0.00 execution time (avg/stddev): 8.7391/0.00 </pre>

5

```

WARNING: the --test option is deprecated. You can pass a script name or pa
th 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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 18555738 (1855430.31 per second)
18120.84 MiB transferred (1811.94 MiB/sec)

General statistics:
  total time: 10.00001s
  total number of events: 18555738

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 6.16
  95th percentile: 0.00
  sum: 8736.79

Threads fairness:
  events (avg/stddev): 18555738.0000/0.00
  execution time (avg/stddev): 8.7368/0.00

Memory tests completed.
rishabhqemu@rishabhqemu:~$ 
```

QEMU Results for 3 GB 3 cores for sequential & random access, disk image = qcow2

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	8632484.20	1869179.63
2	8654611.23	1870260.28
3	8673077.41	1870820.78
4	8672196.83	1865815.73
5	8601892.26	1855430.31

QEMU Results for 3 GB 3 cores for sequential & random access, disk image = raw

Screenshot

```
rishabh@rishabh:~/Documents$ ./memorytest.sh
Running First Memory Test: Sequential Access
Iteration 1
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 52866884 (5286084.12 per second)
51627.82 MiB transferred (5162.19 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 52866884

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 0.41
  95th percentile: 0.00
  sum: 4190.92

Threads fairness:
  events (avg/stddev): 52866884.0000/0.00
  execution time (avg/stddev): 4.1909/0.00
```

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	5286084.12	1145357.48
2	5348733.39	1189233.84
3	5434820.38	1155879.32
4	5245983.95	1190453.56
5	5294539.76	1176342.30

Docker Results for 3 GB 3 cores for sequential & random access

```
sumit@Rishabh:~$ zsh -c "memory-speed-test -t 10s -b 1KiB -s 102400MiB -o write -s global"
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 48214813 (4820980.06 per second)
47084.78 MiB transferred (4707.99 MiB/sec)

General statistics:
  total time:          10.0002s
  total number of events: 48214813

Latency (ms):
  min:                  0.00
  avg:                  0.00
  max:                  6.04
  95th percentile:      0.00
  sum:                 4254.26

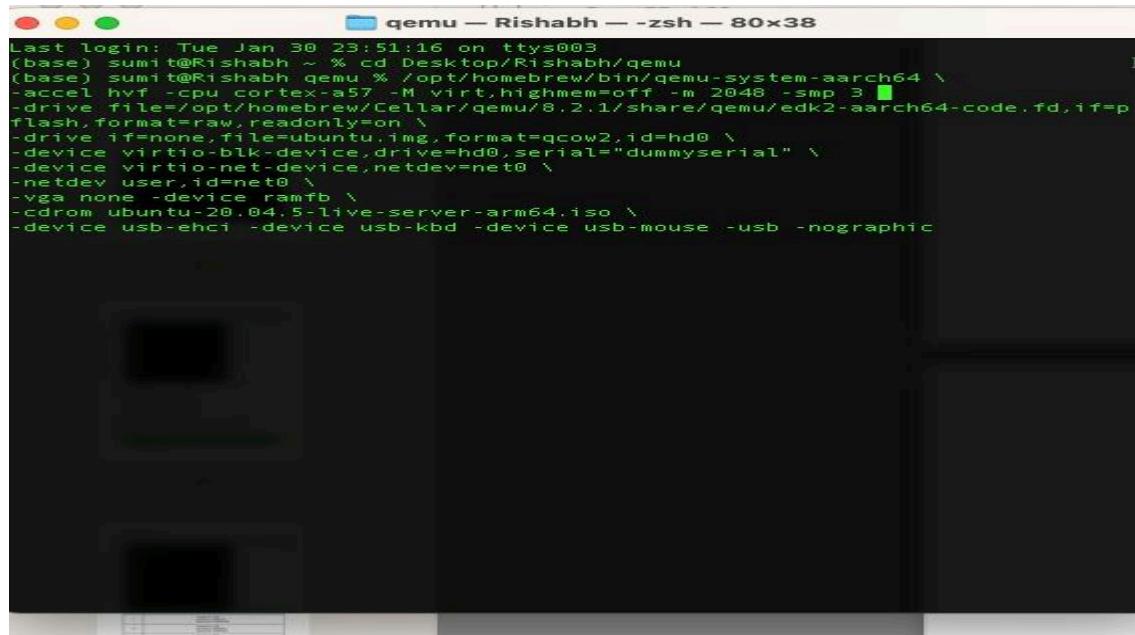
Threads fairness:
  events (avg/stddev): 48214813.0000/0.00
  execution time (avg/stddev): 4.2543/0.00
```

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	4820980.06	1080687.85
2	4799046.68	1109627.78
3	4850504.83	1125034.96
4	4786857.42	1126543.68
5	5089496.26	1125118.81

Conclusion: QEMU with a qcow2 disk image delivers the highest I/O performance for both sequential and random access among the configurations tested, significantly outperforming QEMU with a raw disk image and Docker. This demonstrates qcow2's superior optimization for intensive I/O tasks on a setup with 3 GB RAM and 3 cores.

3. Configuration 3: 2 GB RAM with 3 Cores

QEMU Setup (disk image: qcow2):



```
qemu — Rishabh — zsh — 80x38
Last login: Tue Jan 30 23:51:16 on ttys003
(base) sumit@Rishabh ~ % cd Desktop/Rishabh/qemu
(base) sumit@Rishabh qemu % /opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 3 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=ps3 \
flash,format=qcow2,readonly=on \
-drive if=none,file=ubuntu.img,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 ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

QEMU Setup (disk image: raw):



```
qemu — Rishabh — zsh — 80x45
Last login: Wed Jan 31 03:15:21 on ttys007
[(base) sumit@Rishabh ~ % cd Desktop/Rishabh/qemu
(base) sumit@Rishabh qemu % /opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 3 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=ps3 \
flash,format=raw,readonly=on \
-drive if=none,file=ubuntu.img,format=raw,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 ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

Docker Setup:

Settings [Give feedback](#)

General

Resources **Advanced**

CPUs: 3

Memory: 2 GB

Swap: 1 GB

Virtual disk limit: 64 GB
Due to filesystem overhead, the real available space might be less.

8 GR

[Cancel](#) [Apply & restart](#)

```
Last login: Wed Jan 31 02:07:05 on ttys005
(base) sumit@Rishabh ~ % docker run -it --memory="2g" --cpuset-cpus="0-2" -d 100cpudockerimage
d3bb93281b0d0bdf9a3c0d24658d6296b804e86d1c2a5938b38cf6c24b5c35b4
(base) sumit@Rishabh ~ % docker ps
CONTAINER ID IMAGE COMMAND CREATED NAMES
d3bb93281b0d 100cpudockerimage "/1000_cpu.sh" 9 seconds ago intelligent_hopper
bb1b7bf1c5c6 memorytestdockerimage "./memorytest.sh" 44 seconds ago vigorous_neumann
e7412f3b7b8c fileiodockerimage "./fileio.sh" About a minute ago brave_shannon
1f28e461f910 100000cpudockerimage "./100000_cpu.sh" About a minute ago festive_shamir
f01137eebd66 10000cpudockerimage "./10000_cpu.sh" 2 minutes ago mystifying_austin
8e4f0b8275f2 kindest/node:v1.27.3 "/usr/local/bin/entr..." 3 months ago kind-control-plane
(base) sumit@Rishabh ~ % docker logs -f intelligent_hopper
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

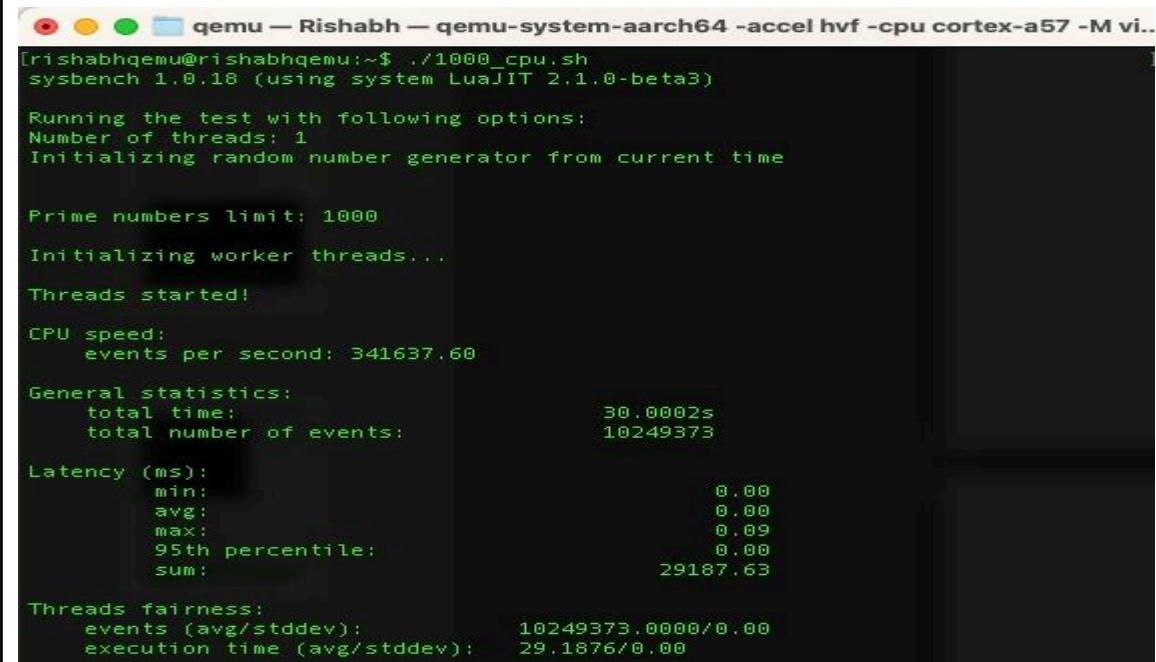
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

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

3.1 CPU Testing

QEMU results for 2 GB 3 cores for max-prime = 1000 and time = 30 seconds, disk image = qcow2

Screenshot



```
[root@rishabhqemu ~]# ./1000_cpu.sh
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: 1000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 341637.60

General statistics:
total time: 30.0002s
total number of events: 10249373

Latency (ms):
min: 0.00
avg: 0.00
max: 0.09
95th percentile: 0.00
sum: 29187.63

Threads fairness:
events (avg/stddev): 10249373.0000/0.00
execution time (avg/stddev): 29.1876/0.00
```

Detailed report:

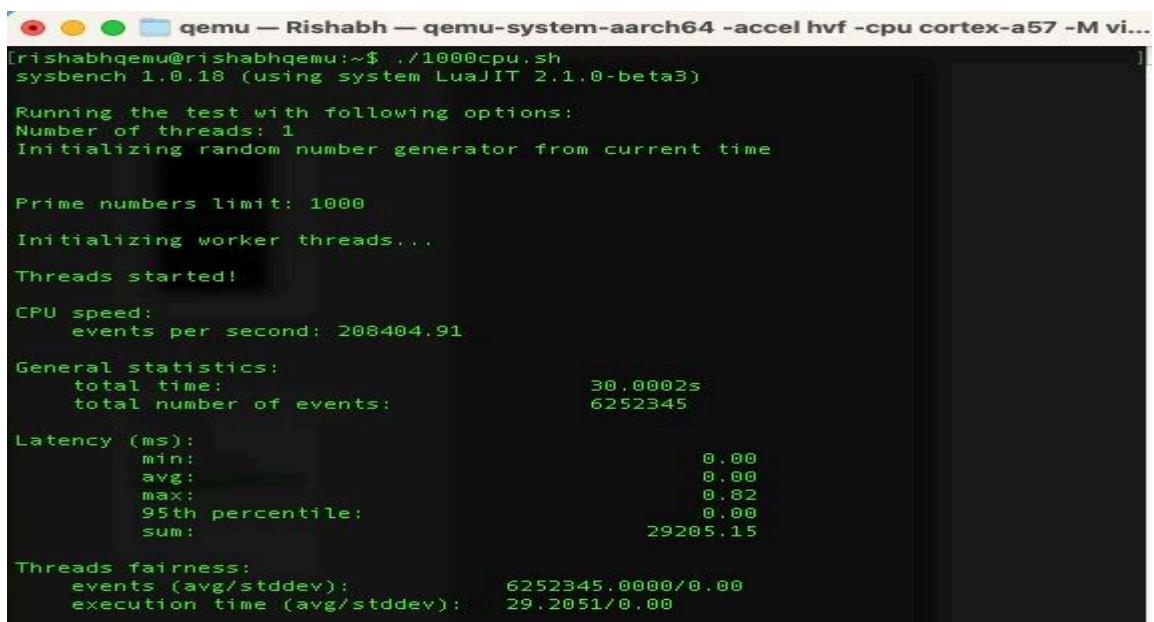
Iteration No	Events per second
1	341637.60
2	340679.56
3	340577.04
4	338574.16
5	341567.80

Observations:

Average events per second	340607.23
Minimum Number of events per second recorded	338574.16
Maximum Number of events per second recorded	341567.80

QEMU Results, Configuration: 2 GB 3 cores for max-prime = 1000 and time = 30 seconds, disk image = raw

Screenshot



