

## COEN241 - Cloud Computing

### HW #1

#### 1. Detailed configurations:

##### > Host System configuration:

MacBook Pro  
OS = MacOS Catalina  
CPU = 2.9 GHz Dual-Core Intel Core i5  
Total no of Cores = 2  
RAM = 16GB  
Storage = 500GB

##### > QEMU configuration:

```
QEMU
karthik@karthik:~$ inxi -Fxz
System:   Kernel: 5.4.0-107-generic x86_64 bits: 64 compiler: gcc v: 9.4.0 Console: tty 1
          Distro: Ubuntu 20.04.4 LTS (Focal Fossa)
Machine:  Type: Qemu System: QEMU product: Standard PC (i440FX + PIIX, 1996) v: pc-i440fx-6.2 serial: <filter>
          Mobo: N/A model: N/A serial: N/A BIOS: SeaBIOS v: rel-1.15.0-0-g2dd4b9b3f840-prebuilt.qemu.org date: 04/01/2014
CPU:      Topology: Single Core model: QEMU Virtual version 2.5+ bits: 64 type: UP arch: K8 rev.F+ rev: 1 L2 cache: 512 KiB
          flags: lm nx pae sse sse2 sse3 svm bogomips: 5799
          Speed: 2900 MHz min/max: N/A Core speed (MHz): 1: 2900
Graphics: Device-1: vendor: Red Hat driver: bochs-drm v: N/A bus ID: 00:02.0
          Display: server: No display server data found. Headless machine? tty: 128x48
          Message: Advanced graphics data unavailable in console. Try -G --display
Audio:    Message: No Device data found.
Network:  Device-1: Intel 82371AB/EB/MB PIIX4 ACPI vendor: Red Hat Qemu virtual machine type: network bridge
          driver: piix4_smbus v: N/A port: c040 bus ID: 00:01.3
          Device-2: Intel 82540EM Gigabit Ethernet vendor: Red Hat QEMU Virtual Machine driver: e1000 v: 7.3.21-k8-NAPI
          port: c000 bus ID: 00:03.0
          IF: ens3 state: up speed: 1000 Mbps duplex: full mac: <filter>
          IF-ID-1: docker0 state: down mac: <filter>
Drives:   Local Storage: total: 20.00 GiB used: 8.12 GiB (40.6%)
          ID-1: /dev/sda vendor: QEMU model: HARDDISK size: 20.00 GiB
Partition: ID-1: / size: 19.56 GiB used: 8.12 GiB (41.5%) fs: ext4 dev: /dev/sda2
Sensors:   Message: No sensors data was found. Is sensors configured?
Info:     Processes: 110 Uptime: 1d 8h 40m Memory: 1.94 GiB used: 287.0 MiB (14.5%) Init: systemd runlevel: 5 Compilers:
          gcc: N/A Shell: bash v: 5.0.17 inxi: 3.0.38
karthik@karthik:~$ _
```

OS = Ubuntu 20.04  
CPU = QEMU  
Total no of Cores = 1  
RAM = 2GB  
Storage = 20GB

> **Docker configuration:**

OS = Ubuntu 20.04  
Total no of Cores = 1  
RAM = 2GB  
Storage = 20GB

## 2.

### 2.1 Main Steps to enable a QEMU VM

- Install homebrew using Terminal by using below command and add it to path variable
  - a. `$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
  - b. `$ export PATH="/opt/homebrew/bin:/usr/local/bin:$PATH"`
- After homebrew is installed, install qemu using homebrew  
`$ brew install qemu`
- Now, download the Ubuntu ISO file from the following link  
<https://releases.ubuntu.com/focal/ubuntu-20.04.4-live-server-amd64.iso>
- Create the virtual disk for VM with size = 20GB (Storage)  
`$ qemu-img create -f raw -o size=20G Ubuntu.img`
- Now, get the path for the downloaded Ubuntu ISO path and replace it with the bold letters below and run the command. Follow the steps and this will install the Ubuntu VM on host.  
`$ qemu-system-x86_64 -boot d -cdrom ISO FILE PATH -m 2048M -hda Ubuntu.img`
- Next, boot the installed Ubuntu OS, replace the “-boot d” argument with “-boot c”  
`$ qemu-system-x86_64 -boot c -cdrom ISO FILE PATH -m 2048M -hda Ubuntu.img`
- If everything is fine you will see the Ubuntu virtual machine running on Mac OS with Hypervisor (Type 1) as QEMU.

### 2.2 QEMU Commands

Source : <https://www.mankier.com/1/qemu>

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

- **-smp[[cpus=n][,maxcpus=maxcpus][,sockets=sockets][,dies=dies][,cores=cores][,threads=threads]]**

Simulate a SMP system with 'n' CPUs initially present on the machine type board. On boards supporting CPU hotplug, the optional 'maxcpus' parameter can be set to enable further CPUs to be added at runtime. When both parameters are omitted, the maximum number of CPUs will be calculated from the provided topology members and the initial CPU count will match the maximum number.

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

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

- **-cpu model**

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

- **-monitor dev**

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

- **-cdrom file**

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

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

Add device driver. prop=value sets driver properties. Valid properties depend on the driver. To get help on possible drivers and properties, use -device help and -device driver,help.

- **-drive option[,option[,option[,...]]]**

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

- **-serial dev**

Redirect the virtual serial port to host character device dev. The default device is vc in graphical mode and stdio in non graphical mode. This option can be used several times to simulate up to 4 serial ports. Use -serial none to disable all serial ports.

- **-vga type**

Select type of VGA card to emulate. Valid values for type are cirrus, std, vmware etc.

3.

### **3.1 Main Steps to enable a Docker**

- Go to Docker website with this link - <https://docs.docker.com/desktop/mac/install/>  
And download the Docker desktop for Mac
- Install the Docker desktop on to your machine and add it to your applications
- Once you run the Docker application the Docker engine starts
- Now we have to pull the Docker image for Sysbench tool from Docker Hub  
`$ docker pull zyclonite/sysbench`
- Create the container for it and run the image  
`$ docker run -it csminpp/ubuntu-sysbench`
- Now, if everything is good you should see the docker running and able to use sysbench

### **3.2 Docker Operations**

**start** - Run an existing Docker container

**create** - Create a Docker container from an image

**run** - Create a new Docker container and start running it

**ls** - List all the containers running currently

**stop** - Stop running containers

**rm** - Remove a container which is not running

**kill** - Kill one or more containers

**logs** - Retrieves logs present at the time of execution

**inspect** - Get more information about the Containers / Images

**build** - Build Docker image from a Dockerfile

**push** - Push an image or a repository to a registry

**ls** - List all the Images

**rm** - Delete the image

**docker login** - Log in to Docker registry to get Images

**docker version** - Get more information about the Docker client and server versions

**docker system prune** - Remove all unused data - like containers, networks, images etc

## **4. Benchmarking Experiments:**

Sysbench tool is used to perform the test on system parameters such as file io performance, scheduler performance, memory allocation, database server performance and much more. We will use the same to test our System Virtualization (QEMU) and OS Virtualization (Docker).

### **4.1 Experiment #1 - CPU Performance:**

We can test on the CPU performance by calculating the prime numbers upto a max value parameter called –cpu-max-primes.

#### **1a] Testing on QEMU:**

```
> $ sysbench --test=cpu --cpu-max-prime=10000 run
```

#### **Test Run #1**

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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

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

CPU speed:
  events per second: 234.42

General statistics:
  total time:          10.0048s
  total number of events: 2348

Latency (ms):
  min:                2.93
  avg:                4.24
  max:                14.83
  95th percentile:    5.37
  sum:               9951.19

Threads fairness:
  events (avg/stddev): 2348.0000/0.00
  execution time (avg/stddev): 9.9512/0.00

karthik@karthik:~$
```

## Test Run #2

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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

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

CPU speed:
events per second: 235.29

General statistics:
total time: 10.0025s
total number of events: 2355

Latency (ms):
min: 2.93
avg: 4.22
max: 10.46
95th percentile: 5.47
sum: 9946.85

Threads fairness:
events (avg/stddev): 2355.0000/0.00
execution time (avg/stddev): 9.9469/0.00

karthik@karthik:~$
```

## Test Run #3

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuAJIT 2.1.0-beta3)

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

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

CPU speed:
events per second: 227.23

General statistics:
total time: 10.0037s
total number of events: 2275

Latency (ms):
min: 2.93
avg: 4.37
max: 18.93
95th percentile: 5.57
sum: 9952.19

Threads fairness:
events (avg/stddev): 2275.0000/0.00
execution time (avg/stddev): 9.9522/0.00

karthik@karthik:~$ _
```

## Test Run #4

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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

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

CPU speed:
events per second: 228.03

General statistics:
total time: 10.0029s
total number of events: 2283

Latency (ms):
min: 2.93
avg: 4.36
max: 13.37
95th percentile: 5.67
sum: 9944.79

Threads fairness:
events (avg/stddev): 2283.0000/0.00
execution time (avg/stddev): 9.9448/0.00

karthik@karthik:~$
```

## Test Run #5

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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

