

Krithik Naralasetty Homework 1 Report

This report is regarding my experimentation with System virtualization v/s OS virtualization. I worked on configuring the VM with Virtual Box and I simulated a Ubuntu Server (version 20.04). The container was built and ran using Docker, and the image was a pre-existing image by csmnpp which already had the sysbench installed.

A little background:

My laptop configurations are:

CPU: Intel i7 - 8th gen (4 cores and 8 threads)

RAM: 16GB (15.88 GB usable, 2400 MHz)

OS: Windows 10: Home Edition (x86 64-bit)

Memory: 512 GB SATA SSD (477 GB usable, 123GB free)

Graphics: 8GB (4 Intel(R) UHD + 4 NVidia 130MX)

1. Configuration of Virtual Box VM and options

The configuration of the VM created using the below commands would be:

Cores: 2 Cores

Memory: 1536 MB

VRAM: 128

Image Size: 20Gb

The shell script to configure the VM on Virtualbox is “vbox_config.sh”

To configure the VM from VBox, I chose to use the command line commands and options. This allowed for better control over the resources a machine was allocated. My base machine (VM) was created (and registered for easier use) with the help of the command

```
VBoxManage createvm --name ubuntu --ostype "Ubuntu-64" --register  
--basefolder ".\Krithik"
```

I enabled the I/O APIC so that I could access my SATA SSD for faster file access (I know it provides a little performance overhead, but the file read and write speeds

would be better). I also gave the machine 2 vCPU cores, 128 MB of VRAM, and 1536MB of RAM using the command below

```
VBoxManage modifyvm --cpus 2 --vram 128 --memory 1536
```

I wanted to also enable the network card so I could download sysbench inside the VM, so I also gave it a network card with my host machine network connection

```
VBoxManage modifyvm --nic1 nat
```

The next step I did was to create the disk image(Disk-space) for my Ubuntu server. The above was just the specification about the hardware resources of the machine. I allocated 20Gb to the server. (VDI is Virtual Disk Image, sort of like an ISO (unpacked))

```
VBoxManage createhd --filename ".\Krithik\ubuntu\ubuntu_DISK.vdi" --size 20000 --format VDI
```

After creating the disk image, we would then need to connect the ISO with Ubuntu live server, by using a SATA controller and an IDE controller (Integrated Drive Electronics) which will add the ISO as a CDROM for the VM.

```
VBoxManage storagectl ubuntu --name "SATA Controller" --port 0 --device 0 --type hdd --medium ".\Krithik\ubuntu\ubuntu_DISK.vdi"
```

```
VBoxManage storagectl ubuntu --name "SATA Controller" --add sata --controller IntelAhci
```

```
VBoxManage storageattach ubuntu --storagectl "SATA Controller" --port 0 --device 0 --type hdd --medium ".\Krithik\ubuntu\ubuntu_DISK.vdi"
```

```
VBoxManage storagectl ubuntu --name "IDE Controller" --add ide --controller PIIX4
```

PIIX4 is PCI IDE ISA () Xcelerator. It is a type of IDE Controller by Intel to communicate with disk drives connected to a machine. Then we can just attach these drives to the VM via the following command.

```
VBoxManage modifyvm ubuntu --boot1 dvd --boot2 disk --boot3 none --boot4 none
```

The next and final step is to run the VM using “VBoxManage startvm ubuntu”. We can power it off using the command “VBoxManage controlvm ubuntu poweroff”. I opted to not ssh into the VM. We could enable SSH using the below commands.

```
VBoxManage modifyvm ubuntu --vrde on
```

```
VBoxManage modifyvm ubuntu --vrde multicon on --vrdeport 10001
```

```
VBoxHeadless --startvm ubuntu
```

Or

```
VBoxManage startvm ubuntu --headless
```

2. Configuring the Container environment

The configurations of the Docker container can be found out using the .wslconfig file in userprofile. I created my config file with memory similar to my VM, i.e; the memory of 1536MB and 2 processing cores.

The main part of configuring a docker image is to download and install Docker Desktop for windows. Most of the container engine is set up for configuring the base resources limits.