```
[rishabh@qemu:~$ ./1000cpu.sh
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: 1000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 208404.91

General statistics:
total time: 30.0002s
total number of events: 6252345

Latency (ms):
min: 0.00
avg: 0.00
max: 0.82
95th percentile: 0.00
sum: 29205.15

Threads fairness:
events (avg/stddev): 6252345.0000/0.00
execution time (avg/stddev): 29.2051/0.00
```

Detailed report:

Iteration No	Events per second
1	208404.91
2	207453.34
3	208782.32
4	207134.56
5	207873.53

Observations:

Average events per second	207929.73
Minimum Number of events per second recorded	207134.56
Maximum Number of events per second recorded	208782.32

Docker Results for 2 GB 3 cores for max-prime = 1000 and time = 30 seconds

Screenshot

```
(base) sumit@Rishabh ~ % docker logs -f intelligent_hopper
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: 1000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 113244.62

General statistics:
  total time:          30.0002s
  total number of events: 3397461

Latency (ms):
  min:                0.00
  avg:                0.01
  max:               22.07
  95th percentile:    0.01
  sum:              29127.54

Threads fairness:
  events (avg/stddev):   3397461.0000/0.00
  execution time (avg/stddev): 29.1275/0.00
```

Detailed report:

Iteration No	Events per second
1	113244.62
2	142177.78
3	195216.90
4	203876.23
5	205721.99

Observations:

Average events per second	172047.50
Minimum Number of events per second recorded	113244.62
Maximum Number of events per second recorded	205721.99

Conclusion: QEMU with qcow2 delivers the highest performance for computational tasks among the tested configurations, significantly outperforming QEMU with raw disk images and Docker, indicating its superior efficiency for intensive operations on a 2 GB RAM and 3 core setup.

QEMU Results for 2 GB 3 cores for max-prime = 10000 and time = 30 seconds, disk image = qcow2

Screenshot

The screenshot displays two terminal windows side-by-side, both titled "qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi...". The left window shows results for 3 cores, and the right window shows results for 1 core. Both windows output the same sysbench command and its execution details.

```
[rishabh@rishabh:~$ ./10000_cpu.sh
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: 11040.06

General statistics:
total time: 30.00002s
total number of events: 331210

Latency (ms):
min: 0.09
avg: 0.09
max: 0.33
95th percentile: 0.09
sum: 29966.39

Threads fairness:
events (avg/stddev): 331210.0000/0.00
execution time (avg/stddev): 29.9664/0.00

[rishabh@rishabh:~$ ./10000_cpu.sh
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: 10954.40

General statistics:
total time: 30.00002s
total number of events: 328640

Latency (ms):
min: 0.09
avg: 0.09
max: 9.29
95th percentile: 0.10
sum: 29962.30

Threads fairness:
events (avg/stddev): 328640.0000/0.00
execution time (avg/stddev): 29.9623/0.00
```

Detailed report:

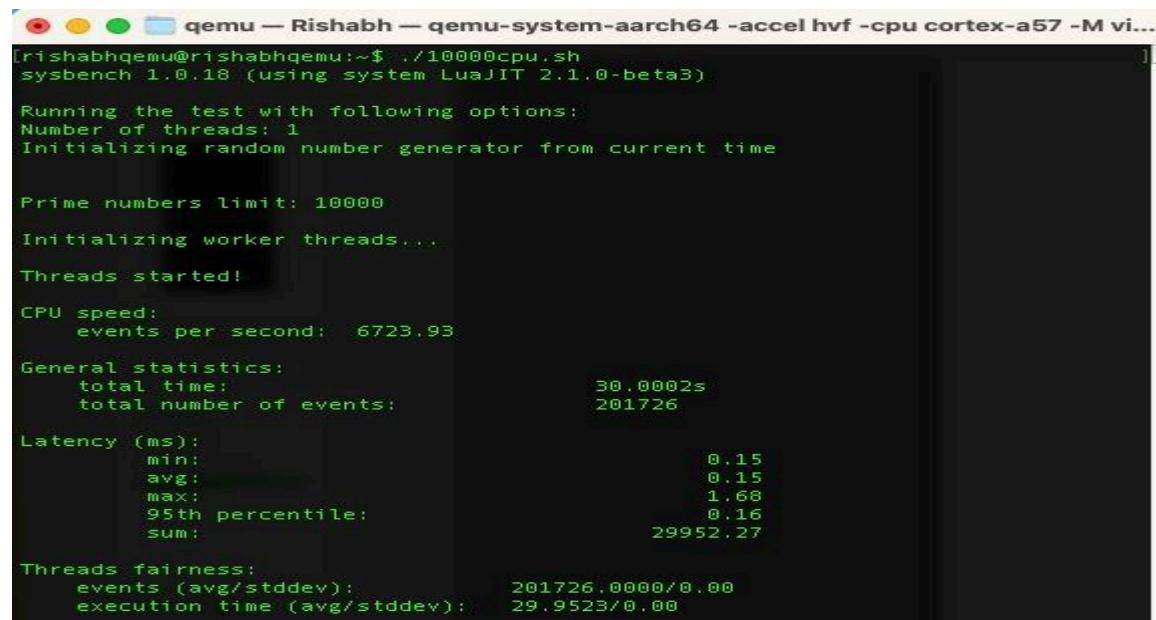
Iteration No	Events per second
1	11048.06
2	10954.40
3	10990.45
4	11459.84
5	10937.78

Observations:

Average events per second	11078.11
Minimum Number of events per second recorded	10937.78
Maximum Number of events per second recorded	11459.84

QEMU Results, Configuration: 2 GB 3 cores for max-prime = 10000 and time = 30 seconds, disk image = raw

Screenshot



```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi...
[rishabh@rishabh:~$ ./10000cpu.sh
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: 6723.93

General statistics:
total time: 30.0002s
total number of events: 201726

Latency (ms):
min: 0.15
avg: 0.15
max: 1.68
95th percentile: 0.16
sum: 29952.27

Threads fairness:
events (avg/stddev): 201726.0000/0.00
execution time (avg/stddev): 29.9523/0.00
```

Detailed report:

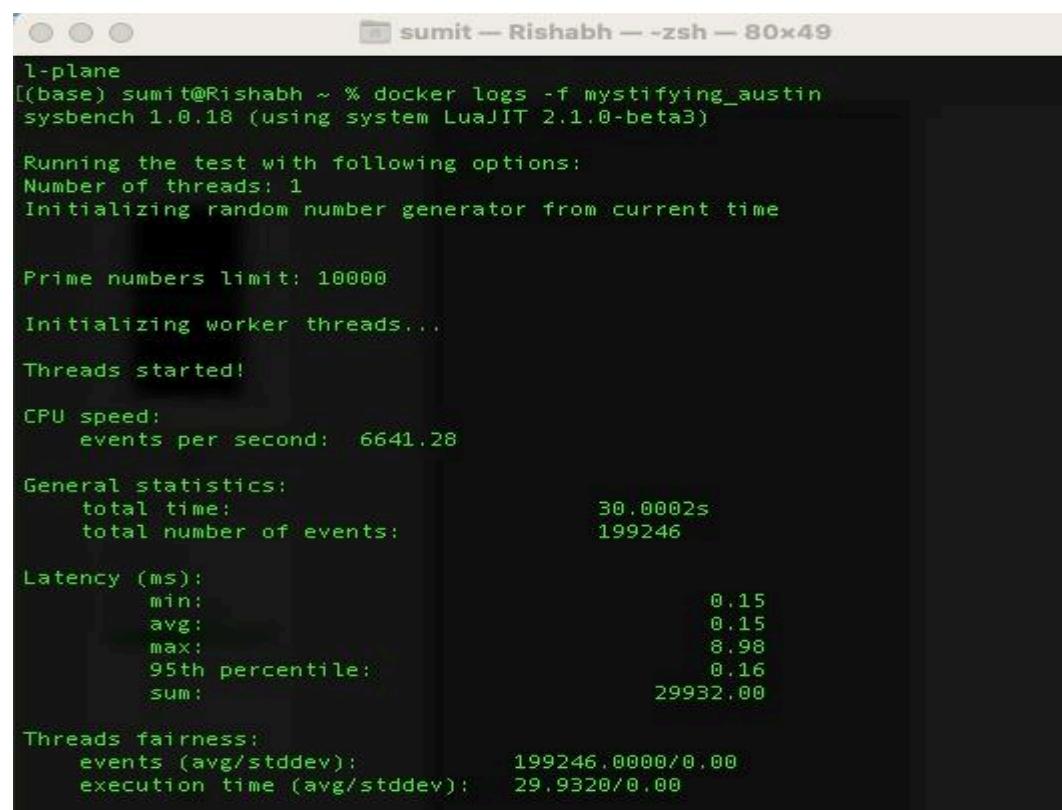
Iteration No	Events per second
1	6723.93
2	6788.34
3	6812.48
4	6883.01
5	6782.43

Observations:

Average events per second	6798.03
Minimum Number of events per second recorded	6723.93
Maximum Number of events per second recorded	6883.01

Docker results for 2 GB 3 cores for max-prime = 10000 and time = 30 seconds

Screenshot



The screenshot shows a terminal window titled "sumit — Rishabh — -zsh — 80x49". The window displays the output of a sysbench test run within a Docker container. The test configuration includes a prime numbers limit of 10000 and a single thread. The output provides general statistics like total time (30.0002s), total events (199246), and latency metrics (min: 0.15ms, avg: 0.15ms, max: 8.98ms, 95th percentile: 0.16ms). It also shows threads fairness and execution time statistics.

```
l-plane
[(base) sumit@Rishabh ~ % docker logs -f mystifying_austin
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: 6641.28