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

CPU speed:
events per second: 221.57

General statistics:
total time: 10.0045s
total number of events: 2219

Latency (ms):
min: 2.92
avg: 4.48
max: 17.61
95th percentile: 6.21
sum: 9943.61

Threads fairness:
events (avg/stddev): 2219.0000/0.00
execution time (avg/stddev): 9.9436/0.00

karthik@karthik:~$ _
```

## 1b] Testing on Docker:

```
> $ sysbench --test=cpu --cpu-max-prime=10000 run
```

### Test Run #1

```
/ # sysbench --test=cpu --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 378.32

General statistics:
  total time:          10.0031s
  total number of events: 3785

Latency (ms):
  min:                1.07
  avg:                2.64
  max:                44.00
  95th percentile:   11.04
  sum:               9973.73

Threads fairness:
  events (avg/stddev): 3785.0000/0.00
  execution time (avg/stddev): 9.9737/0.00

/ # █
```

## Test Run #2

```
[/ # sysbench --test(cpu) --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 369.39

General statistics:
  total time:          10.0012s
  total number of events: 3695

Latency (ms):
  min:                 1.08
  avg:                 2.70
  max:                 80.50
  95th percentile:    11.24
  sum:                9991.12

Threads fairness:
  events (avg/stddev): 3695.0000/0.00
  execution time (avg/stddev): 9.9911/0.00
```

## Test Run #3

```
[/ # sysbench --test=cpu --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 368.17

General statistics:
  total time:          10.0017s
  total number of events: 3683

Latency (ms):
  min:                 1.08
  avg:                 2.71
  max:                 53.57
  95th percentile:    11.04
  sum:                9977.80

Threads fairness:
  events (avg/stddev): 3683.0000/0.00
  execution time (avg/stddev): 9.9778/0.00

/ # █
```

## Test Run #4

```
[/ # sysbench --test=cpu --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 299.60

General statistics:
  total time:          10.0014s
  total number of events: 2997

Latency (ms):
  min:                 1.08
  avg:                 3.33
  max:                 99.59
  95th percentile:    12.30
  sum:                 9987.95

Threads fairness:
  events (avg/stddev): 2997.0000/0.00
  execution time (avg/stddev): 9.9879/0.00

/ # ]
```

## Test Run #5

```
// # sysbench --test=cpu --cpu-max-prime=10000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 357.30

General statistics:
  total time:          10.0011s
  total number of events: 3574

Latency (ms):
  min:                 1.11
  avg:                 2.79
  max:                 80.33
  95th percentile:    11.24
  sum:                9976.44

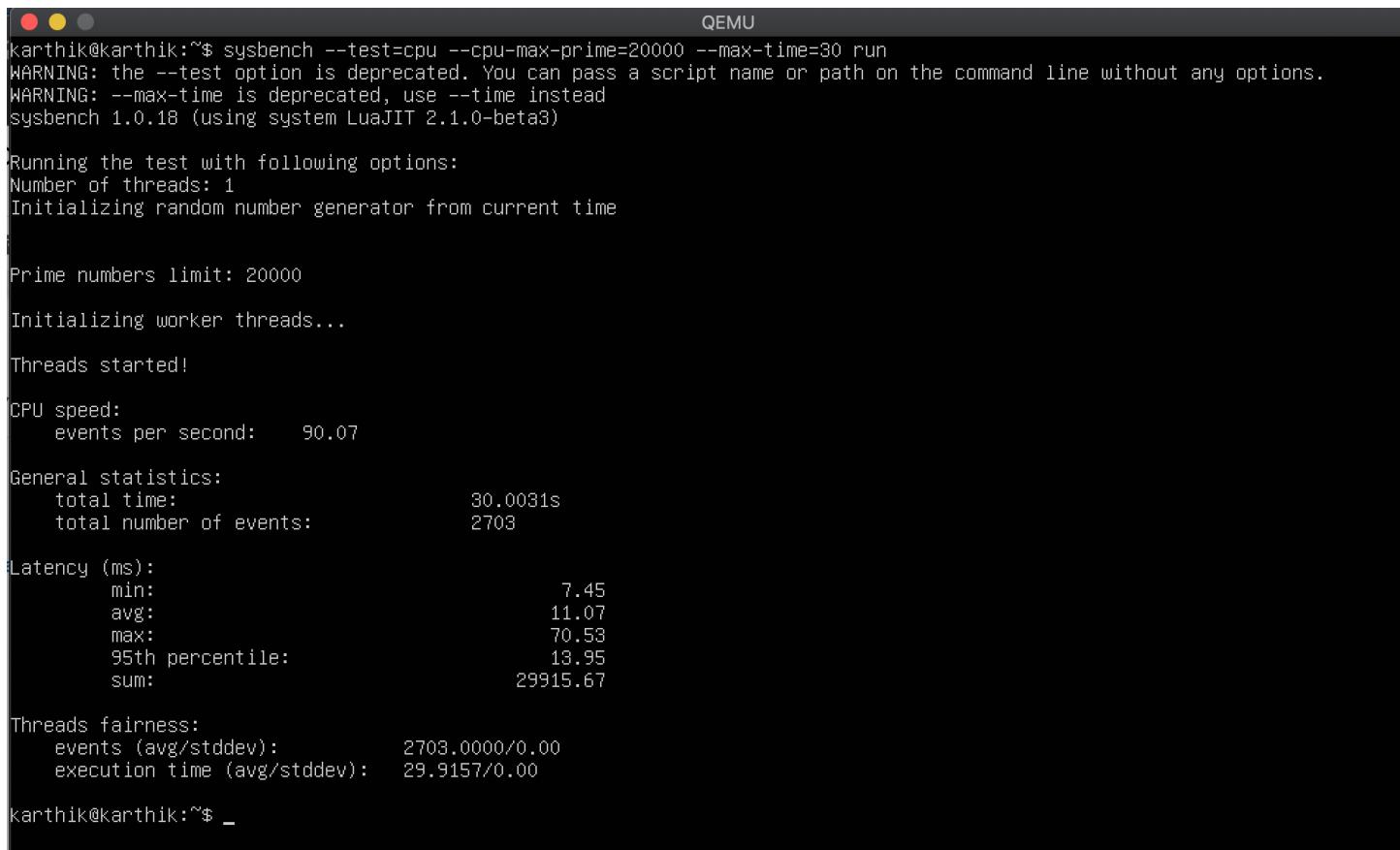
Threads fairness:
  events (avg/stddev): 3574.0000/0.00
  execution time (avg/stddev): 9.9764/0.00

/ # █
```

## **2a] Testing on QEMU:**

```
> $ sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
```

### **Test Run #1**



```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
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: 20000
Initializing worker threads...

Threads started!

CPU speed:
events per second: 90.07

General statistics:
total time: 30.0031s
total number of events: 2703

Latency (ms):
min: 7.45
avg: 11.07
max: 70.53
95th percentile: 13.95
sum: 29915.67

Threads fairness:
events (avg/stddev): 2703.0000/0.00
execution time (avg/stddev): 29.9157/0.00

karthik@karthik:~$ _
```

## Test Run #2

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
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: 20000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 89.30

General statistics:
  total time:          30.0081s
  total number of events: 2680

Latency (ms):
  min:                 7.45
  avg:                11.17
  max:                134.42
  95th percentile:    13.95
  sum:               29925.98

Threads fairness:
  events (avg/stddev): 2680.0000/0.00
  execution time (avg/stddev): 29.9260/0.00

karthik@karthik:~$ _
```

## Test Run #3

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
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: 20000

Initializing worker threads...
Threads started!

CPU speed:
  events per second: 91.01

General statistics:
  total time:          30.0131s
  total number of events: 2732

Latency (ms):
  min:                 7.47
  avg:                10.96
  max:                22.55
  95th percentile:    13.46
  sum:               29931.97

Threads fairness:
  events (avg/stddev): 2732.0000/0.00
  execution time (avg/stddev): 29.9320/0.00

karthik@karthik:~$ _
```

## Test Run #4

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
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: 20000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 91.18

General statistics:
  total time:          30.0116s
  total number of events: 2737

Latency (ms):
  min:                 7.52
  avg:                10.93
  max:                26.75
  95th percentile:    13.70
  sum:               29926.14

Threads fairness:
  events (avg/stddev): 2737.0000/0.00
  execution time (avg/stddev): 29.9261/0.00

karthik@karthik:~$ _
```

## Test Run #5

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
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: 20000
Initializing worker threads...
Threads started!