Alternatively, we can configure the max resources we want to give to the containers using a “.wslconfig” file. It is editable and works to enable the WSL (Windows Subsystem for Linux) resource manager for Docker (Enable it is the setting for Docker). To do that, we would first have to close all the WSL based systems (which might include all the Linux dependent processes, like Chrome, OneNote, etc) using the command

```
wsl --shutdown
```

Then I opened/created a “.wslconfig” file using

```
notepad “$env:USERPROFILE\.wslconfig”
```

And added the below lines

```
[wsl2]
```

```
memory=1.5G
```

```
processors=2
```

I saved the file and restarted my Docker Desktop to finalize the changes.

The next part is to create/pull a container image. I decided to pull a container image of ubuntu, which already had installed the sysbench inside the server. The

docker image is “csminpp/ubuntu-sysbench” and I pulled it from the docker hub using the command.

```
docker pull csminpp/ubuntu-sysbench
```

Once I had the image downloaded and the resources set up, all I had to do was to run the Docker image.

```
docker run -it csminpp/ubunu-sysbench bash
```

Following are some other operations that I found are important/useful:

create — Create a container from an image.

start — Start an existing container.

run — Create a new container and start it.

ls — List running containers.

inspect — See lots of info about a container.

logs — Print logs.

stop — Gracefully stop the running container.

kill — Stop the main process in the container abruptly.

rm — Delete a stopped container.

When benchmarking the VM/Container, I used sysbench’s CPU and file I/O tests to measure the competency of the machine/container.

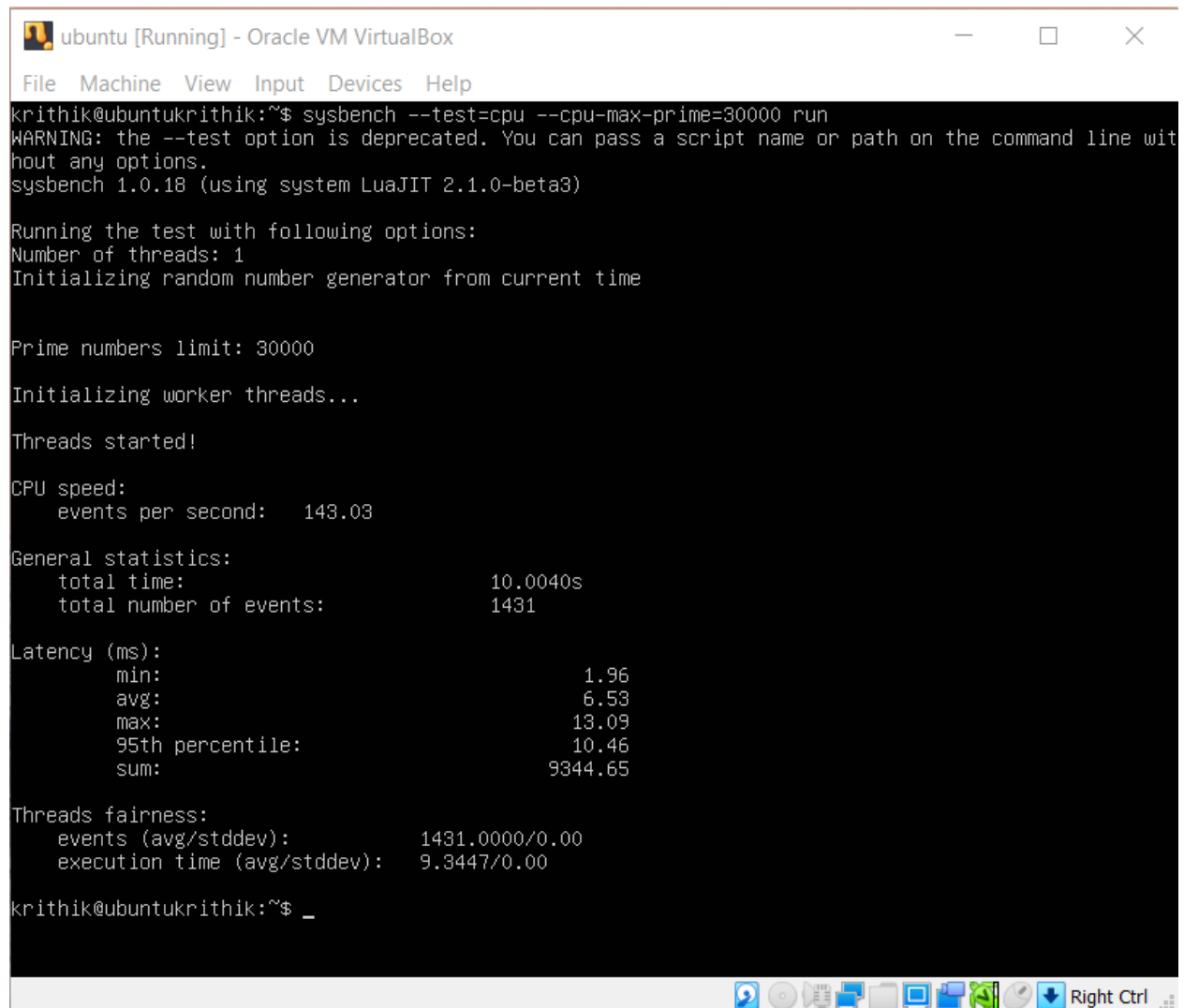
To average out the values across the board, I decided to run the tests 5 times (for every test I conducted).