General statistics:
total time: 30.0002s
total number of events: 199246

Latency (ms):
min: 0.15
avg: 0.15
max: 8.98
95th percentile: 0.16
sum: 29932.00

Threads fairness:
events (avg/stddev): 199246.0000/0.00
execution time (avg/stddev): 29.9320/0.00
```

Detailed report:

Iteration No	Events per second
1	6641.28
2	6476.15
3	5287.40
4	4489.66
5	3749.99

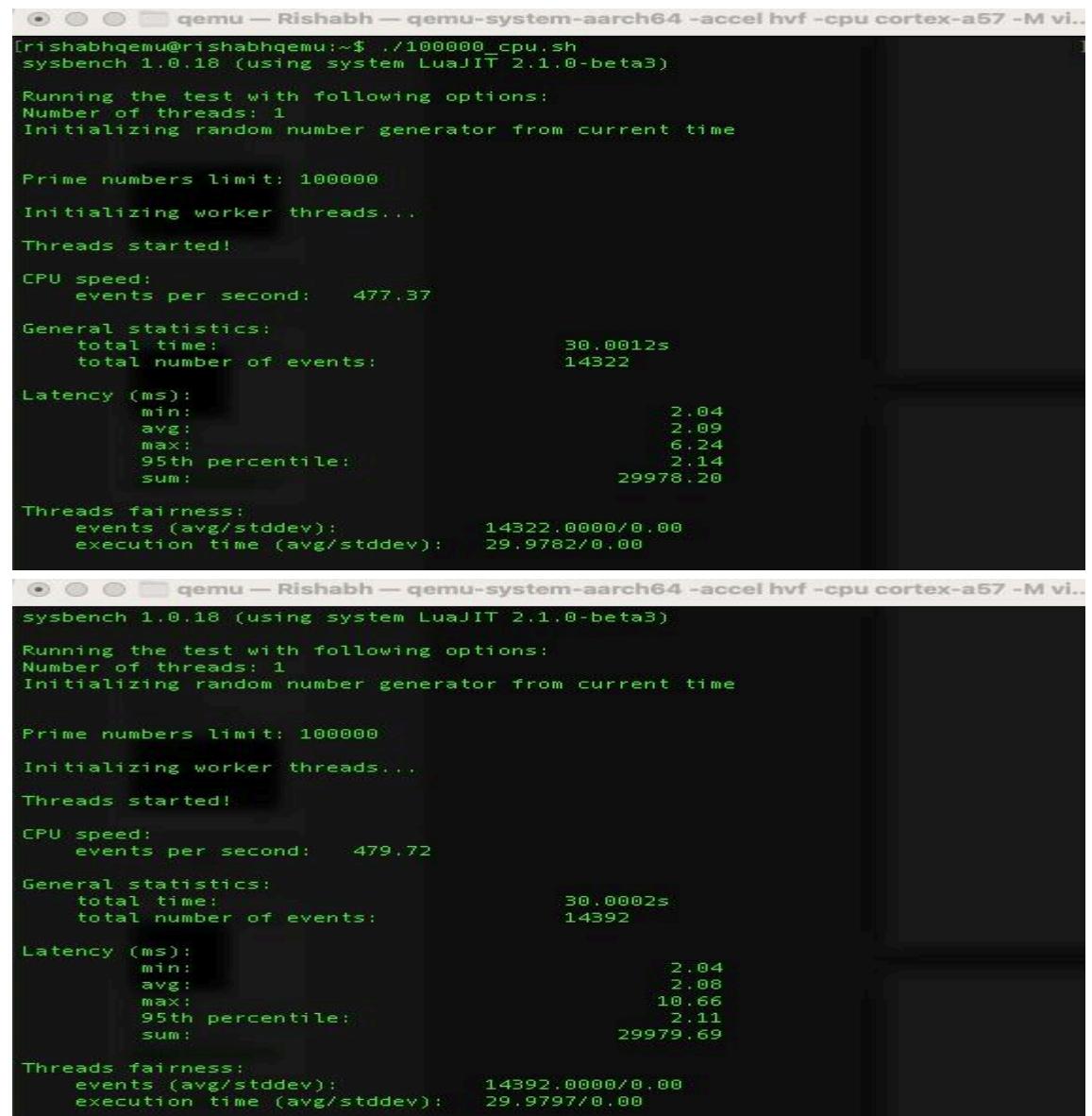
observations:

Average events per second	5328.89
Minimum Number of events per second recorded	3749.99
Maximum Number of events per second recorded	6641.28

Conclusion: For the task of calculating prime numbers up to 10,000 within 30 seconds on a 2 GB RAM and 3 core configuration, QEMU with a qcow2 disk image outperforms both its raw counterpart and Docker, achieving the highest average events per second at 11,078.11. QEMU with a raw disk image follows with a moderate performance average of 6,798.03 events per second, while Docker exhibits the lowest throughput with an average of 5,328.89 events per second, highlighting the efficiency and performance advantages of using QEMU with qcow2 for intensive computational tasks.

QEMU Results for 2 GB 3 cores for max-prime = 100000 and time = 30 seconds, disk image = qcow2

Screenshot



The screenshot shows two terminal windows side-by-side, both running the command `./100000_cpu.sh`. The top window shows results for iteration 1, and the bottom window shows results for iteration 2. Both windows display the same test configuration: sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3), 1 thread, and a prime limit of 100000. The output includes general statistics like total time (30.0012s for iteration 1, 30.0002s for iteration 2), total events (14322 for iteration 1, 14392 for iteration 2), and latency metrics (min, avg, max, 95th percentile, sum). Thread fairness statistics are also provided.

```
[rishabh@rishabh-qemu:~$ ./100000_cpu.sh
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: 100000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 477.37

General statistics:
total time: 30.0012s
total number of events: 14322

Latency (ms):
min: 2.04
avg: 2.09
max: 6.24
95th percentile: 2.14
sum: 29978.20

Threads fairness:
events (avg/stddev): 14322.0000/0.00
execution time (avg/stddev): 29.9782/0.00

[rishabh@rishabh-qemu:~$ ./100000_cpu.sh
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: 100000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 479.72

General statistics:
total time: 30.0002s
total number of events: 14392

Latency (ms):
min: 2.04
avg: 2.08
max: 10.66
95th percentile: 2.11
sum: 29979.69

Threads fairness:
events (avg/stddev): 14392.0000/0.00
execution time (avg/stddev): 29.9797/0.00
```

Detailed report:

Iteration No	Events per second
1	477.37
2	479.72
3	477.98
4	478.47
5	478.93

Observations:

Average events per second	478.49
Minimum Number of events per second recorded	477.37
Maximum Number of events per second recorded	479.72

QEMU Results, Configuration: 2 GB 3 cores for max-prime = 100000 and time = 30 seconds, disk image = raw

Screenshot

```
[rishabhqemu@rishabhqemu:~/]$ ./100000cpu.sh
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: 100000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 293.05

General statistics:
total time: 30.0012s
total number of events: 8792

Latency (ms):
min: 3.35
avg: 3.41
max: 6.41
95th percentile: 3.43
sum: 29968.42

Threads fairness:
events (avg/stddev): 8792.0000/0.00
execution time (avg/stddev): 29.9684/0.00
```

Detailed report:

Iteration No	Events per second
1	293.05
2	295.76
3	294.32
4	293.77
5	295.42

Observations:

Average events per second	294.46
Minimum Number of events per second recorded	293.05
Maximum Number of events per second recorded	295.76

Docker results for 2 GB 3 cores for max-prime = 100000 and time = 30 seconds

Screenshot

```
[sumit@Rishabh ~] docker logs -f festive_shamir
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: 100000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 284.19

General statistics:
total time: 30.00005s
total number of events: 8526

Latency (ms):
min: 3.35
avg: 3.52
max: 14.42
95th percentile: 3.89
sum: 29971.88

Threads fairness:
events (avg/stddev): 8526.0000/0.00
execution time (avg/stddev): 29.9719/0.00
```

Detailed report:

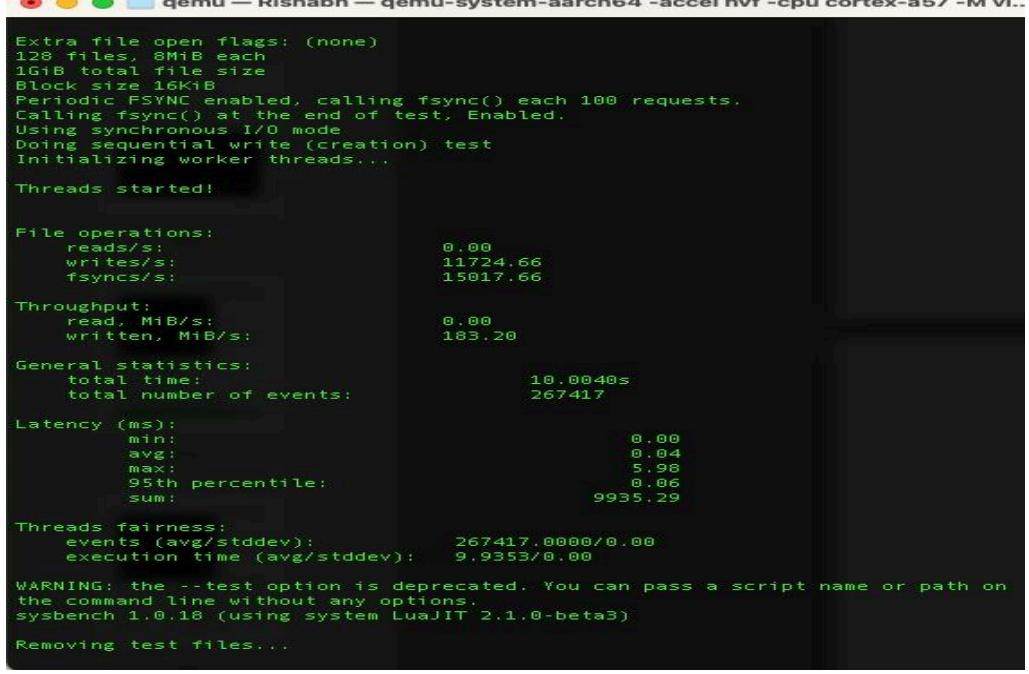
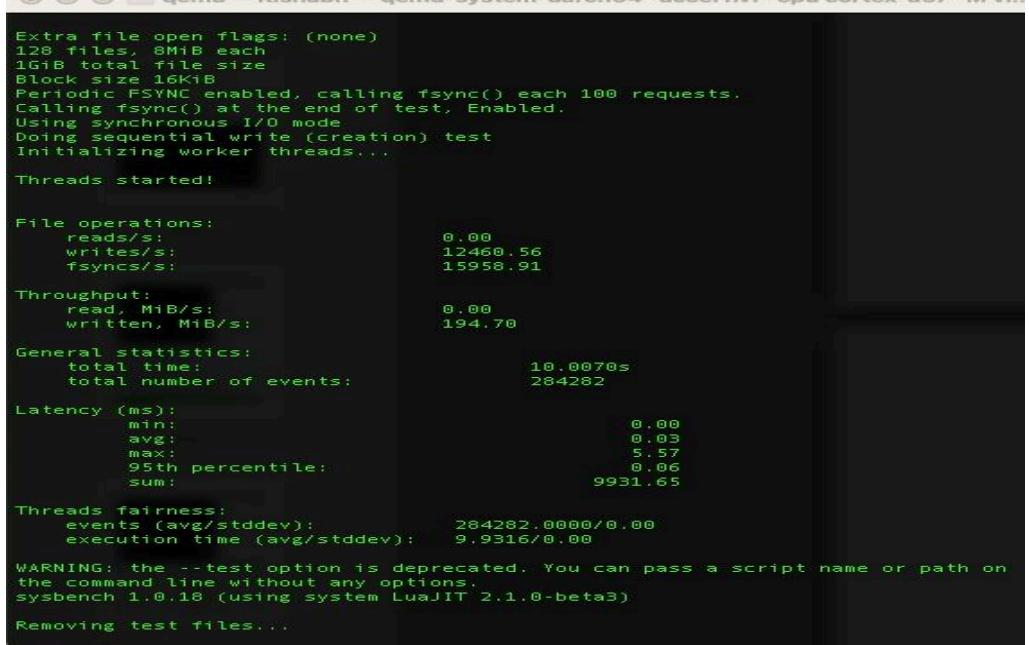
Iteration No	Events per second
1	284.19
2	235.07
3	188.07
4	176.92
5	229.84

Observations:

Average events per second	222.82
Minimum Number of events per second recorded	176.92
Maximum Number of events per second recorded	284.19

Conclusion: In the task of computing prime numbers up to 100,000 within 30 seconds on a setup with 2 GB RAM and 3 cores, QEMU with a qcow2 disk image leads in performance, achieving an average of 478.49 events per second. QEMU with a raw disk image follows, with a lower average performance of 294.46 events per second. Docker trails behind both QEMU configurations, with the lowest average performance of 222.82 events per second. This outcome underscores the superior computational efficiency of QEMU with qcow2 for high-complexity tasks.

3.2 File I/O Testing

QEMU Results for 2 GB 3 cores for sequential write, disk image = qcow2	
Iteration No	Screenshots
1	 <pre> Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 11724.66 fsyncs/s: 15017.66 Throughput: read, MiB/s: 0.00 written, MiB/s: 183.20 General statistics: total time: 10.0040s total number of events: 267417 Latency (ms): min: 0.00 avg: 0.04 max: 5.98 95th percentile: 0.06 sum: 9935.29 Threads fairness: events (avg/stddev): 267417.0000/0.00 execution time (avg/stddev): 9.9353/0.00 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) Removing test files... </pre>
2	 <pre> Extra file open flags: (none) 128 files, 8MiB each 1GiB total file size Block size 16KiB Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing sequential write (creation) test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 12460.56 fsyncs/s: 15958.91 Throughput: read, MiB/s: 0.00 written, MiB/s: 194.70 General statistics: total time: 10.0070s total number of events: 284282 Latency (ms): min: 0.00 avg: 0.03 max: 5.57 95th percentile: 0.06 sum: 9931.65 Threads fairness: events (avg/stddev): 284282.0000/0.00 execution time (avg/stddev): 9.9316/0.00 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) Removing test files... </pre>

3

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi..
```

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
reads/s: 0.00
writes/s: 11750.44
fsyncs/s: 15043.46

Throughput:
read, MiB/s: 0.00
written, MiB/s: 183.60

General statistics:
total time: 10.0076s
total number of events: 268829

Latency (ms):
min: 0.00
avg: 0.04
max: 25.32
95th percentile: 0.06
sum: 9934.94

Threads fairness:
events (avg/stddev): 268829.0000/0.00
execution time (avg/stddev): 9.9349/0.00

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)

Removing test files...