CPU speed:
  events per second: 91.02

General statistics:
  total time:          30.0057s
  total number of events: 2732

Latency (ms):
  min:                 7.51
  avg:                10.95
  max:                33.72
  95th percentile:    13.70
  sum:               29920.71

Threads fairness:
  events (avg/stddev): 2732.0000/0.00
  execution time (avg/stddev): 29.9207/0.00

karthik@karthik:~$ _
```

## 2b] Testing on Docker:

```
> $ sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
```

### Test Run #1

```
/ # sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 144.56

General statistics:
  total time:          30.0006s
  total number of events: 4337

Latency (ms):
  min:                2.68
  avg:                6.91
  max:              130.23
  95th percentile:    18.95
  sum:            29968.74

Threads fairness:
  events (avg/stddev): 4337.0000/0.00
  execution time (avg/stddev): 29.9687/0.00

/ # █
```

## Test Run #2

```
[/ # sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 146.35

General statistics:
  total time:          30.0008s
  total number of events: 4391

Latency (ms):
  min:                2.69
  avg:                6.82
  max:                84.83
  95th percentile:    18.28
  sum:               29952.21

Threads fairness:
  events (avg/stddev): 4391.0000/0.00
  execution time (avg/stddev): 29.9522/0.00

/ # ]
```

## Test Run #3

```
./ # sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 144.82

General statistics:
  total time:          30.0007s
  total number of events: 4345

Latency (ms):
  min:                2.77
  avg:                6.90
  max:                62.11
  95th percentile:    19.29
  sum:               29973.68

Threads fairness:
  events (avg/stddev): 4345.0000/0.00
  execution time (avg/stddev): 29.9737/0.00

/ # █
```

## Test Run #4

```
[/ # sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 151.07

General statistics:
  total time:          30.0051s
  total number of events: 4533

Latency (ms):
  min:                2.69
  avg:                6.61
  max:                48.14
  95th percentile:    17.32
  sum:               29981.58

Threads fairness:
  events (avg/stddev): 4533.0000/0.00
  execution time (avg/stddev): 29.9816/0.00

/ # ]
```

## Test Run #5

```
/ # sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 151.01

General statistics:
  total time:          30.0024s
  total number of events: 4531

Latency (ms):
  min:                 2.69
  avg:                 6.61
  max:                132.73
  95th percentile:    17.95
  sum:                29953.82

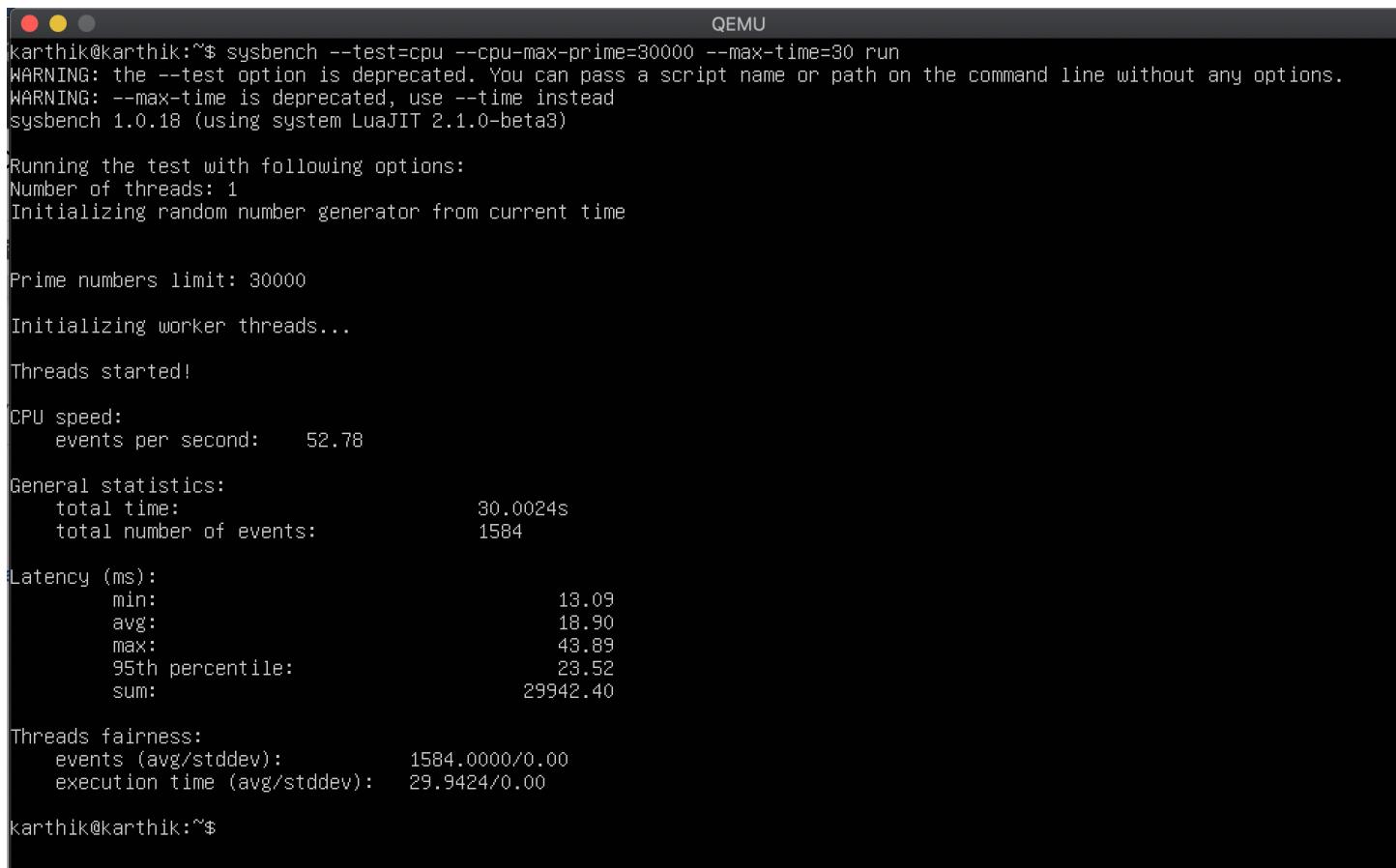
Threads fairness:
  events (avg/stddev): 4531.0000/0.00
  execution time (avg/stddev): 29.9538/0.00

/ # █
```

### **3a] Testing on QEMU:**

```
> $ sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
```

#### **Test Run #1**



```
Karthik@Karthik:~$ sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.18 (using system LuAJIT 2.1.0-beta3)

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

Prime numbers limit: 30000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 52.78

General statistics:
total time: 30.0024s
total number of events: 1584

Latency (ms):
min: 13.09
avg: 18.90
max: 43.89
95th percentile: 23.52
sum: 29942.40

Threads fairness:
events (avg/stddev): 1584.0000/0.00
execution time (avg/stddev): 29.9424/0.00

Karthik@Karthik:~$
```

## Test Run #2

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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

Prime numbers limit: 30000
Initializing worker threads...

Threads started!

CPU speed:
    events per second: 51.17

General statistics:
    total time:          30.0105s
    total number of events: 1536

Latency (ms):
    min:                 13.21
    avg:                 19.50
    max:                 112.96
    95th percentile:     23.95
    sum:                29955.18

Threads fairness:
    events (avg/stddev): 1536.0000/0.00
    execution time (avg/stddev): 29.9552/0.00

karthik@Karthik:~$
```

### Test Run #3

```
QEMU
karthik@karthik:~$ sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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

Prime numbers limit: 30000
Initializing worker threads...

Threads started!

CPU speed:
  events per second: 50.48

General statistics:
  total time:          30.0040s
  total number of events: 1515

Latency (ms):
  min:                13.29
  avg:                19.76
  max:                69.52
  95th percentile:    24.38
  sum:               29940.61

Threads fairness:
  events (avg/stddev): 1515.0000/0.00
  execution time (avg/stddev): 29.9406/0.00

karthik@karthik:~$ _
```

## Test Run #4

```
QEMU
Karthik@Karthik:~$ sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.18 (using system LuAJIT 2.1.0-beta3)

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

Prime numbers limit: 30000
Initializing worker threads...
Threads started!

CPU speed:
  events per second: 52.23

General statistics:
  total time:          30.0098s
  total number of events: 1568

Latency (ms):
  min:                 18.12
  avg:                 19.10
  max:                 50.69
  95th percentile:    28.52
  sum:                29953.33

Threads fairness:
  events (avg/stddev): 1568.0000/0.00
  execution time (avg/stddev): 29.9533/0.00

Karthik@Karthik:~$
```

## Test Run #5

```
QEMU
Karthik@Karthik:~$ sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.18 (using system LuAJIT 2.1.0-beta3)

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

Prime numbers limit: 30000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 52.75

General statistics:
total time: 30.0207s
total number of events: 1584

Latency (ms):
min: 13.08
avg: 18.91
max: 30.56
95th percentile: 23.52
sum: 29960.59

Threads fairness:
events (avg/stddev): 1584.0000/0.00
execution time (avg/stddev): 29.9606/0.00

Karthik@Karthik:~$ _
```

### 3b] Testing on Docker:

```
> $ sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
```

### Test Run #1

```
[// # sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 60.02

General statistics:
  total time: 30.0071s
  total number of events: 1801

Latency (ms):
  min: 4.84
  avg: 16.65
  max: 157.95
  95th percentile: 45.79
  sum: 29986.48

Threads fairness:
  events (avg/stddev): 1801.0000/0.00
  execution time (avg/stddev): 29.9865/0.00

/ # ]
```

## Test Run #2

```
[/ # sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 87.90

General statistics:
  total time:          30.0208s
  total number of events: 2639

Latency (ms):
  min:                 4.61
  avg:                11.37
  max:                73.71
  95th percentile:    23.52
  sum:               30001.72

Threads fairness:
  events (avg/stddev): 2639.0000/0.00
  execution time (avg/stddev): 30.0017/0.00

/ # ]
```

## Test Run #3

```
./ # sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 85.59

General statistics:
  total time:          30.0149s
  total number of events: 2569

Latency (ms):
  min:                 4.60
  avg:                 11.67
  max:                 92.74
  95th percentile:    24.38
  sum:                29976.19

Threads fairness:
  events (avg/stddev): 2569.0000/0.00
  execution time (avg/stddev): 29.9762/0.00

/ # █
```

## Test Run #4

```
[/ # sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:    78.10

General statistics:
  total time:          30.0120s
  total number of events: 2344

Latency (ms):
  min:                  4.80
  avg:                 12.80
  max:                107.14
  95th percentile:     28.67
  sum:                29997.77

Threads fairness:
  events (avg/stddev): 2344.0000/0.00
  execution time (avg/stddev): 29.9978/0.00

/ # ]
```

## Test Run #5

```
[/ # sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:    66.15

General statistics:
  total time:          30.0048s
  total number of events: 1985

Latency (ms):
  min:                 4.79
  avg:                15.11
  max:                187.44
  95th percentile:    51.94
  sum:               29993.07

Threads fairness:
  events (avg/stddev): 1985.0000/0.00
  execution time (avg/stddev): 29.9931/0.00

/ # ]
```

## **4.2 Experiment #2 - File I/O:**

File I/O tests can be used to produce various kinds of file I/O workloads.