3. Benchmarking System Virtualization using VirtualBox

The CPU benchmarking test was to run the CPU-Max-Prime test, which finds the max prime number for a given limit of n.

Below are the details of the benchmarking results.

Test 1 (CPU Max Prime)



```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
krithik@ubuntukrithik:~$ sysbench --test=cpu --cpu-max-prime=30000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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

Prime numbers limit: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:   143.03

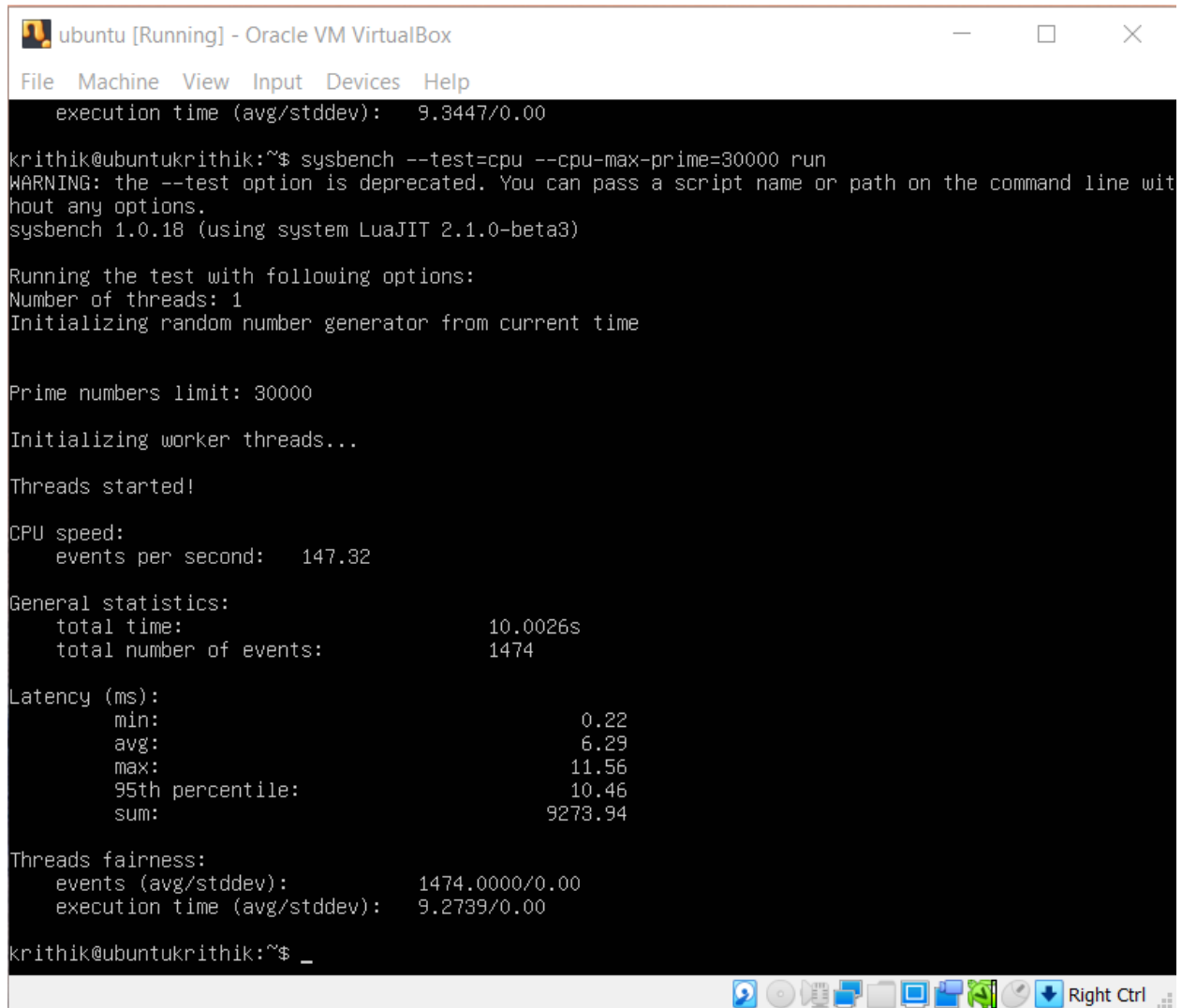
General statistics:
  total time:          10.0040s
  total number of events: 1431

Latency (ms):
  min:                 1.96
  avg:                 6.53
  max:                 13.09
  95th percentile:    10.46
  sum:                 9344.65

Threads fairness:
  events (avg/stddev): 1431.0000/0.00
  execution time (avg/stddev): 9.3447/0.00

krithik@ubuntukrithik:~$ _
```