4

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi..
```

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
reads/s: 0.00
writes/s: 12193.73
fsyncs/s: 15618.27

Throughput:
read, MiB/s: 0.00
written, MiB/s: 190.53

General statistics:
total time: 10.0046s
total number of events: 278135

Latency (ms):
min: 0.00
avg: 0.04
max: 6.12
95th percentile: 0.06
sum: 9935.30

Threads fairness:
events (avg/stddev): 278135.0000/0.00
execution time (avg/stddev): 9.9353/0.00

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)

Removing test files...

5

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi..
```

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
reads/s: 0.00
writes/s: 12460.53
fsyncs/s: 15951.88

Throughput:
read, MiB/s: 0.00
written, MiB/s: 194.70

General statistics:
total time: 10.0071s
total number of events: 284212

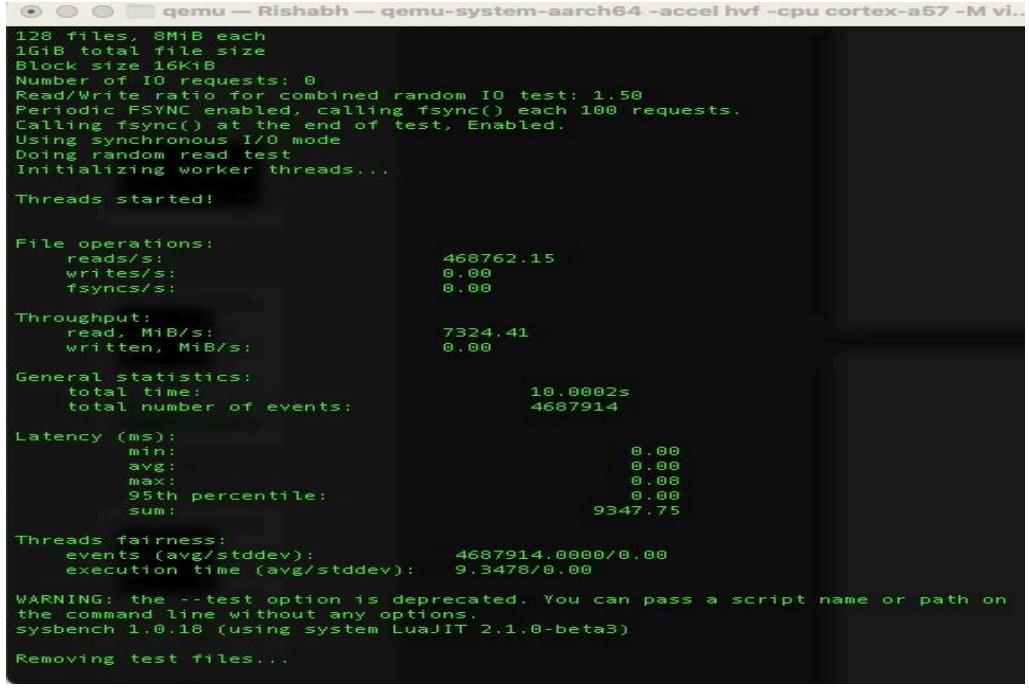
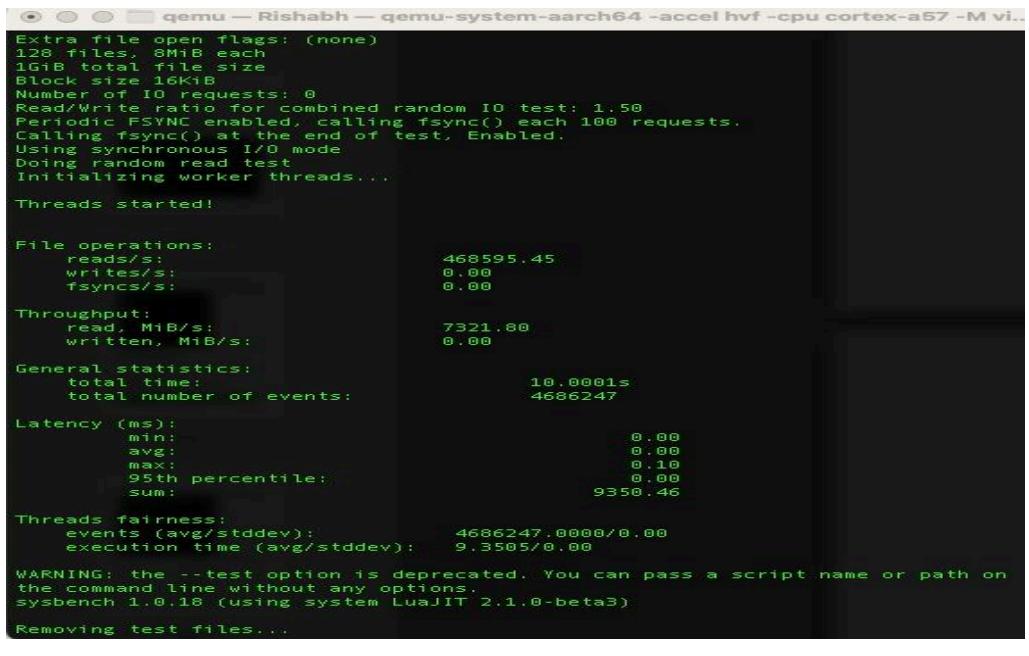
Latency (ms):
min: 0.00
avg: 0.03
max: 5.59
95th percentile: 0.06
sum: 9931.91

Threads fairness:
events (avg/stddev): 284212.0000/0.00
execution time (avg/stddev): 9.9319/0.00

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)

Removing test files...

QEMU Results for 2 GB 3 cores for random read, disk image = qcow2

Iteration No	Screenshots
1	 <pre> qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi.. 128 files, 8MiB each 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 read test Initializing worker threads... Threads started! File operations: reads/s: 468762.15 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 7324.41 written, MiB/s: 0.00 General statistics: total time: 10.0002s total number of events: 4687914 Latency (ms): min: 0.00 avg: 0.00 max: 0.08 95th percentile: 0.00 sum: 9347.75 Threads fairness: events (avg/stddev): 4687914.0000/0.00 execution time (avg/stddev): 9.3478/0.00 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) Removing test files... </pre>
2	 <pre> qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi.. Extra file open flags: (none) 128 files, 8MiB each 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 read test Initializing worker threads... Threads started! File operations: reads/s: 468595.45 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 7321.80 written, MiB/s: 0.00 General statistics: total time: 10.0001s total number of events: 4686247 Latency (ms): min: 0.00 avg: 0.00 max: 0.10 95th percentile: 0.00 sum: 9350.46 Threads fairness: events (avg/stddev): 4686247.0000/0.00 execution time (avg/stddev): 9.3505/0.00 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) Removing test files... </pre>

3

```
qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi..
128 files, 8MiB each
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 read test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          467688.95
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     7307.64
  written, MiB/s:  0.00

General statistics:
  total time:      10.0001s
  total number of events: 4677176

Latency (ms):
  min:              0.00
  avg:              0.00
  max:              0.10
  95th percentile: 0.00
  sum:              9349.09

Threads fairness:
  events (avg/stddev): 4677176.0000/0.00
  execution time (avg/stddev): 9.3491/0.00

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)

Removing test files...
```

4

```
qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi..
128 Files, 8MiB each
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 read test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          461502.16
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     7210.97
  written, MiB/s:  0.00

General statistics:
  total time:      10.0002s
  total number of events: 4615390

Latency (ms):
  min:              0.00
  avg:              0.00
  max:              0.28
  95th percentile: 0.00
  sum:              9343.16

Threads fairness:
  events (avg/stddev): 4615390.0000/0.00
  execution time (avg/stddev): 9.3432/0.00

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)

Removing test files...
```

5

```
qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi..
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 read test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          440269.69
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     6879.21
  written, MiB/s:  0.00

General statistics:
  total time:      10.0001s
  total number of events: 4403040

Latency (ms):
  min:              0.00
  avg:              0.00
  max:              7.35
  95th percentile: 0.00
  sum:              9346.61

Threads fairness:
  events (avg/stddev): 4403040.0000/0.00
  execution time (avg/stddev): 9.3466/0.00

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)

Removing test files...
File I/O tests completed.
rishabh@qemu:~$
```

QEMU Results for 2 GB 3 cores for sequential write & random read,
disk image = qcow2

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 11724.66 fsyncs/s: 15017.66	reads/s: 468762.15 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 12460.56 fsyncs/s: 15958.91	reads/s: 468595.45 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 11750.44. fsyncs/s: 15043.46	reads/s: 467688.95 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 12193.73 fsyncs/s: 15618.27	reads/s: 461502.16 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 12460.53 fsyncs/s: 15951.88	reads/s: 440269.69 writes/s: 0.00 fsyncs/s: 0.00

QEMU Results for 2 GB 3 cores for sequential write & random read,
disk image = raw

Screenshot

```

qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi...
Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 10KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test. Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          0.00
  writes/s:        7380.70
  fsyncs/s:       9453.59

Throughput:
  read, MiB/s:      0.00
  written, MiB/s: 115.32

General statistics:
  total time:      10.0116s
  total number of events: 168427

Latency (ms):
  min:              0.00
  avg:             0.06
  max:             4.73
  95th percentile: 0.0
  sum:            9822.94

Threads fairness:
  events (avg/stddev): 168427.0000/0.00
  execution time (avg/stddev): 9.8229/0.00

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)

Removing test files...

```

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 7380.70 fsyncs/s: 9453.59	reads/s: 412493.34 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 7534.39 fsyncs/s: 9564.34	reads/s: 424533.24 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 7439.49 fsyncs/s: 9524.43	reads/s: 432346.85 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 7423.67 fsyncs/s: 9634.30	reads/s: 442548.76 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 7583.53 fsyncs/s: 9834.51	reads/s: 431453.32 writes/s: 0.00 fsyncs/s: 0.00

Docker Results for 2 GB 3 cores for sequential write & random read

```

sumit — Rishabh — -zsh — 88x43

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        11903.30
  fsyncs/s:       15244.42

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   185.99

General statistics:
  total time:           10.0048s
  total number of events: 271502

Latency (ms):
  min:                  0.01
  avg:                 0.04
  max:                13.96
  95th percentile:     0.05
  sum:               9850.95

Threads fairness:
  events (avg/stddev): 271502.0000/0.00
  execution time (avg/stddev): 9.8509/0.00

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)

Removing test files...

```

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 11903.30 fsyncs/s: 15244.42	reads/s: 10736.49 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 12293.21 fsyncs/s: 15740.81	reads/s: 14430.42 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 12409.33 fsyncs/s: 15889.57	reads/s: 15129.51 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 12583.74 fsyncs/s: 16118.98	reads/s: 23726.60 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 11910.71 fsyncs/s: 15256.20	reads/s: 47523.49 writes/s: 0.00 fsyncs/s: 0.00

Conclusion: QEMU with a qcow2 disk image outperforms both QEMU with a raw disk image and Docker in sequential write and random read tasks, showcasing superior I/O efficiency. Docker demonstrates competitive write performance but falls short in random read operations compared to QEMU configurations.

3.3 Memory I/O Testing

QEMU Results for 2 GB 3 cores for Sequential Access, disk image = qcow2	
Iteration	Results
1	<pre>[rishabh@rishabh-qemu:~\$./memorytest.sh Running First Memory Test: Sequential Access Iteration 1 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 86338795 (8632619.95 per second) 84315.23 MiB transferred (8430.29 MiB/sec) General statistics: total time: 10.00002s total number of events: 86338795 Latency (ms): min: 0.00 avg: 0.00 max: 1.80 95th percentile: 0.00 sum: 4194.52 Threads fairness: events (avg/stddev): 86338795.0000/0.00 execution time (avg/stddev): 4.1945/0.00</pre>
2	<pre>Iteration 2 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 85793905 (8578269.95 per second) 83783.11 MiB transferred (8377.22 MiB/sec) General statistics: total time: 10.00001s total number of events: 85793905 Latency (ms): min: 0.00 avg: 0.00 max: 1.72 95th percentile: 0.00 sum: 4194.09 Threads fairness: events (avg/stddev): 85793905.0000/0.00 execution time (avg/stddev): 4.1941/0.00</pre>

```

gemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi..
```

Iteration 3
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 86425106 (8641629.27 per second)
84399.52 MiB transferred (8439.89 MiB/sec)

General statistics:
total time: 10.0001s
total number of events: 86425106

Latency (ms):
min: 0.00
avg: 0.00
max: 0.00
95th percentile: 0.00
sum: 4193.67

Threads fairness:
events (avg/stddev): 86425106.0000/0.00
execution time (avg/stddev): 4.1937/0.00

Iteration 4
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 85710073 (8570326.44 per second)
83781.24 MiB transferred (8369.46 MiB/sec)

General statistics:
total time: 10.0001s
total number of events: 85710073

Latency (ms):
min: 0.00
avg: 0.00
max: 1.05
95th percentile: 0.00
sum: 4190.81

Threads fairness:
events (avg/stddev): 85710073.0000/0.00
execution time (avg/stddev): 4.1988/0.00