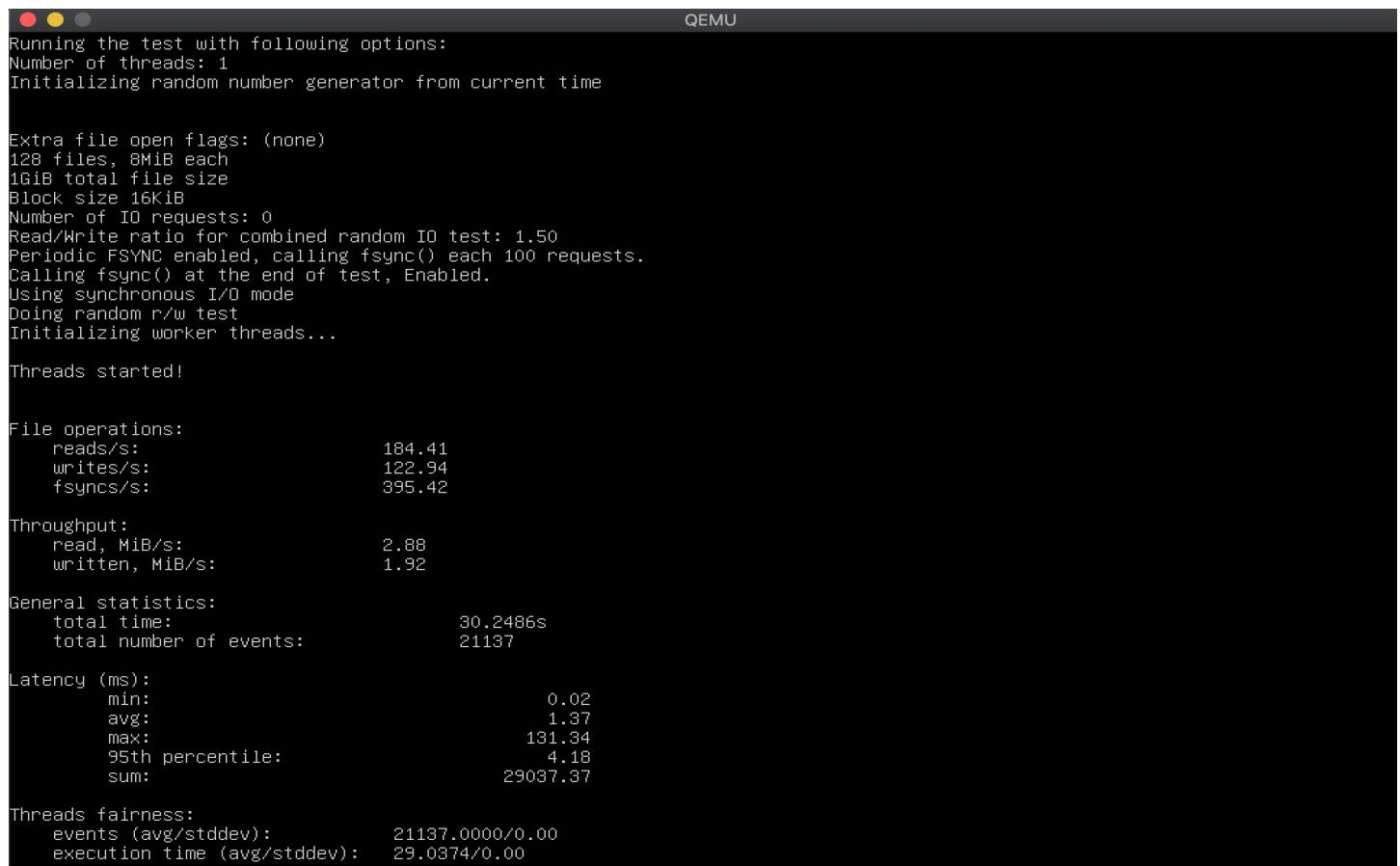
At the prepare stage Sysbench creates a specified no of files with a specified total size, then at run stage each thread performs I/O operations on this set of files.

**Use sync && sudo purge to cleanup the cache so that runs are always non-cached**

### **1a] Testing on QEMU:**

```
> sysbench --test=fileio --file-total-size=1G --max-time=30 --max-requests=0  
--file-test-mode=rndrw prepare  
> sysbench --test=fileio --file-total-size=1G --max-time=30 --max-requests=0  
--file-test-mode=rndrw run  
> sysbench --test=fileio --file-total-size=1G --max-time=30 --max-requests=0  
--file-test-mode=rndrw cleanup
```

### **Test Run #1**



```
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: 0  
Read/Write ratio for combined random IO test: 1.50  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing random r/w test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s: 184.41  
  writes/s: 122.94  
  fsync/s: 395.42  
  
Throughput:  
  read, MiB/s: 2.88  
  written, MiB/s: 1.92  
  
General statistics:  
  total time: 30.2486s  
  total number of events: 21137  
  
Latency (ms):  
  min: 0.02  
  avg: 1.37  
  max: 131.34  
  95th percentile: 4.18  
  sum: 29037.37  
  
Threads fairness:  
  events (avg/stddev): 21137.0000/0.00  
  execution time (avg/stddev): 29.0374/0.00
```

## Test Run #2

```
QEMU
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: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          181.64
  writes/s:         121.09
  fsyncs/s:        387.93

Throughput:
  read, MiB/s:      2.84
  written, MiB/s:   1.89

General statistics:
  total time:       30.3834s
  total number of events: 20861

Latency (ms):
  min:              0.02
  avg:             1.39
  max:            196.95
  95th percentile: 3.96
  sum:           28987.49

Threads fairness:
  events (avg/stddev): 20861.0000/0.00
  execution time (avg/stddev): 28.9875/0.00
```

## Test Run #3

```
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: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          262.14
  writes/s:         174.76
  fsyncs/s:        559.87

Throughput:
  read, MiB/s:      4.10
  written, MiB/s:   2.73

General statistics:
  total time:       30.2062s
  total number of events: 29987

Latency (ms):
  min:                0.02
  avg:                0.97
  max:              100.46
  95th percentile:    2.86
  sum:            29199.98

Threads fairness:
  events (avg/stddev): 29987.0000/0.00
  execution time (avg/stddev): 29.2000/0.00
```

## Test Run #4

```
QEMU
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: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          266.22
  writes/s:         177.48
  fsyncs/s:        567.97

Throughput:
  read, MiB/s:      4.16
  written, MiB/s:   2.77

General statistics:
  total time:       30.1961s
  total number of events: 30425

Latency (ms):
  min:                0.02
  avg:                0.96
  max:                62.04
  95th percentile:    2.91
  sum:            29117.63

Threads fairness:
  events (avg/stddev): 30425.0000/0.00
  execution time (avg/stddev): 29.1176/0.00
```

## Test Run #5

```
QEMU  
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: 0  
Read/Write ratio for combined random IO test: 1.50  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing random r/w test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s:          256.42  
  writes/s:         170.95  
  fsyncs/s:        549.06  
  
Throughput:  
  read, MiB/s:      4.01  
  written, MiB/s:   2.67  
  
General statistics:  
  total time:       30.1789s  
  total number of events: 29345  
  
Latency (ms):  
  min:              0.02  
  avg:             1.00  
  max:            145.71  
  95th percentile:  3.13  
  sum:           29325.43  
  
Threads fairness:  
  events (avg/stddev): 29345.0000/0.00  
  execution time (avg/stddev): 29.3254/0.00
```

## 1b] Testing on Docker:

```
> sysbench --test=fileio --file-total-size=1G --max-time=30 --max-requests=0  
--file-test-mode=rndrw prepare  
> sysbench --test=fileio --file-total-size=1G --max-time=30 --max-requests=0  
--file-test-mode=rndrw run  
> sysbench --test=fileio --file-total-size=1G --max-time=30 --max-requests=0  
--file-test-mode=rndrw cleanup
```