Test 2 (CPU Max Prime)



The screenshot shows a terminal window titled "ubuntu [Running] - Oracle VM VirtualBox". The terminal output displays the execution of the sysbench CPU test with the following details:

```
execution time (avg/stddev): 9.3447/0.00

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

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

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

CPU speed:
  events per second: 147.32

General statistics:
  total time: 10.0026s
  total number of events: 1474

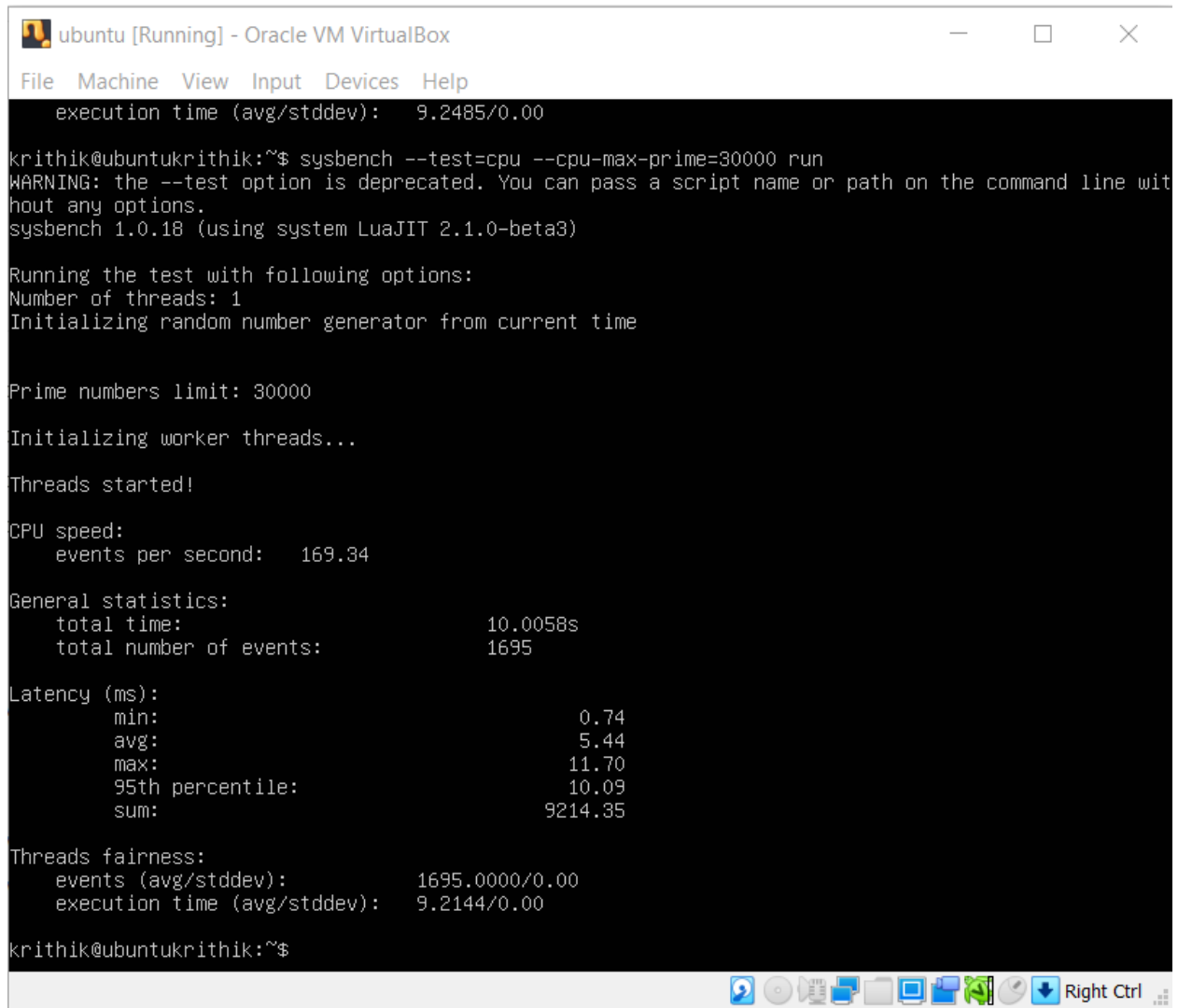
Latency (ms):
  min: 0.22
  avg: 6.29
  max: 11.56
  95th percentile: 10.46
  sum: 9273.94

Threads fairness:
  events (avg/stddev): 1474.0000/0.00
  execution time (avg/stddev): 9.2739/0.00

krithik@ubuntukrithik:~$ _
```