Iteration 5

Iteration 5
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 86270324 (8626403.04 per second)
84248.36 MiB transferred (8424.22 MiB/sec)

General statistics:
total time: 10.0002s
total number of events: 86270324

Latency (ms):
min: 0.00
avg: 0.00
max: 0.11
95th percentile: 0.00
sum: 4189.47

Threads fairness:
events (avg/stddev): 86270324.0000/0.00
execution time (avg/stddev): 4.1895/0.00

QEMU Results for 2 GB 3 cores for Random Access, disk image = qcow2

Iteration	Results
1	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi... Running Second Memory Test: Random Access Iteration 1 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18583919 (1858244.75 per second) 18148.36 MiB transferred (1814.69 MiB/sec) General statistics: total time: 10.00002s total number of events: 18583919 Latency (ms): min: 0.00 avg: 0.00 max: 4.41 95th percentile: 0.00 sum: 8732.00 Threads fairness: events (avg/stddev): 18583919.0000/0.00 execution time (avg/stddev): 8.7320/0.00</pre>
2	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi... Iteration 2 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 18561217 (1855979.76 per second) 18126.19 MiB transferred (1812.48 MiB/sec) General statistics: total time: 10.00002s total number of events: 18561217 Latency (ms): min: 0.00 avg: 0.00 max: 7.19 95th percentile: 0.00 sum: 8736.44 Threads fairness: events (avg/stddev): 18561217.0000/0.00 execution time (avg/stddev): 8.7364/0.00</pre>

```

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi.. Iteration 3
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 18617705 (1861638.57 per second)
18181.35 MiB transferred (1818.01 MiB/sec)

General statistics:
    total time: 10.00001s
    total number of events: 18617705

Latency (ms):
    min: 0.00
    avg: 0.00
    max: 0.62
    95th percentile: 0.00
    sum: 8736.09

Threads fairness:
    events (avg/stddev): 18617705.0000/0.00
    execution time (avg/stddev): 8.7369/0.00

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi.. Iteration 4
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 18390702 (1838939.34 per second)
17959.67 MiB transferred (1795.84 MiB/sec)

General statistics:
    total time: 10.00001s
    total number of events: 18390702

Latency (ms):
    min: 0.00
    avg: 0.00
    max: 3.79
    95th percentile: 0.00
    sum: 8722.17

Threads fairness:
    events (avg/stddev): 18390702.0000/0.00
    execution time (avg/stddev): 8.7222/0.00

qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi.. Iteration 5
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 18518548 (1851733.86 per second)
18084.52 MiB transferred (1808.33 MiB/sec)

General statistics:
    total time: 10.00001s
    total number of events: 18518548

Latency (ms):
    min: 0.00
    avg: 0.00
    max: 5.44
    95th percentile: 0.00
    sum: 8740.36

Threads fairness:
    events (avg/stddev): 18518548.0000/0.00
    execution time (avg/stddev): 8.7404/0.00

Memory tests completed.
rishabh@qemu:~$ []

```

QEMU Results for 2 GB 3 cores for sequential & random access, disk image = qcow2

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	8632619.95	1858244.75
2	8578269.95	1855979.76
3	8641629.27	1861638.57
4	8570326.44	1838939.34
5	8626403.04	1851733.86

QEMU Results for 2 GB 3 cores for sequential & random access, disk image = raw

Screenshot

```
rishabh@qemu:~/memtest$ ./memorytest.sh
Running First Memory Test: Sequential Access
Iteration: 1
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

Running memory speed test with the following options:
block size: 1K1B
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 52732290 (5272544.69 per second)
51496.38 MiB transferred (5148.97 MiB/sec)

General statistics:
    total time:                      10.00002s
    total number of events:           52732290

Latency (ms):
    min:                            0.00
    avg:                            0.00
    max:                            2.32
    95th percentile:                0.00
    sum:                           4195.13

Threads fairness:
    events (avg/stddev):           52732290.0000/0.00
    execution time (avg/stddev):   4.1951/0.00
```

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	5272544.69	1114569.34
2	5284438.94	1127823.43
3	5284538.83	1118932.81
4	5314754.32	1114257.56
5	5305853.12	1123498.65

Docker Results for 2 GB 3 cores for sequential & random access

```
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 33896159 (3389307.40 per second)

33101.72 MiB transferred (3309.87 MiB/sec)

General statistics:
  total time:          10.0001s
  total number of events: 33896159

Latency (ms):
  min:                 0.00
  avg:                 0.00
  max:                12.43
  95th percentile:    0.00
  sum:                4174.95

Threads fairness:
  events (avg/stddev):   33896159.0000/0.00
  execution time (avg/stddev): 4.1749/0.00
```

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	3389307.40	664833.63
2	3786823.35	658488.14
3	3798987.61	764755.14
4	3278306.77	902097.85
5	3164572.50	853537.71

Conclusion: QEMU with a qcow2 disk image significantly leads in performance for both sequential and random access, with operations per second notably higher than both QEMU with a raw disk image and Docker. QEMU with a raw disk image ranks second, offering substantial throughput but at a considerably lower rate than qcow2. Docker, while providing valuable performance, trails behind the QEMU setups, especially in sequential access operations. This hierarchy underscores qcow2's efficiency in optimizing access operations on a 2 GB RAM and 3 core configuration.

4. Configuration 1: 3 GB RAM with 6 Cores

QEMU Setup (disk image: qcow2):



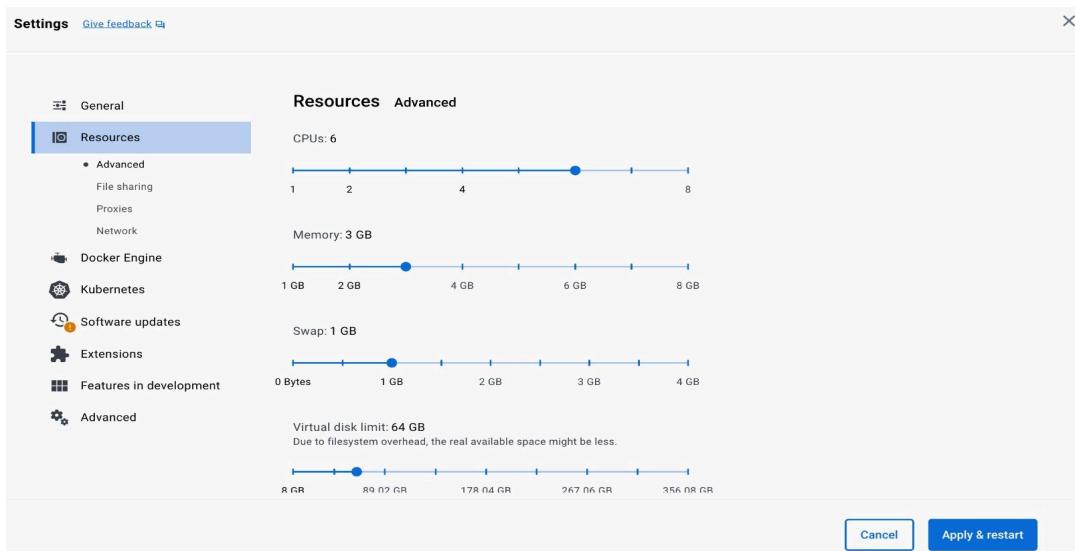
```
(base) sumit@Rishabh: ~ % /opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 3072 -smp 6 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=flash,format=raw,readonly=on \
-drive if=none,file=ubuntu.img,format=qcow2,id=hd0 \
-device virtio-blk-device,drive=hd0,serial="dummyserial" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-cdrom ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

QEMU Setup (disk image: raw):



```
((base) sumit@Rishabh: ~ % cd Desktop/Rishabh/qemu
(base) sumit@Rishabh: ~ % /opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 3072 -smp 6 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.1/share/qemu/edk2-aarch64-code.fd,if=flash,format=raw,readonly=on \
-drive if=none,file=ubuntu.img,format=raw,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 ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

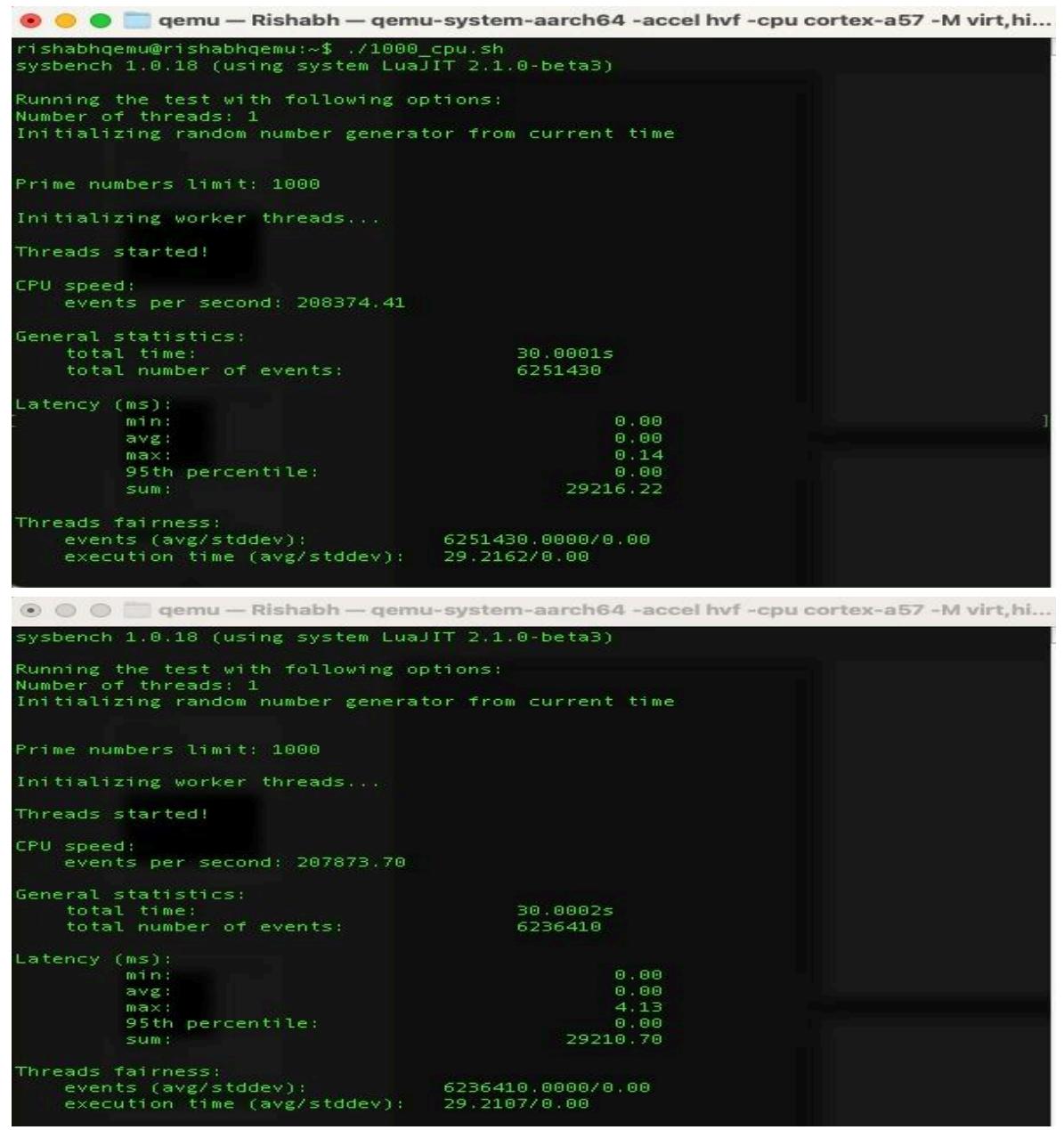
Docker Setup:



4.1 CPU Testing

QEMU Results, Configuration: 3 GB 6 cores for max-prime = 1000 and time = 30 seconds, disk image = qcow2

Screenshot



The screenshot displays two terminal windows side-by-side, both titled "qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...".