## Test Run #1

```
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: 0  
Read/Write ratio for combined random IO test: 1.50  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing random r/w test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s:          1461.98  
  writes/s:         974.66  
  fsyncs/s:        3121.76  
  
Throughput:  
  read, MiB/s:      22.84  
  written, MiB/s:   15.23  
  
General statistics:  
  total time:           30.0311s  
  total number of events: 166854  
  
Latency (ms):  
  min:                  0.00  
  avg:                  0.18  
  max:                 292.22  
  95th percentile:     0.48  
  sum:                29794.73  
  
Threads fairness:  
  events (avg/stddev): 166854.0000/0.00  
  execution time (avg/stddev): 29.7947/0.00  
  
/ # █
```

## Test Run #2

```
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: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
 reads/s:          1412.27
 writes/s:         941.51
 fsyncs/s:        3013.60

Throughput:
 read, MiB/s:      22.07
 written, MiB/s:   14.71

General statistics:
 total time:       30.0777s
 total number of events: 161319

Latency (ms):
 min:              0.00
 avg:              0.19
 max:             223.00
 95th percentile:  0.53
 sum:            29873.98

Threads fairness:
 events (avg/stddev): 161319.0000/0.00
 execution time (avg/stddev): 29.8740/0.00

/ # █
```

## Test Run #3

```
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: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          1419.56
  writes/s:         946.40
  fsyncs/s:        3028.68

Throughput:
  read, MiB/s:      22.18
  written, MiB/s:   14.79

General statistics:
  total time:       30.1738s
  total number of events: 162658

Latency (ms):
  min:              0.00
  avg:              0.18
  max:             827.68
  95th percentile: 0.50
  sum:            29827.37

Threads fairness:
  events (avg/stddev):    162658.0000/0.00
  execution time (avg/stddev): 29.8274/0.00

/ # █
```

## Test Run #4

```
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: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          1390.08
  writes/s:         926.72
  fsyncs/s:        2967.58

Throughput:
  read, MiB/s:      21.72
  written, MiB/s:   14.48

General statistics:
  total time:       30.2555s
  total number of events: 159763

Latency (ms):
  min:              0.00
  avg:              0.19
  max:             438.67
  95th percentile:  0.53
  sum:            30069.01

Threads fairness:
  events (avg/stddev):    159763.0000/0.00
  execution time (avg/stddev): 30.0690/0.00
```

## Test Run #5

```
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: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          1524.75
  writes/s:         1016.50
  fsyncs/s:        3255.97

Throughput:
  read, MiB/s:      23.82
  written, MiB/s:   15.88

General statistics:
  total time:       30.0229s
  total number of events: 173931

Latency (ms):
  min:              0.00
  avg:              0.17
  max:             139.19
  95th percentile:  0.50
  sum:            29796.70

Threads fairness:
  events (avg/stddev): 173931.0000/0.00
  execution time (avg/stddev): 29.7967/0.00

/ #
```

## 2a] Testing on QEMU:

```
> sysbench --test=fileio --file-total-size=2G --max-time=30 --max-requests=0  
--file-test-mode=rndrw prepare  
> sysbench --test=fileio --file-total-size=2G --max-time=30 --max-requests=0  
--file-test-mode=rndrw run  
> sysbench --test=fileio --file-total-size=2G --max-time=30 --max-requests=0  
--file-test-mode=rndrw cleanup
```

### Test Run #1

```
QEMU  
Running the test with following options:  
Number of threads: 1  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 16MiB each  
2GiB total file size  
Block size 16KiB  
Number of IO requests: 0  
Read/Write ratio for combined random IO test: 1.50  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing random r/w test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s:          101.26  
  writes/s:         67.53  
  fsyncs/s:        219.68  
  
Throughput:  
  read, MiB/s:      1.58  
  written, MiB/s:   1.06  
  
General statistics:  
  total time:       30.1793s  
  total number of events: 11642  
  
Latency (ms):  
  min:                0.02  
  avg:                2.50  
  max:               217.66  
  95th percentile:    8.13  
  sum:              29152.44  
  
Threads fairness:  
  events (avg/stddev): 11642.0000/0.00  
  execution time (avg/stddev): 29.1524/0.00
```

## Test Run #2

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

Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          92.54
  writes/s:         61.69
  fsyncs/s:        197.75

Throughput:
  read, MiB/s:      1.45
  written, MiB/s:   0.96

General statistics:
  total time:       30.4452s
  total number of events: 10598

Latency (ms):
  min:              0.02
  avg:              2.77
  max:             147.93
  95th percentile:  9.73
  sum:            29381.62

Threads fairness:
  events (avg/stddev): 10598.0000/0.00
  execution time (avg/stddev): 29.3816/0.00
```

## Test Run #3

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

Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          102.42
  writes/s:         68.26
  fsyncs/s:        218.64

Throughput:
  read, MiB/s:      1.60
  written, MiB/s:   1.07

General statistics:
  total time:       30.4264s
  total number of events: 11724

Latency (ms):
  min:              0.02
  avg:              2.47
  max:             167.27
  95th percentile:  8.28
  sum:            28970.89

Threads fairness:
  events (avg/stddev): 11724.0000/0.00
  execution time (avg/stddev): 28.9709/0.00
```

## Test Run #4

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

Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          73.99
  writes/s:         49.33
  fsyncs/s:        159.95

Throughput:
  read, MiB/s:      1.16
  written, MiB/s:   0.77

General statistics:
  total time:       30.3698s
  total number of events: 8486

Latency (ms):
  min:                 0.02
  avg:                 3.40
  max:                358.80
  95th percentile:    10.27
  sum:               28856.54

Threads fairness:
  events (avg/stddev): 8486.0000/0.00
  execution time (avg/stddev): 28.8565/0.00
```

## Test Run #5

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

Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          94.68
  writes/s:         63.13
  fsyncs/s:        202.12

Throughput:
  read, MiB/s:      1.48
  written, MiB/s:   0.99

General statistics:
  total time:       30.3544s
  total number of events: 10813

Latency (ms):
  min:              0.02
  avg:              2.71
  max:             297.38
  95th percentile:  8.74
  sum:            29255.65

Threads fairness:
  events (avg/stddev): 10813.0000/0.00
  execution time (avg/stddev): 29.2556/0.00
```

## 2b] Testing on Docker:

```
> sysbench --test=fileio --file-total-size=2G --max-time=30 --max-requests=0  
--file-test-mode=rndrw prepare  
> sysbench --test=fileio --file-total-size=2G --max-time=30 --max-requests=0  
--file-test-mode=rndrw run  
> sysbench --test=fileio --file-total-size=2G --max-time=30 --max-requests=0  
--file-test-mode=rndrw cleanup
```

## Test Run #1

```
Running the test with following options:  
Number of threads: 1  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 16MiB each  
2GiB total file size  
Block size 16KiB  
Number of IO requests: 0  
Read/Write ratio for combined random IO test: 1.50  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing random r/w test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s:          1476.60  
  writes/s:         984.40  
  fsyncs/s:        3150.98  
  
Throughput:  
  read, MiB/s:      23.07  
  written, MiB/s:   15.38  
  
General statistics:  
  total time:           30.0266s  
  total number of events: 168391  
  
Latency (ms):  
  min:                  0.00  
  avg:                  0.18  
  max:                 1005.02  
  95th percentile:      0.50  
  sum:                29804.49  
  
Threads fairness:  
  events (avg/stddev): 168391.0000/0.00  
  execution time (avg/stddev): 29.8045/0.00  
  
/ #
```

## Test Run #2

```
Desktop — com.docker.cli • docker run -it cmk — 128x45
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          1384.86
  writes/s:         923.24
  fsyncs/s:        2955.34

Throughput:
  read, MiB/s:      21.64
  written, MiB/s:   14.43

General statistics:
  total time:       30.0224s
  total number of events: 157905

Latency (ms):
  min:              0.00
  avg:              0.19
  max:             1003.62
  95th percentile:  0.50
  sum:            29826.05

Threads fairness:
  events (avg/stddev):    157905.0000/0.00
  execution time (avg/stddev): 29.8261/0.00

/ #
```

## Test Run #3

```
 Desktop — com.docker.cli ▾ docker run -it cmk — 128x45
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          1410.64
  writes/s:         940.43
  fsyncs/s:        3010.86

Throughput:
  read, MiB/s:      22.04
  written, MiB/s:   14.69

General statistics:
  total time:       30.0265s
  total number of events: 160885

Latency (ms):
  min:              0.00
  avg:              0.19
  max:             1007.35
  95th percentile:  0.50
  sum:            29825.60

Threads fairness:
  events (avg/stddev):    160885.0000/0.00
  execution time (avg/stddev): 29.8256/0.00
/ #
```

## Test Run #4

```
Desktop — com.docker.cli • docker run -it cmk — 128x45
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          1340.30
  writes/s:         893.53
  fsyncs/s:        2861.33

Throughput:
  read, MiB/s:      20.94
  written, MiB/s:   13.96

General statistics:
  total time:       30.0361s
  total number of events: 152921

Latency (ms):
  min:              0.00
  avg:              0.19
  max:             117.66
  95th percentile:  0.55
  sum:            29794.11

Threads fairness:
  events (avg/stddev):    152921.0000/0.00
  execution time (avg/stddev): 29.7941/0.00
/ #
```