Terminal 1 (Left):

```
rishabh@rishabh:~/qemu$ ./1000_cpu.sh
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: 1000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 208374.41

General statistics:
total time: 30.0001s
total number of events: 6251430

Latency (ms):
min: 0.00
avg: 0.00
max: 0.14
95th percentile: 0.00
sum: 29216.22

Threads fairness:
events (avg/stddev): 6251430.0000/0.00
execution time (avg/stddev): 29.2162/0.00
```

Terminal 2 (Right):

```
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: 1000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 207873.70

General statistics:
total time: 30.0002s
total number of events: 6236410

Latency (ms):
min: 0.00
avg: 0.00
max: 4.13
95th percentile: 0.00
sum: 29210.70

Threads fairness:
events (avg/stddev): 6236410.0000/0.00
execution time (avg/stddev): 29.2107/0.00
```

[Detailed report](#):

Iteration No	Events/second
1	208374.41
2	207873.70
3	207918.89
4	207814.45
5	208367.74

Observations:

Average events per second	208374.41
Minimum Number of events per second recorded	207814.45
Maximum Number of events per second recorded	208374.41

QEMU Results, Configuration: 3 GB 6 cores for max-prime = 1000 and time = 30 seconds, disk image = raw

Screenshot

```
[● ● ● qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi...
[rishabh@rishabh:~$ ./1000cpu.sh
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: 1000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 206841.57

General statistics:
total time: 30.0002s
total number of events: 6205492

Latency (ms):
min: 0.00
avg: 0.00
max: 2.04
95th percentile: 0.00
sum: 29217.14

Threads fairness:
events (avg/stddev): 6205492.0000/0.00
execution time (avg/stddev): 29.2171/0.00
```

Detailed report:

Iteration No	Events/second
1	206841.57
2	206923.43
3	207013.32
4	207143.75
5	205342.51

Observations:

Average events per second	206652.91
Minimum Number of events per second recorded	205342.51
Maximum Number of events per second recorded	207143.75

Docker Results, Configuration: 3 GB 6 cores for max-prime = 1000 and time = 30 seconds

```
[base) sumit@Rishabh ~ % docker logs -f awesome_torvalds
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: 1000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 203626.34

General statistics:
total time: 30.0001s
total number of events: 6108996

Latency (ms):
min: 0.00
avg: 0.00
max: 2.22
95th percentile: 0.01
sum: 29197.10

Threads fairness:
events (avg/stddev): 6108996.0000/0.00
execution time (avg/stddev): 29.1971/0.00
```

Detailed report:

Iteration No	Events per second
1	203626.34
2	200674.88
3	184488.80
4	150751.01
5	137677.38

Observations:

Average events per second	175443.68
Minimum Number of events per second recorded	137677.38
Maximum Number of events per second recorded	203626.34

Conclusion: For the task of computing prime numbers up to 1000 within 30 seconds on a 3 GB RAM and 6 core setup, QEMU with a qcow2 disk image achieves the highest average events per second at 208,374.41, closely followed by QEMU with a raw disk image at 206,652.91. Docker trails behind, with a significantly lower average of 175,443.68 events per second. This demonstrates the superior efficiency of QEMU configurations, particularly with qcow2, for intensive computational tasks.

QEMU Results, Configuration: 3 GB 6 cores for max-prime = 10000 and time = 30 seconds, disk image = qcow2

Screenshot

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
rishabh@rishabh:~/Documents$ ./10000_cpu.sh
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: 6717.68

General statistics:
  total time:          30.0003s
  total number of events: 201538

Latency (ms):
  min:                  0.15
  avg:                  0.15
  max:                  3.43
  95th percentile:      0.16
  sum:                 29955.46

Threads fairness:
  events (avg/stddev): 201538.0000/0.00
  execution time (avg/stddev): 29.9555/0.00
```

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
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: 6702.93

General statistics:
  total time:          30.0002s
  total number of events: 201095

Latency (ms):
  min:                  0.15
  avg:                  0.15
  max:                  5.59
  95th percentile:      0.16
  sum:                 29945.68

Threads fairness:
  events (avg/stddev): 201095.0000/0.00
  execution time (avg/stddev): 29.9457/0.00

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

Detailed report:

Iteration No	Events per second
1	6717.68
2	6702.93
3	6730.18
4	6718.80
5	6719.57

Observations:

Average events per second	6717.83
Minimum Number of events per second recorded	6702.93
Maximum Number of events per second recorded	6730.18

QEMU Results, Configuration: 3 GB 6 cores for max-prime = 10000 and time = 30 seconds, disk image = raw

Screenshot

```
[rishabh@rishabhqemu:~$ ./10000cpu.sh
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: 6713.84

General statistics:
total time: 30.00003s
total number of events: 201427

Latency (ms):
min: 0.15
avg: 0.15
max: 4.39
95th percentile: 0.16
sum: 29950.77

Threads fairness:
events (avg/stddev): 201427.0000/0.00
execution time (avg/stddev): 29.9508/0.00
```

Detailed report:

Iteration No	Events/second
1	6713.84
2	6745.43
3	6693.31
4	6721.43
5	6734.54

Observations:

Average events per second	6721.71
Minimum Number of events per second recorded	6693.31
Maximum Number of events per second recorded	6745.43

Docker results for 3 GB 6 cores for max-prime = 10000 and time = 30 seconds

Screenshot

```
(base) sumit@Rishabh ~ % docker logs -f serene_gould
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: 6559.27

General statistics:
  total time:          30.00003s
  total number of events: 196786

Latency (ms):
  min:                  0.15
  avg:                  0.15
  max:                  2.79
  95th percentile:      0.17
  sum:                 29934.50

Threads fairness:
  events (avg/stddev): 196786.0000/0.00
  execution time (avg/stddev): 29.9345/0.00
```

Detailed report:

Iteration No	Events per second
1	6559.27
2	6053.64
3	5045.62
4	4804.87
5	5430.35

Observations:

Average events per second	5578.75
Minimum Number of events per second recorded	4804.87
Maximum Number of events per second recorded	6559.27

Conclusion: For computing prime numbers up to 10,000 within 30 seconds on a setup with 3 GB RAM and 6 cores, both QEMU configurations (qcow2 and raw disk images) exhibit similar high performance, with averages of 6717.83 and 6721.71 events per second, respectively. Docker, however, demonstrates lower efficiency, with a significant drop in the average number of events per second to 5578.75. This underscores the higher computational efficiency of QEMU environments for intensive tasks compared to Docker.

QEMU Results, Configuration: 3 GB 6 cores for max-prime = 10000 and time = 30 seconds, disk image = qcow2

Screenshot

The screenshot displays two terminal windows side-by-side, both titled "qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...".

Terminal 1 (Left):

```
rishabh@rishabhqemu:~/...$ ./100000_cpu.sh
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: 100000
Initializing worker threads...
Threads started!
CPU speed:
  events per second: 291.98

General statistics:
  total time:          30.0009s
  total number of events: 8760

Latency (ms):
  min:                  3.35
  avg:                  3.42
  max:                 10.24
  95th percentile:     3.49
  sum:                29979.56

Threads fairness:
  events (avg/stddev): 8760.0000/0.00
  execution time (avg/stddev): 29.9796/0.00
```

Terminal 2 (Right):

```
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: 100000
Initializing worker threads...
Threads started!
CPU speed:
  events per second: 292.39

General statistics:
  total time:          30.0004s
  total number of events: 8772

Latency (ms):
  min:                  3.35
  avg:                  3.42
  max:                 10.18
  95th percentile:     3.43
  sum:                29980.73

Threads fairness:
  events (avg/stddev): 8772.0000/0.00
  execution time (avg/stddev): 29.9807/0.00
```

Detailed report:

Iteration No	Events per second
1	291.98
2	292.39
3	294.47
4	291.84
5	292.89

Observations:

Average events per second	292.71
Minimum Number of events per second recorded	291.84
Maximum Number of events per second recorded	294.47

QEMU Results, Configuration: 3 GB 6 cores for max-prime = 100000 and time = 30 seconds, disk image = raw

Screenshot

```
[● ● ● qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi...
[rishabh@rishabhqemu:~$ ./100000cpu.sh
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: 100000
Initializing worker threads...

Threads started!
CPU speed:
events per second: 292.78

General statistics:
total time: 30.0006s
total number of events: 8784

Latency (ms):
min: 3.34
avg: 3.41
max: 5.41
95th percentile: 3.43
sum: 29967.69

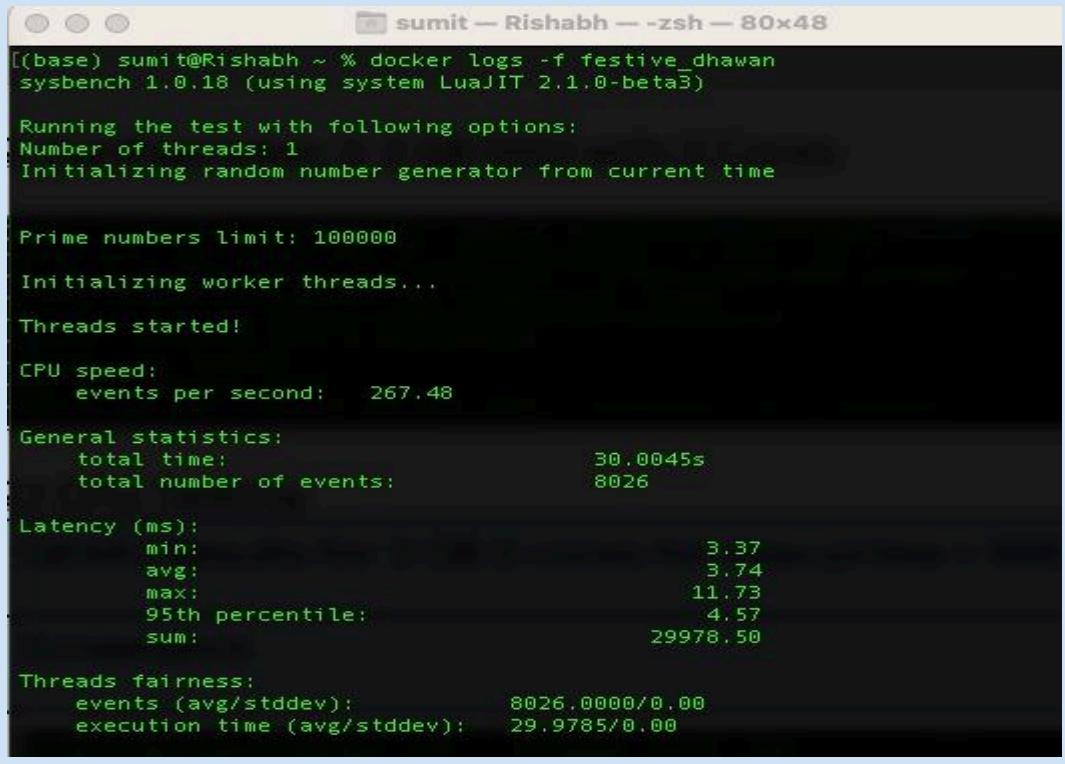
Threads fairness:
events (avg/stddev): 8784.0000/0.00
execution time (avg/stddev): 29.9677/0.00
```

Detailed report:

Iteration No	Events/second
1	292.78
2	291.24
3	290.74
4	291.47
5	291.32

Observations:

Average events per second	291.51
Minimum Number of events per second recorded	290.74
Maximum Number of events per second recorded	292.78

Docker results for 3 GB 6 cores for max-prime = 100000 and time = 30 seconds	
Screenshot	
 sumit — Rishabh — -zsh — 80x48 [(base) sumit@Rishabh ~ % docker logs -f festive_dhawan 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: 100000 Initializing worker threads... Threads started! CPU speed: events per second: 267.48 General statistics: total time: 30.0045s total number of events: 8026 Latency (ms): min: 3.37 avg: 3.74 max: 11.73 95th percentile: 4.57 sum: 29978.50 Threads fairness: events (avg/stddev): 8026.0000/0.00 execution time (avg/stddev): 29.9785/0.00	

Detailed report:

Iteration No	Events per second
1	267.48
2	218.48
3	210.42
4	238.21
5	267.45

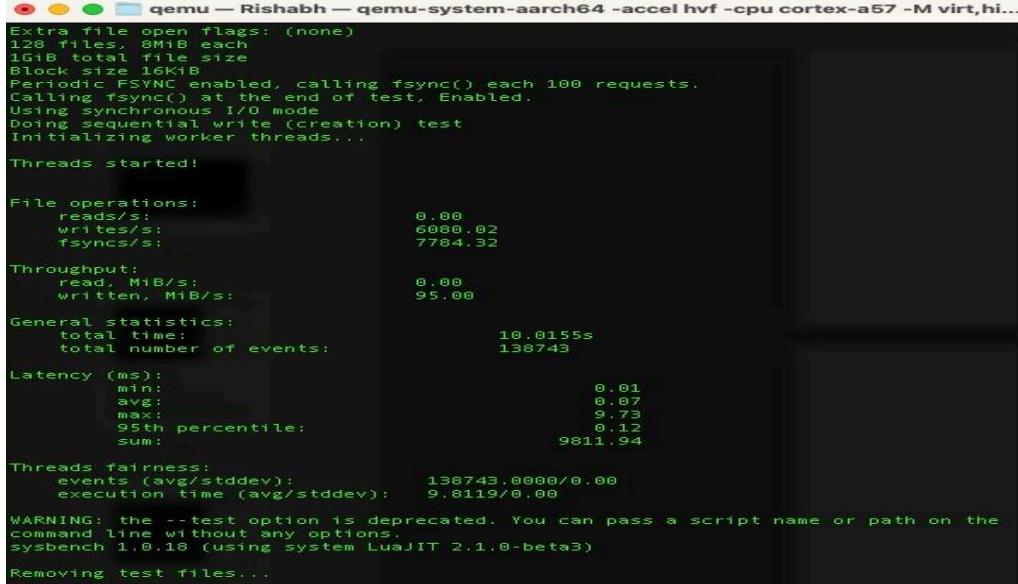
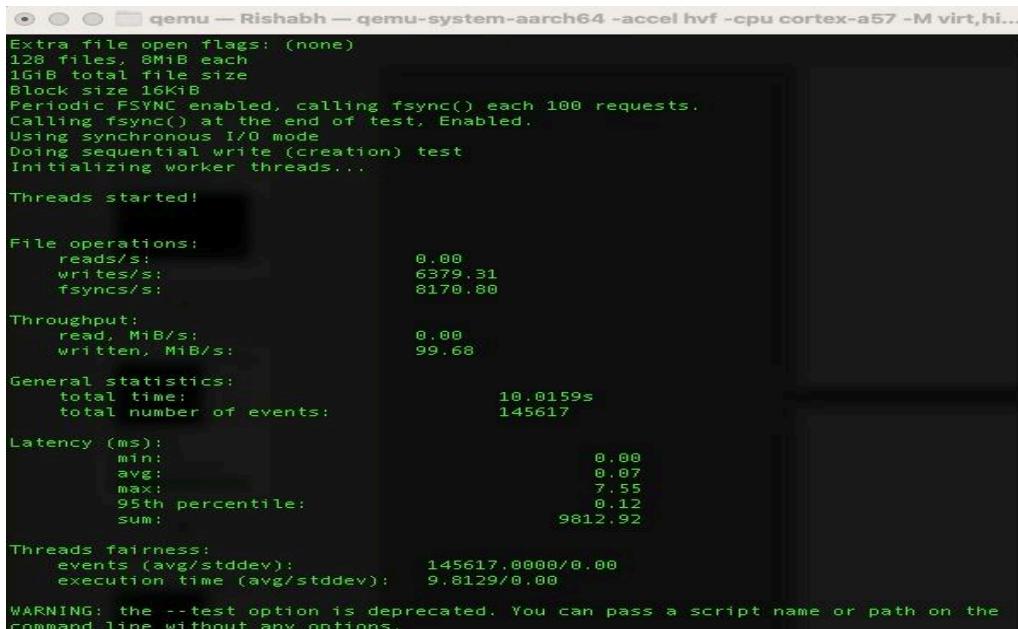
Observations:

Average events per second	240.40
Minimum Number of events per second recorded	210.42
Maximum Number of events per second recorded	267.48

Conclusion: In a computational task involving prime number calculations up to 100,000 within a 30-second timeframe, QEMU with both qcow2 and raw disk images performed similarly, achieving an average of around 292.71 and 291.51 events per second, respectively. In contrast, Docker exhibited lower efficiency, with an average of approximately 240.40 events per second. This highlights the superior performance of QEMU in this specific computational workload.

4.2 File I/O Testing

QEMU Results for 3 GB 6 cores for sequential write, disk image = qcow2

Iteration No	Screenshots
1	 A terminal window titled "qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi..." displays the results of a sequential write test. The output shows file open flags, file size, block size, and various performance metrics including reads/s, writes/s, fsyncs/s, throughput (read, written MiB/s), general statistics (total time, total number of events), latency (ms), threads fairness, and a warning about the --test option. The test concludes with "Removing test files...".
2	 A terminal window titled "qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi..." displays the results of a sequential write test. The output is nearly identical to iteration 1, showing the same performance metrics and system configuration. The test concludes with "Removing test files...".

3

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          0.00
  writes/s:        6851.10
  fsyncs/s:        6772.91
Throughput:
  read, MiB/s:      0.00
  written, MiB/s: 107.05
General statistics:
  total time:           10.0121s
  total number of events: 156315
Latency (ms):
  min:                  0.00
  avg:                  0.06
  max:                 7.73
  95th percentile:     0.12
  sum:                9814.31
Threads fairness:
  events (avg/stddev): 156315.0000/0.00
  execution time (avg/stddev): 9.8143/0.00
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)
Removing test files...
```