## Test Run #5

```
Desktop — com.docker.cli ▾ docker run -it cmk — 128x45
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          1366.43
  writes/s:         910.95
  fsyncs/s:        2916.49

Throughput:
  read, MiB/s:      21.35
  written, MiB/s:   14.23

General statistics:
  total time:       30.0324s
  total number of events: 155867

Latency (ms):
  min:              0.00
  avg:              0.19
  max:             1006.09
  95th percentile:  0.52
  sum:            29842.55

Threads fairness:
  events (avg/stddev):    155867.0000/0.00
  execution time (avg/stddev): 29.8426/0.00
/ #
```

### 3a] Testing on QEMU:

```
> sysbench --test=fileio --file-total-size=3G --max-time=30 --max-requests=0  
--file-test-mode=rndrw prepare  
> sysbench --test=fileio --file-total-size=3G --max-time=30 --max-requests=0  
--file-test-mode=rndrw run  
> sysbench --test=fileio --file-total-size=3G --max-time=30 --max-requests=0  
--file-test-mode=rndrw cleanup
```

### Test Run #1

```
QEMU  
Running the test with following options:  
Number of threads: 1  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 24MiB each  
3GiB total file size  
Block size 16KiB  
Number of IO requests: 0  
Read/Write ratio for combined random IO test: 1.50  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing random r/w test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s:          85.20  
  writes/s:         56.80  
  fsyncs/s:        182.12  
  
Throughput:  
  read, MiB/s:      1.33  
  written, MiB/s:   0.89  
  
General statistics:  
  total time:       30.2798s  
  total number of events: 9687  
  
Latency (ms):  
  min:              0.02  
  avg:              2.99  
  max:             162.41  
  95th percentile:  9.73  
  sum:            28955.92  
  
Threads fairness:  
  events (avg/stddev): 9687.0000/0.00  
  execution time (avg/stddev): 28.9559/0.00
```

## Test Run #2

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

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

Threads started!

File operations:
  reads/s:          83.70
  writes/s:         55.78
  fsyncs/s:        181.23

Throughput:
  read, MiB/s:      1.31
  written, MiB/s:   0.87

General statistics:
  total time:       30.2548s
  total number of events: 9612

Latency (ms):
  min:              0.02
  avg:              3.03
  max:             274.99
  95th percentile: 10.46
  sum:            29100.14

Threads fairness:
  events (avg/stddev): 9612.0000/0.00
  execution time (avg/stddev): 29.1001/0.00
```

## Test Run #3

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

Extra file open flags: (none)
128 files, 24MiB each
8GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

file operations:
  reads/s:          89.05
  writes/s:         59.37
  fsyncs/s:        192.28

Throughput:
  read, MiB/s:      1.39
  written, MiB/s:   0.93

General statistics:
  total time:       30.1661s
  total number of events: 10200

Latency (ms):
  min:                0.02
  avg:                2.86
  max:              208.53
  95th percentile:    9.39
  sum:            29188.73

Threads fairness:
  events (avg/stddev): 10200.0000/0.00
  execution time (avg/stddev): 29.1887/0.00
```

## Test Run #4

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

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

Threads started!

File operations:
  reads/s:          82.63
  writes/s:         55.09
  fsyncs/s:        176.77

Throughput:
  read, MiB/s:      1.29
  written, MiB/s:   0.86

General statistics:
  total time:       30.4925s
  total number of events: 9463

Latency (ms):
  min:                0.03
  avg:                3.10
  max:               396.77
  95th percentile:    9.39
  sum:              29327.95

Threads fairness:
  events (avg/stddev): 9463.0000/0.00
  execution time (avg/stddev): 29.3280/0.00
```

## Test Run #5

```
QEMU  
Running the test with following options:  
Number of threads: 1  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 24MiB each  
3GiB total file size  
Block size 16KiB  
Number of IO requests: 0  
Read/Write ratio for combined random IO test: 1.50  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing random r/w test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s:          89.29  
  writes/s:         59.53  
  fsyncs/s:        194.53  
  
Throughput:  
  read, MiB/s:      1.40  
  written, MiB/s:   0.93  
  
General statistics:  
  total time:       30.1798s  
  total number of events: 10254  
  
Latency (ms):  
  min:                 0.02  
  avg:                 2.86  
  max:                150.93  
  95th percentile:    8.74  
  sum:               29354.72  
  
Threads fairness:  
  events (avg/stddev): 10254.0000/0.00  
  execution time (avg/stddev): 29.3547/0.00
```

### 3b] Testing on Docker:

```
> sysbench --test=fileio --file-total-size=3G --max-time=30 --max-requests=0  
--file-test-mode=rndrw prepare  
> sysbench --test=fileio --file-total-size=3G --max-time=30 --max-requests=0  
--file-test-mode=rndrw run  
> sysbench --test=fileio --file-total-size=3G --max-time=30 --max-requests=0  
--file-test-mode=rndrw cleanup
```

### Test Run #1

```
Running the test with following options:  
Number of threads: 1  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 24MiB each  
3GiB total file size  
Block size 16KiB  
Number of IO requests: 0  
Read/Write ratio for combined random IO test: 1.50  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing random r/w test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s:          1425.41  
  writes/s:         950.28  
  fsyncs/s:        3041.91  
  
Throughput:  
  read, MiB/s:      22.27  
  written, MiB/s:   14.85  
  
General statistics:  
  total time:           30.0526s  
  total number of events: 162695  
  
Latency (ms):  
  min:                  0.00  
  avg:                  0.18  
  max:                 1004.02  
  95th percentile:      0.50  
  sum:                 29835.28  
  
Threads fairness:  
  events (avg/stddev): 162695.0000/0.00  
  execution time (avg/stddev): 29.8353/0.00
```

## Test Run #2

```
Desktop — com.docker.cli ▾ docker run -it cmk — 128x45
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 24MiB each
36GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          1496.34
  writes/s:         997.56
  fsyncs/s:        3193.78

Throughput:
  read, MiB/s:      23.38
  written, MiB/s:   15.59

General statistics:
  total time:       30.0315s
  total number of events: 170692

Latency (ms):
  min:              0.00
  avg:              0.17
  max:             229.61
  95th percentile:  0.50
  sum:            29764.34

Threads fairness:
  events (avg/stddev): 170692.0000/0.00
  execution time (avg/stddev): 29.7643/0.00

/ #
```

## Test Run #3

```
Desktop — com.docker.cli - docker run -it cmk — 128x45
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

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

Threads started!

File operations:
  reads/s:          1436.89
  writes/s:         957.92
  fsyncs/s:        3069.06

Throughput:
  read, MiB/s:      22.45
  written, MiB/s:   14.97

General statistics:
  total time:       30.0214s
  total number of events: 163915

Latency (ms):
  min:              0.00
  avg:              0.18
  max:             719.46
  95th percentile:  0.51
  sum:            29835.31

Threads fairness:
  events (avg/stddev): 163915.0000/0.00
  execution time (avg/stddev): 29.8353/0.00

/ #
```

## Test Run #4

```
Desktop — com.docker.cli ▾ docker run -it cmk — 128x45
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

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

Threads started!

File operations:
  reads/s:          1490.86
  writes/s:         993.91
  fsyncs/s:        3184.03

Throughput:
  read, MiB/s:      23.29
  written, MiB/s:   15.53

General statistics:
  total time:       30.0212s
  total number of events: 170066

Latency (ms):
  min:              0.00
  avg:              0.18
  max:             1007.76
  95th percentile:  0.50
  sum:            29831.57

Threads fairness:
  events (avg/stddev):    170066.0000/0.00
  execution time (avg/stddev): 29.8316/0.00

/ #
```

## Test Run #5

```
Desktop — com.docker.cli -> docker run -it cmk — 128x45
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

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

Threads started!

File operations:
  reads/s:          1520.99
  writes/s:         1014.00
  fsyncs/s:        3247.08

Throughput:
  read, MiB/s:      23.77
  written, MiB/s:   15.84

General statistics:
  total time:       30.0570s
  total number of events: 173677

Latency (ms):
  min:              0.00
  avg:              0.17
  max:             348.50
  95th percentile:  0.49
  sum:            29825.15

Threads fairness:
  events (avg/stddev):    173677.0000/0.00
  execution time (avg/stddev): 29.8251/0.00
/ #
```

## 4.3 System Virtualization (QEMU) v/s OS Virtualization (Docker)

### Experiment #1 - CPU

#### 1a] Testing on QEMU:

```
> $ sysbench --test=cpu --cpu-max-prime=10000 run
```

Attributes	Total Time (in seconds)	CPU Speed (Events per second)	Average Latency (in ms)
Minimum	10.0025	221.57	4.22
Maximum	10.0048	235.29	4.48
Average	10.00368	229.308	4.3344
Standard Deviation	0.00088634079224638	5.0545994895738	0.094601479903858

#### 1b] Testing on Docker:

Attributes	Total Time (in seconds)	CPU Speed (Events per second)	Average Latency (in ms)
Minimum	10.0011	299.60	2.64
Maximum	10.0031	378.32	3.33
Average	10.0017	354.556	2.834
Standard Deviation	0.00072938330115237	28.276519304893	0.25255494451703

## 2a] Testing on QEMU:

```
> $ sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
```

Attributes	Total Time (in seconds)	CPU Speed (Events per second)	Average Latency (in ms)
<b>Minimum</b>	30.0031	89.30	10.93
<b>Maximum</b>	30.0131	91.18	11.17
<b>Average</b>	30.00832	90.516	11.016
<b>Standard Deviation</b>	0.0036815214246292	0.72339753939311	0.091126285999156

## 2b] Testing on Docker:

Attributes	Total Time (in seconds)	CPU Speed (Events per second)	Average Latency (in ms)
<b>Minimum</b>	30.0007	144.56	6.61
<b>Maximum</b>	30.006	151.07	6.91
<b>Average</b>	30.003	147.562	6.77
<b>Standard Deviation</b>	0.002186321109073	2.9049709120747	0.13431306712305

### **3a] Testing on QEMU:**

```
> $ sysbench --test=cpu --cpu-max-prime=30000 --max-time=30 run
```

<b>Attributes</b>	<b>Total Time (in seconds)</b>	<b>CPU Speed (Events per second)</b>	<b>Average Latency (in ms)</b>
<b>Minimum</b>	30.0024	50.48	18.90
<b>Maximum</b>	30.0207	52.78	19.76
<b>Average</b>	30.00948	51.882	19.234
<b>Standard Deviation</b>	0.0064372043621435	0.9114252574951	0.34115099296353

### **3b] Testing on Docker:**

<b>Attributes</b>	<b>Total Time (in seconds)</b>	<b>CPU Speed (Events per second)</b>	<b>Average Latency (in ms)</b>
<b>Minimum</b>	30.0048	60.02	11.37
<b>Maximum</b>	30.0208	87.90	16.65
<b>Average</b>	30.01192	75.552	13.52
<b>Standard Deviation</b>	0.0056862641514447	10.856956111176	2.0439373767315

## Experiment #2 - FILE I/O

### 1a] Testing on QEMU:

```
> sysbench --test=fileio --file-total-size=1G --max-time=30 --max-requests=0 --file-test-mode=rndrw prepare  
> sysbench --test=fileio --file-total-size=1G --max-time=30 --max-requests=0 --file-test-mode=rndrw run  
> sysbench --test=fileio --file-total-size=1G --max-time=30 --max-requests=0 --file-test-mode=rndrw cleanup
```

Attributes	Total Time (in seconds)	Read Throughput (MiB/s)	Write Throughput (MiB/s)	Average Latency (in ms)
Minimum	30.1789	2.84	1.89	0.96
Maximum	30.3834	4.16	2.77	1.39
Average	30.24264	3.598	2.396	1.138
Standard Deviation	0.074034494662962	0.60459573270079	0.40227353877679	0.19813126961689

### 1b] Testing on Docker:

Attributes	Total Time (in seconds)	Read Throughput (MiB/s)	Write Throughput (MiB/s)	Average Latency (in ms)
Minimum	30.0229	21.72	14.48	0.17
Maximum	30.2555	23.82	15.88	0.19
Average	30.1122	22.526	15.018	0.182
Standard Deviation	0.089522511135469	0.7416629962456	0.49482926348388	0.0074833147735479

## **2a] Testing on QEMU:**

```
> sysbench --test=fileio --file-total-size=2G --max-time=30 --max-requests=0 --file-test-mode=rndrw prepare  
> sysbench --test=fileio --file-total-size=2G --max-time=30 --max-requests=0 --file-test-mode=rndrw run  
> sysbench --test=fileio --file-total-size=2G --max-time=30 --max-requests=0 --file-test-mode=rndrw cleanup
```

<b>Attributes</b>	<b>Total Time (in seconds)</b>	<b>Read Throughput (MiB/s)</b>	<b>Write Throughput (MiB/s)</b>	<b>Average Latency (in ms)</b>
<b>Minimum</b>	30.1793	1.16	0.77	2.47
<b>Maximum</b>	30.4452	1.60	1.07	3.40
<b>Average</b>	30.35492	1.438	0.97	2.77
<b>Standard Deviation</b>	0.094137461193724	0.15829087149928	0.10825894882179	0.33567841753678

## **2b] Testing on Docker:**

<b>Attributes</b>	<b>Total Time (in seconds)</b>	<b>Read Throughput (MiB/s)</b>	<b>Write Throughput (MiB/s)</b>	<b>Average Latency (in ms)</b>
<b>Minimum</b>	30.0224	20.94	13.96	0.18
<b>Maximum</b>	30.0361	23.07	15.38	0.19
<b>Average</b>	30.0288	21.808	14.538	0.188
<b>Standard Deviation</b>	0.00484437818507	0.72634427099	0.48428916981489	0.004

### **3a] Testing on QEMU:**

```
> sysbench --test=fileio --file-total-size=3G --max-time=30 --max-requests=0 --file-test-mode=rndrw prepare  
> sysbench --test=fileio --file-total-size=3G --max-time=30 --max-requests=0 --file-test-mode=rndrw run  
> sysbench --test=fileio --file-total-size=3G --max-time=30 --max-requests=0 --file-test-mode=rndrw cleanup
```

<b>Attributes</b>	<b>Total Time (in seconds)</b>	<b>Read Throughput (MiB/s)</b>	<b>Write Throughput (MiB/s)</b>	<b>Average Latency (in ms)</b>
<b>Minimum</b>	30.1798	1.29	0.86	2.86
<b>Maximum</b>	30.4925	1.40	0.93	3.10
<b>Average</b>	30.2746	1.344	0.896	2.968
<b>Standard Deviation</b>	0.11718257549653	0.043634848458543	0.029393876913398	0.094952619763754

### **3b] Testing on Docker:**

<b>Attributes</b>	<b>Total Time (in seconds)</b>	<b>Read Throughput (MiB/s)</b>	<b>Write Throughput (MiB/s)</b>	<b>Average Latency (in ms)</b>
<b>Minimum</b>	30.0212	22.27	14.85	0.17
<b>Maximum</b>	30.0570	23.77	15.84	0.18
<b>Average</b>	30.03674	23.032	15.356	0.176
<b>Standard Deviation</b>	0.015272668398155	0.57474864071175	0.38061003665169	0.0048989794855663

## 5. Performance Data

We can record the performance data of host machines using the `top` command.

### 5.1 During CPU Test

CPU utilization in case of QEMU was around ~65% whereas in case of Docker it was ~58%

**QEMU :**

```
Processes: 488 total, 97 running, 1 stuck, 390 sleeping, 2150 threads                                         12:57:02
Load Avg: 86.16, 49.14, 45.02  CPU usage: 47.41% user, 6.10% sys, 46.47% idle
SharedLibs: 200M resident, 27M data, 23M linkedit.
MemRegions: 221276 total, 4069M resident, 115M private, 1005M shared.
PhysMem: 12G used (2385M wired), 4483M unused.
VM: 3401G vsize, 1989M framework vsize, 46350534(0) swapins, 47703704(0) swapouts.
Networks: packets: 39074047/41G in, 21436507/3848M out. Disks: 23461058/506G read, 17206211/567G written.

PID  COMMAND  %CPU TIME #TH #WQ #PORT MEM PURG CMPRS PGRP PPID STATE BOOSTS
1992 qemu-system- 64.5 06:10:29 9/1 2 223+ 3305M+ 2048K 1412M 1992 1988 running *0[6727]
92763 Google Chrom 62.7 07:44:45 18/2 1 276 230M 0B 194M 852 852 running *0[3]
83226 Atom Helper 59.2 02:58:53 11/2 1 61 88M+ 0B 43M 829 1 running *0[1]
0 kernel_task 6.5 05:17:09 171/4 0 0 198M+ 0B 0B 0 0 running 0[0]
96333 top 5.5 00:05:95 1/1 0 36+ 3856K 0B 0B 96333 96160 running *0[1]
236 WindowServer 2.6 05:28:28 10 5 1238 500M- 9864K 70M 236 1 sleeping *0[1]
178 hidd 1.6 34:28:59 5 3 293 3972K 0B 944K 178 1 sleeping *0[1]
852 Google Chrom 1.4 02:42:34 33/2 1 3183 396M 0B 169M 852 1 running *0[23829]
6611 Google Chrom 1.2 62:39:50 11/2 1 149 44M 0B 20M 852 852 running *0[3]
186 PrinterInsta 1.1 49:55:22 11 1 47 43M 0B 4868K 136 136 sleeping *0[1]
30187 Google Chrom 0.5 04:34:94 14/1 1 270 156M+ 0B 32M 852 852 running *0[6]
827 Terminal 0.5 05:38:00 11/2 4/2 349 64M 16M 12M 827 1 running *0[11028]
829 Atom 0.4 38:34:25 28/1 1 448 204M+ 0B 178M 829 1 running *0[5856]
90616 Google Chrom 0.2 06:27:15 13/1 1 1549 207M+ 0B 37M- 852 852 running *0[6]
1 launchd 0.2 20:28:91 3 2 3684+ 25M 0B 12M 1 0 sleeping 0[0]
6635 Google Chrom 0.2 13:48:69 13/1 1 142 473M 0B 336M 852 852 running *0[8]
11023 SophosScanDL 0.1 17:45:39 38 4 3328 348M 0B 224M 11023 1 stuck *0[1]
154 opendirector 0.1 03:29:88 10 9 1250+ 9740K+ 64K 4892K 154 1 sleeping *0[1]
13317 Atom Helper 0.1 04:41:02 20/1 1 214 129M 0B 123M 829 829 running *0[101]
```

**Docker:**

```
Processes: 489 total, 95 running, 394 sleeping, 2231 threads                                         13:21:48
Load Avg: 94.24, 89.14, 78.39  CPU usage: 33.40% user, 20.4% sys, 46.55% idle
SharedLibs: 125M resident, 28M data, 12M linkedit.
MemRegions: 185736 total, 4222M resident, 117M private, 1492M shared. PhysMem: 16G used (2318M wired), 35M unused.
VM: 3169G vsize, 1989M framework vsize, 46632043(0) swapins, 48196297(0) swapouts.
Networks: packets: 39099504/41G in, 21464160/3859M out. Disks: 23598006/510G read, 17384126/577G written.