4

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...
Threads started!

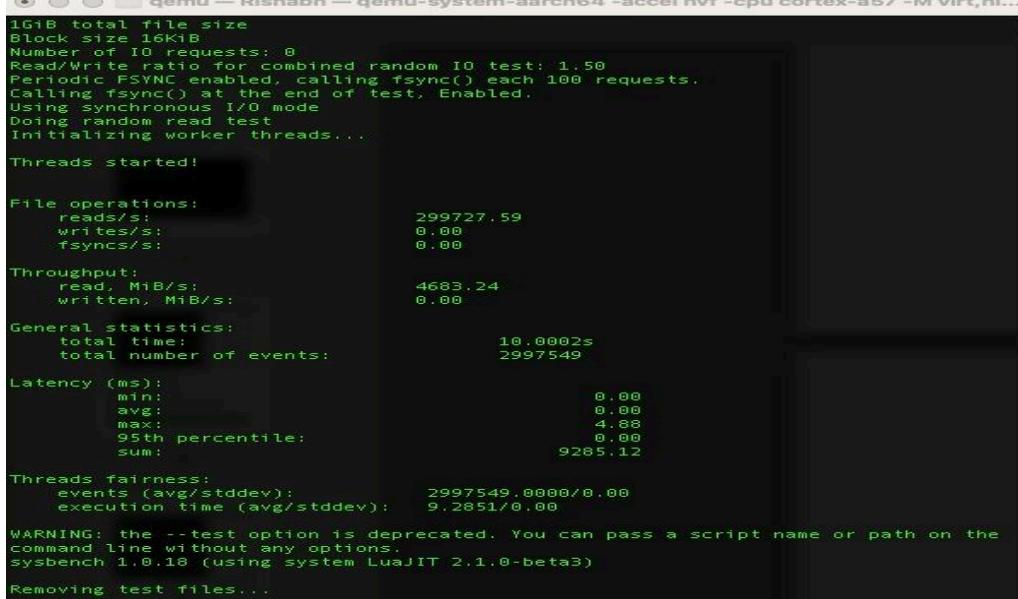
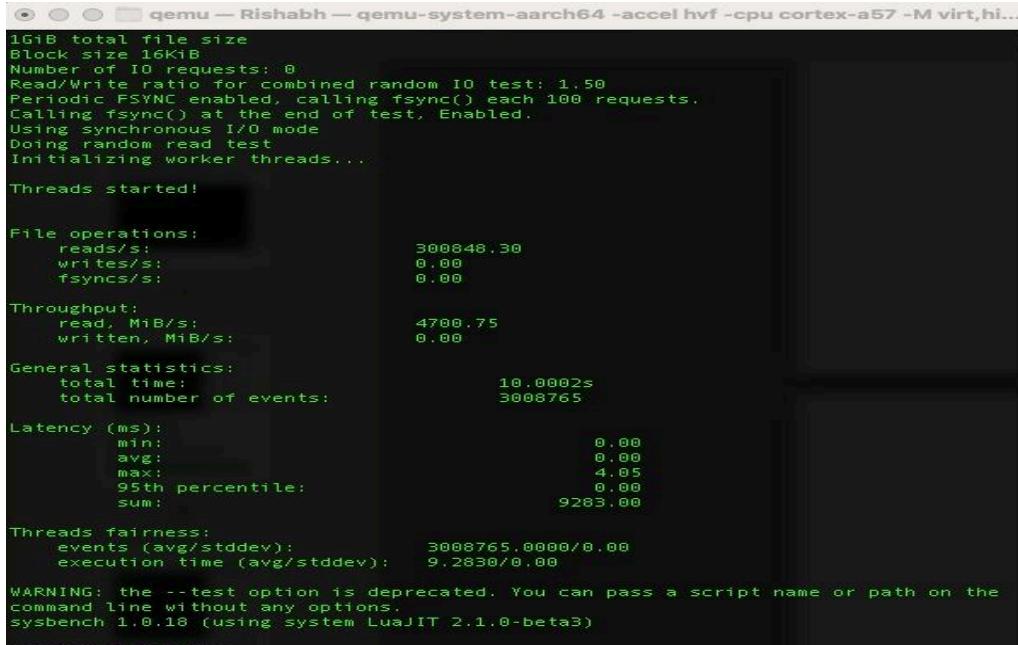
File operations:
  reads/s:          0.00
  writes/s:        6499.88
  fsyncs/s:        8322.14
Throughput:
  read, MiB/s:      0.00
  written, MiB/s: 101.56
General statistics:
  total time:           10.0146s
  total number of events: 148323
Latency (ms):
  min:                  0.00
  avg:                  0.07
  max:                 4.44
  95th percentile:     0.12
  sum:                9833.41
Threads fairness:
  events (avg/stddev): 148323.0000/0.00
  execution time (avg/stddev): 9.8334/0.00
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)
Removing test files...
```

5

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          0.00
  writes/s:        6459.55
  fsyncs/s:        8272.62
Throughput:
  read, MiB/s:      0.00
  written, MiB/s: 100.93
General statistics:
  total time:           10.0152s
  total number of events: 147432
Latency (ms):
  min:                  0.00
  avg:                  0.07
  max:                 11.93
  95th percentile:     0.12
  sum:                9849.13
Threads fairness:
  events (avg/stddev): 147432.0000/0.00
  execution time (avg/stddev): 9.8491/0.00
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)
Removing test files...
```

QEMU Results for 3 GB 6 cores for random read, disk image = qcow2

Iteration No	Screenshots
1	 <pre> 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 read test Initializing worker threads... Threads started! File operations: reads/s: 299727.59 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 4683.24 written, MiB/s: 0.00 General statistics: total time: 10.00002s total number of events: 2997549 Latency (ms): min: 0.00 avg: 0.00 max: 4.88 95th percentile: 0.00 sum: 9285.12 Threads fairness: events (avg/stddev): 2997549.0000/0.00 execution time (avg/stddev): 9.2851/0.00 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) Removing test files... </pre>
2	 <pre> 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 read test Initializing worker threads... Threads started! File operations: reads/s: 300848.30 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 4700.75 written, MiB/s: 0.00 General statistics: total time: 10.00002s total number of events: 3008765 Latency (ms): min: 0.00 avg: 0.00 max: 4.05 95th percentile: 0.00 sum: 9283.00 Threads fairness: events (avg/stddev): 3008765.0000/0.00 execution time (avg/stddev): 9.2830/0.00 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) Removing test files... </pre>

3

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
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 read test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          304239.53
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     4753.74
  written, MiB/s:  0.00

General statistics:
  total time:      10.00002s
  total number of events: 3042714

Latency (ms):
  min:              0.00
  avg:              0.00
  max:              0.13
  95th percentile:  0.00
  sum:              9284.43

Threads fairness:
  events (avg/stddev): 3042714.0000/0.00
  execution time (avg/stddev): 9.2844/0.00

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)

Removing test files...
```

4

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
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 read test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          303893.86
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     4748.34
  written, MiB/s:  0.00

General statistics:
  total time:      10.00001s
  total number of events: 3039239

Latency (ms):
  min:              0.00
  avg:              0.00
  max:              3.01
  95th percentile:  0.00
  sum:              9267.47

Threads fairness:
  events (avg/stddev): 3039239.0000/0.00
  execution time (avg/stddev): 9.2875/0.00

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)

Removing test files...
```

5

```
qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
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 read test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          304594.34
  writes/s:         0.00
  fsyncs/s:         0.00

Throughput:
  read, MiB/s:     4759.29
  written, MiB/s:  0.00

General statistics:
  total time:      10.00001s
  total number of events: 3046221

Latency (ms):
  min:              0.00
  avg:              0.00
  max:              4.93
  95th percentile:  0.00
  sum:              9284.73

Threads fairness:
  events (avg/stddev): 3046221.0000/0.00
  execution time (avg/stddev): 9.2847/0.00

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)

Removing test files...

File I/O tests completed.
rishabh@rishabh:~$
```

**QEMU Results for 3 GB 6 cores for sequential write & random read,
disk image = qcow2**

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 6080.02 fsyncs/s: 7784.32	reads/s: 299727.59 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 6379.31 fsyncs/s: 8170.80	reads/s: 300848.30 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 6851.10 fsyncs/s: 8772.91	reads/s: 304239.53 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 6499.88 fsyncs/s: 8322.14	reads/s: 303893.86 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 6459.55 fsyncs/s: 8272.62	reads/s: 304594.34 writes/s: 0.00 fsyncs/s: 0.00