PID  COMMAND  %CPU TIME #TH #WQ #PORT MEM PURG CMPRS PGRP PPID STATE BOOSTS
92763 Google Chrom 67.4 08:02:08 18/1 1 276 232M 0B 221M 852 852 running *0[3]
83226 Atom Helper 67.0 03:16:20 11/1 1 61 84M+ 0B 44M 829 1 running *0[1]
96531 com.docker.h 57.7 03:13:62 11/1 0 34 11G 0B 280M 96461 96527 running *0[1]
96378 top 11.9 01:12:83 1/1 0 63 4312K 0B 908K 96378 96160 running *0[1]
0 kernel_task 5.6 05:19:12 171/4 0 0 226M- 0B 0B 0 0 running 0[0]
178 hidd 1.4 34:52:77 6 3 291 3968K 0B 1816K 178 1 sleeping *0[1]
936 Adobe_CCXPro 1.4 16:12:74 18 2 84 32M 0B 19M- 936 1 sleeping *0[1]
827 Terminal 0.7 05:57:48 12/3 5/3 360- 84M+ 5412K- 27M 827 1 running *0[11250]
236 WindowServer 0.6 05:30:26 11 5 1225 513M 8296K+ 85M 236 1 sleeping *0[1]
852 Google Chrom 0.4 02:43:32 32 1 3126 369M 0B 303M 852 1 sleeping *0[24092]
7271 Slack Helper 0.2 25:39:39 17 2 403 190M+ 0B 157M- 868 868 sleeping *0[5]
6611 Google Chrom 0.1 62:49:43 10/1 1 129 43M 0B 30M 852 852 running *0[3]
134 logd 0.0 02:38:94 4 3 1462+ 24M+ 0B 29M- 134 1 sleeping *0[1]
1 launchd 0.0 20:35:59 3/1 2/1 3569+ 25M+ 0B 16M 1 0 running 0[0]
1017 AdobeCRDaemon 0.0 02:38:12 5 4 58 1940K 0B 1260K- 1017 1012 sleeping *0[1]
1026 AdobeCRDaemon 0.0 02:38:30 4 3 56 1964K 0B 1288K 1026 1021 sleeping *0[1]
123 fseventsds 0.0 06:21:22 11 1 249 9048K 0B 7512K 123 1 sleeping *0[1]
1009 AdobeCRDaemon 0.0 02:38:49 5 4 58 1980K 0B 1312K 1009 997 sleeping *0[1]
946 AdobeCRDaemon 0.0 02:38:56 5 4 58 1992K 0B 1296K 946 930 sleeping *0[1]
90616 Google Chrom 0.0 06:28:37 13 1 1546 209M 0B 183M 852 852 sleeping *0[6]
```

## 5.1 During FILE I/O Test

### QEMU

karthikmanjunath — top — 115x28														12:59:28
PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP	PPID	STATE	BOOSTS	
1992	qemu-system-	68.0	06:11:19	8/1	1	223	3337M+	36M+	332M-	1992	1988	running	*0[6748]	
92763	Google Chrom	65.3	07:46:32	18/1	1	276	230M	0B	194M	852	852	running	*0[3]	
83226	Atom Helper	57.7	03:00:40	11/1	1	61	88M+	0B	43M	829	1	running	*0[1]	
0	kernel_task	12.8	05:17:18	171/4	0	0	207M	0B	0B	0	0	running	0[0]	
96378	top	6.1	00:04:15	1/1	0	27	4220K	0B	0B	96378	96160	running	*0[1]	
236	WindowServer	3.3	05:28:37	10	5	1201	482M	12M	68M	236	1	sleeping	*0[1]	
178	hidd	1.8	34:30.43	5	3	279	3972K	0B	940K	178	1	sleeping	*0[1]	
852	Google Chrom	1.1	02:42:37	31/1	1	3178	396M	0B	169M	852	1	running	*0[23851]	
6611	Google Chrom	0.6	62:40.44	11/2	1	152	44M+	0B	20M	852	852	running	*0[3]	
827	Terminal	0.4	05:39.38	10/2	3/2	347	70M	16M	12M	827	1	running	*0[11052]	
30185	Google Chrom	0.3	01:37.80	14	1	228	104M	0B	27M	852	852	sleeping	*0[3]	
6635	Google Chrom	0.1	13:48.82	15	1	148	473M	0B	336M	852	852	sleeping	*0[8]	
6610	Google Chrom	0.0	02:31:39	13/1	1	376	734M	16M	123M	852	852	running	*0[7825]	
123	fseventsds	0.0	06:20.48	11	1	259	9048K	0B	5820K	123	1	sleeping	*0[1]	
181	AirPlayXPCHe	0.0	01:15.74	7/1	4/1	122	4448K+	0B	1484K	181	1	running	*0[1]	
1026	AdobeCRDaemo	0.0	02:37.67	4/2	3	56	1952K	0B	940K	1026	1021	running	*0[1]	
1017	AdobeCRDaemon	0.0	02:37.48	5/2	4	58	1940K	0B	932K	1017	1012	running	*0[1]	
1009	AdobeCRDaemon	0.0	02:37.86	5/2	4	58	1980K	0B	964K	1009	997	running	*0[1]	
7271	Slack Helper	0.0	25:36.95	17	2	415	182M	0B	59M	868	868	sleeping	*0[5]	

### Docker

karthikmanjunath — top — 115x28														13:20:08
PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP	PPID	STATE	BOOSTS	
96531	com.docker.h	61.5	02:51.57	11/2	0	34	11G+	0B	263M-	96461	96527	running	*0[1]	
92763	Google Chrom	45.3	08:01:07	18/1	1	276	232M	0B	221M+	852	852	running	*0[3]	
83226	Atom Helper	45.0	03:15:19	11/1	1	61	84M-	0B	44M	829	1	running	*0[1]	
1992	qemu-system-	44.2	06:12:53	8/1	1	223	3337M	14M	2908M-	1992	1988	running	*0[6840]	
236	WindowServer	22.9	05:30:15	11	5	1222	513M+	5252K+	86M-	236	1	sleeping	*0[1]	
0	kernel_task	15.3	05:19:02	171/5	0	0	269M	0B	0B	0	0	running	0[0]	
97911	screenCaptur	9.8	00:00:41	3/1	2	58	3024K+	580K-	0B	870	870	running	*0[1]	
96378	top	6.7	01:06.22	1/1	0	63	4308K	0B	1024K	96378	96160	running	*0[1]	
178	hidd	5.0	34:50.26	6	3	291	3980K	0B	1804K	178	1	sleeping	*0[1]	
527	SophosMcAfee	1.2	04:01.61	24	15	5757+	22M+	0B	10M-	527	1	sleeping	*0[1]	
97912	screenCaptur	1.1	00:00:18	7	5	162	4016K	0B	0B	97912	1	sleeping	*0[207+]	
827	Terminal	1.1	05:56.07	12	5/2	360-	79M+	9700K	35M-	827	1	sleeping	*0[11227]	
852	Google Chrom	0.7	02:43:26	32	2	3122	369M	0B	308M-	852	1	sleeping	*0[24079]	
372	SophosConfig	0.7	03:29.23	5	1	7605	26M	0B	16M-	372	1	sleeping	*0[1]	
237	mDNSResponde	0.6	03:24.70	4	1	72	4472K	0B	3216K	237	1	sleeping	*0[1]	
137	SophosService	0.5	04:26.99	4	1	6389-	53M-	0B	42M-	137	1	sleeping	*0[1]	
96602	mdworker_sha	0.3	00:00.31	4/1	1	50	5740K-	0B	3560K-	96602	1	running	*0[1]	
144	mds	0.3	17:53.90	9/2	6/2	937+	81M+	0B	75M-	144	1	running	*0[1]	
441	mds_stores	0.3	12:07.20	8	6/2	112	128M-	0B	121M-	441	1	sleeping	*0[1]	

Memory Utilization depends more on the type of virtualization that the tests performed

## **6. Conclusion:**

When experimenting with CPU Performance, since Dockers are light weighted than Virtual Machines clearly Docker containers out performed Virtual Machines. In terms of File I/O also, Docker clearly is performing better than Virtual Machine. The data from the Experiment #1 & #2 also conform to the same. So, Containers are always the best choice for Virtualization as compared to Virtual Machines. But in terms of security, Virtual Machines are always placed ahead of Containers.

## **7.Git Repository Information**

Please find my Git Repository URL which contains all the information.

<https://github.com/karthik-cm/COEN241-CloudComputing.git>

**Name:** Karthik Cheernalli Manjunath  
**Student ID:** 00001587860