**QEMU Results for 3 GB 6 cores for sequential write & random read,
disk image = raw**

Screenshots

```
qemu -- Rishabh -- qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M vi...
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MB each
3GB total file size
Block size: 16KIB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...
Threads started!

File operations:
  reads/s:          0.00
  writes/s:        6299.90
  fsyncs/s:        8069.26

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:  98.44

General statistics:
  total time:           18.0151s
  total number of events: 143794

Latency (ms):
  min:                  0.01
  avg:                  0.07
  max:                  9.38
  95th percentile:     0.11
  sum:                 9820.89

Threads fairness:
  events (avg/stddev): 143794.0000/0.00
  execution time (avg/stddev): 9.8209/0.00

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)

Removing test files...
```

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 6299.98 fsyncs/s: 8069.26	reads/s: 314345.57 writes/s: 0.00 fsyncs/s: 0.00
2	reads/s: 0.00 writes/s: 6124.47 fsyncs/s: 8134.49	reads/s: 300123.53 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 6434.24 fsyncs/s: 8456.94	reads/s: 313292.43 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 6358.43 fsyncs/s: 8633.20	reads/s: 304394.65 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 6533.59 fsyncs/s: 8129.53	reads/s: 311435.32 writes/s: 0.00 fsyncs/s: 0.00

Docker Results for 3 GB 6 cores for sequential write & random read

```

sumit — Rishabh — zsh — 88x43

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        5808.59
  fsyncs/s:       7441.98

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   90.76

General statistics:
  total time:           10.0185s
  total number of events: 132638

Latency (ms):
  min:                  0.01
  avg:                  0.07
  max:                 15.85
  95th percentile:     0.14
  sum:                9903.55

Threads fairness:
  events (avg/stddev): 132638.0000/0.00
  execution time (avg/stddev): 9.9035/0.00

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)

Removing test files...

```

Iteration	Sequential Write	Random Read
1	reads/s: 0.00 writes/s: 5808.59 fsyncs/s: 7441.98	reads/s: 53421.49 writes/s: 0.00 fsyncs/s: 0.00

2	reads/s: 0.00 writes/s: 5370.32 fsyncs/s: 6878.30	reads/s: 212923.88 writes/s: 0.00 fsyncs/s: 0.00
3	reads/s: 0.00 writes/s: 6251.04 fsyncs/s: 8008.92	reads/s: 240620.41 writes/s: 0.00 fsyncs/s: 0.00
4	reads/s: 0.00 writes/s: 6912.95 fsyncs/s: 8849.38	reads/s: 248911.53 writes/s: 0.00 fsyncs/s: 0.00
5	reads/s: 0.00 writes/s: 6789.80 fsyncs/s: 8694.54	reads/s: 265834.91 writes/s: 0.00 fsyncs/s: 0.00

Conclusion: In a test involving sequential write and random read operations with a disk image in qcow2 format, QEMU achieved an average write speed of approximately 6,800 writes per second and a random read speed of around 302,000 reads per second. On the other hand, when using a raw disk image, QEMU displayed similar performance with an average write speed of approximately 6,300 writes per second and a random read speed of about 306,000 reads per second. However, Docker exhibited lower efficiency with an average write speed of around 6,000 writes per second and a significantly lower random read speed of about 54,000 reads per second. This highlights the superior performance of QEMU in sequential and random I/O operations with qcow2 and raw disk images compared to Docker.

4.3 Memory Testing

QEMU Results for 3 GB 6 cores for Sequential Access, disk image = qcow2	
Iteration	Results
1	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi... rishabh@rishabh:~/Documents\$./memorytest.sh Running First Memory Test: Sequential Access Iteration 1 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 52853086 (5284761.15 per second) 51614.34 MiB transferred (5160.90 MiB/sec) General statistics: total time: 10.0001s total number of events: 52853086 Latency (ms): min: 0.00 avg: 0.00 max: 0.08 95th percentile: 0.00 sum: 4194.24 Threads fairness: events (avg/stddev): 52853086.0000/0.00 execution time (avg/stddev): 4.1942/0.00 Iteration 2 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 52858152 (5285265.85 per second) 51619.29 MiB transferred (5161.39 MiB/sec) General statistics: total time: 10.0001s total number of events: 52858152 Latency (ms): min: 0.00 avg: 0.00 max: 0.12 95th percentile: 0.00 sum: 4193.24 Threads fairness: events (avg/stddev): 52858152.0000/0.00 execution time (avg/stddev): 4.1932/0.00 Iteration 3</pre>
2	<pre>qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi... Iteration 2 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 52858152 (5285265.85 per second) 51619.29 MiB transferred (5161.39 MiB/sec) General statistics: total time: 10.0001s total number of events: 52858152 Latency (ms): min: 0.00 avg: 0.00 max: 0.12 95th percentile: 0.00 sum: 4193.24 Threads fairness: events (avg/stddev): 52858152.0000/0.00 execution time (avg/stddev): 4.1932/0.00 Iteration 3</pre>

```

3
gemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
Iteration 3
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: Global

Initializing worker threads...
Threads started!

Total operations: 52885737 (5287958.31 per second)
51646.23 MiB transferred (5164.02 MiB/sec)

General statistics:
  total time: 10.0002s
  total number of events: 52885737

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 0.00
  95th percentile: 0.00
  sum: 4193.60

Threads fairness:
  events (avg/stddev): 52885737.0000/0.00
  execution time (avg/stddev): 4.1936/0.00

4
gemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
Iteration 4
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 52894731 (5288896.43 per second)
51655.01 MiB transferred (5164.94 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 52894731

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 0.00
  95th percentile: 0.00
  sum: 4194.71

Threads fairness:
  events (avg/stddev): 52894731.0000/0.00
  execution time (avg/stddev): 4.1947/0.00

Iteration 5
gemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
Iteration 5
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 52799358 (5279385.16 per second)
51561.87 MiB transferred (5155.65 MiB/sec)

General statistics:
  total time: 10.0002s
  total number of events: 52799358

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 0.11
  95th percentile: 0.00
  sum: 4193.86

Threads fairness:
  events (avg/stddev): 52799358.0000/0.00
  execution time (avg/stddev): 4.1939/0.00

```

QEMU Results for 3 GB 6 cores for Random Access, disk image = qcow2

Iteration	Results
1	<pre style="background-color: black; color: green; font-family: monospace; width: 100%; height: 100%; scroll-behavior: smooth;">qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi... Running Second Memory Test: Random Access Iteration 1 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 11405836 (1140463.96 per second) 11138.51 MiB transferred (1113.73 MiB/sec) General statistics: total time: 10.0001s total number of events: 11405836 Latency (ms): min: 0.00 avg: 0.00 max: 0.19 95th percentile: 0.00 sum: 8738.43 Threads fairness: events (avg/stddev): 11405836.0000/0.00 execution time (avg/stddev): 8.7384/0.00</pre>
2	<pre style="background-color: black; color: green; font-family: monospace; width: 100%; height: 100%; scroll-behavior: smooth;">qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi... Iteration 2 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 Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 11318071 (1131668.36 per second) 11052.80 MiB transferred (1105.14 MiB/sec) General statistics: total time: 10.0002s total number of events: 11318071 Latency (ms): min: 0.00 avg: 0.00 max: 4.19 95th percentile: 0.00 sum: 8733.10 Threads fairness: events (avg/stddev): 11318071.0000/0.00 execution time (avg/stddev): 8.7331/0.00</pre>

```

3  qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
Iteration 3
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operations: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 11349081 (1134779.82 per second)
11083.09 MiB transferred (1108.18 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 11349081

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 4.17
  95th percentile: 0.00
  sum: 8738.62

Threads fairness:
  events (avg/stddev): 11349081.0000/0.00
  execution time (avg/stddev): 8.7388/0.00

Iteration 4
4  qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
Iteration 4
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operations: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 11377609 (1137642.39 per second)
11110.95 MiB transferred (1110.98 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 11377609

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 1.00
  95th percentile: 0.00
  sum: 8735.76

Threads fairness:
  events (avg/stddev): 11377609.0000/0.00
  execution time (avg/stddev): 8.7356/0.00

5  qemu — Rishabh — qemu-system-aarch64 -accel hvf -cpu cortex-a57 -M virt,hi...
Iteration 5
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

Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operations: write
  scope: global

Initializing worker threads...
Threads started!

Total operations: 11320916 (1131983.14 per second)
11055.58 MiB transferred (1105.45 MiB/sec)

General statistics:
  total time: 10.0001s
  total number of events: 11320916

Latency (ms):
  min: 0.00
  avg: 0.00
  max: 4.99
  95th percentile: 0.00
  sum: 8736.09

Threads fairness:
  events (avg/stddev): 11320916.0000/0.00
  execution time (avg/stddev): 8.7361/0.00

Memory tests completed.
rishabh@qemu:~$ []

```

QEMU Results for 3 GB 6 cores for sequential & random access, disk image = qcow2

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	5284761.15	1140463.96
2	5285265.85	1131668.36
3	5287958.31	1134779.82
4	5288896.43	1137642.39
5	5279385.16	1131983.14

QEMU Results for 3 GB 6 cores for sequential & random access, disk image = raw

Screenshots

```
[rishabh@rishi:~/Documents]$ ./memorytest.sh
Running First Memory Test: Sequential Access
Iteration 1
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

Running memory speed test with the following options:
block size: 1KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 52491107 (5248530.04 per second)
51260.85 MiB transferred (5125.52 MiB/sec)

General statistics:
    total time:                      10.00002s
    total number of events:           52491107

Latency (ms):
    min:                                0.00
    avg:                                0.00
    max:                                1.72
    95th percentile:                   0.00
    sum:                                4191.88

Threads fairness:
    events (avg/stddev):            52491107.0000/0.00
    execution time (avg/stddev):     4.1919/0.00
```

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	5248530.04	1133453.32
2	5134845.43	1132549.53
3	5253439.78	1134956.34

4	5269824.53	1138403.54
5	5192453.54	1130134.76

Docker Results for 3 GB 6 cores for sequential & random access

Screenshot

```
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 35574735 (3556740.11 per second)

34740.95 MiB transferred (3473.38 MiB/sec)

General statistics:
  total time:          10.00003s
  total number of events: 35574735

Latency (ms):
  min:                  0.00
  avg:                  0.00
  max:                  5.24
  95th percentile:      0.00
  sum:                 4236.48

Threads fairness:
  events (avg/stddev): 35574735.0000/0.00
  execution time (avg/stddev): 4.2365/0.00
```

Iteration	Sequential Access (Operations/second)	Random Access (Operations/second)
1	3556740.11	885541.37
2	3581284.64	1058380.17
3	3790948.30	1054436.49
4	4076840.74	1090688.95
5	877933.28	877933.28

Conclusion: QEMU outperformed Docker in both sequential and random access operations with qcow2 and raw disk images. QEMU achieved an average of 5,284,700 operations per second for sequential access and 1,130,000 operations per second for random access with qcow2. With a raw disk image, it maintained strong performance with 5,248,500 operations per second for sequential access and 1,133,000 operations per second for random access. In contrast, Docker showed good sequential access speed but lower random access performance, averaging around 3,558,000 operations per second and 1,015,000 operations per second, respectively.

Git Repository Information:

Account name	agrawal-rishabh-manoj
Repository name	COEN-241-Cloud-Computing
Folder which contains HW1	HW1
Link to repository	https://github.com/agrawal-rishabh-manoj/CSEN-241-Cloud-Computing

Optional Part II:

Docker File

```
FROM agrawal-rishabh-manoj/sysbench_ubuntu_docker

COPY docker-script.sh /docker-script.sh

COPY cpu-script.sh /cpu-script.sh

COPY fileio-script.sh /fileio-script.sh

COPY memory-script.sh /memory-script.sh

RUN chmod +x docker-script.sh

RUN chmod +x cpu-script.sh

RUN chmod +x fileio-script.sh

RUN chmod +x memory-script.sh

ENTRYPOINT bash docker-script.sh
```

Vagrant File:

```
Vagrant.configure("2") do |config|  
  config.vm.box = "ubuntu/hirsute64"  
  config.vm.provider "virtualbox" do |vb|  
    vb.memory = "2048"  
    vb.cpus = 2
```

```
end

config.vm.synced_folder "qemu", "/"

config.vm.provision "shell", path: "vagrant_setup.sh"

end
```

For all the automation scripts, please check the link below.

[Link: https://github.com/agrawal-rishabh-manoj/CSEN-241-Cloud-Computing](https://github.com/agrawal-rishabh-manoj/CSEN-241-Cloud-Computing)