The terminal window includes a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The bottom status bar shows various icons and the text "Right Ctrl".

Test 3 (CPU Max Prime)



The screenshot shows a terminal window titled "ubuntu [Running] - Oracle VM VirtualBox". The terminal output displays the results of a sysbench CPU test. The test was run with the command `sysbench --test=cpu --cpu-max-prime=30000 run`. The output includes the execution time (9.2485/0.00), a warning about the deprecated `--test` option, and various performance metrics such as CPU speed (169.34 events per second), general statistics (total time: 10.0058s, total number of events: 1695), latency (min: 0.74, avg: 5.44, max: 11.70, 95th percentile: 10.09, sum: 9214.35), and threads fairness (events: 1695.0000/0.00, execution time: 9.2144/0.00).

```
File Machine View Input Devices Help
execution time (avg/stddev):  9.2485/0.00

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

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

Prime numbers limit: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:   169.34

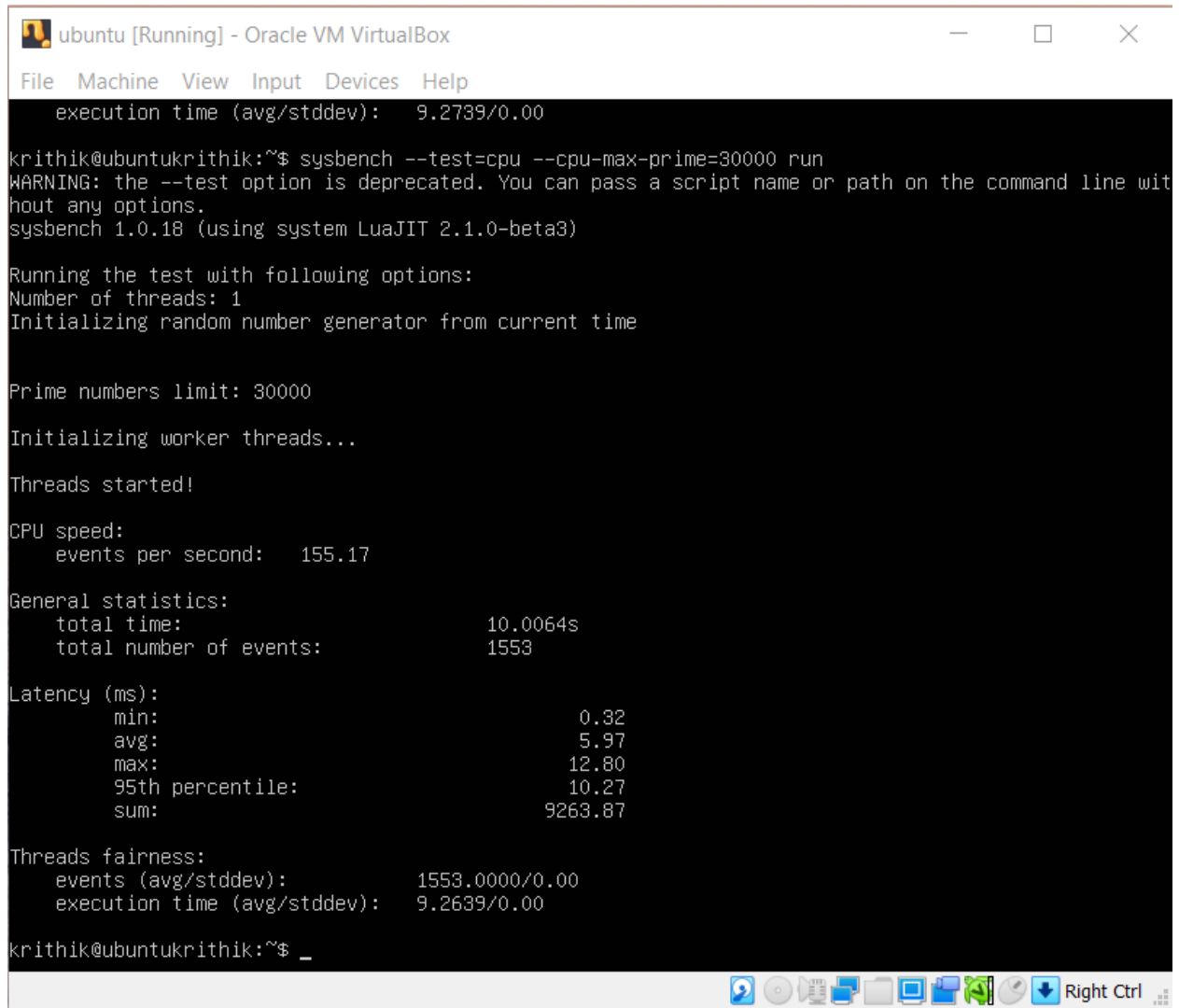
General statistics:
  total time:          10.0058s
  total number of events: 1695

Latency (ms):
  min:                 0.74
  avg:                 5.44
  max:                 11.70
  95th percentile:    10.09
  sum:                 9214.35

Threads fairness:
  events (avg/stddev): 1695.0000/0.00
  execution time (avg/stddev): 9.2144/0.00

krithik@ubuntukrithik:~$
```

Test 4 (CPU Max Prime)



The screenshot shows a terminal window titled "ubuntu [Running] - Oracle VM VirtualBox". The terminal output displays the results of a sysbench CPU test. The test was run with the command `sysbench --test=cpu --cpu-max-prime=30000 run`. The output includes the execution time (9.2739/0.00), a warning about the deprecated `--test` option, and various performance metrics such as CPU speed (155.17 events per second), general statistics (total time 10.0064s, total number of events 1553), latency (min 0.32, avg 5.97, max 12.80, 95th percentile 10.27, sum 9263.87), and threads fairness (events 1553.0000/0.00, execution time 9.2639/0.00). The terminal prompt is `krithik@ubuntukrithik:~$`.

```
execution time (avg/stddev):  9.2739/0.00

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

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

Prime numbers limit: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:   155.17

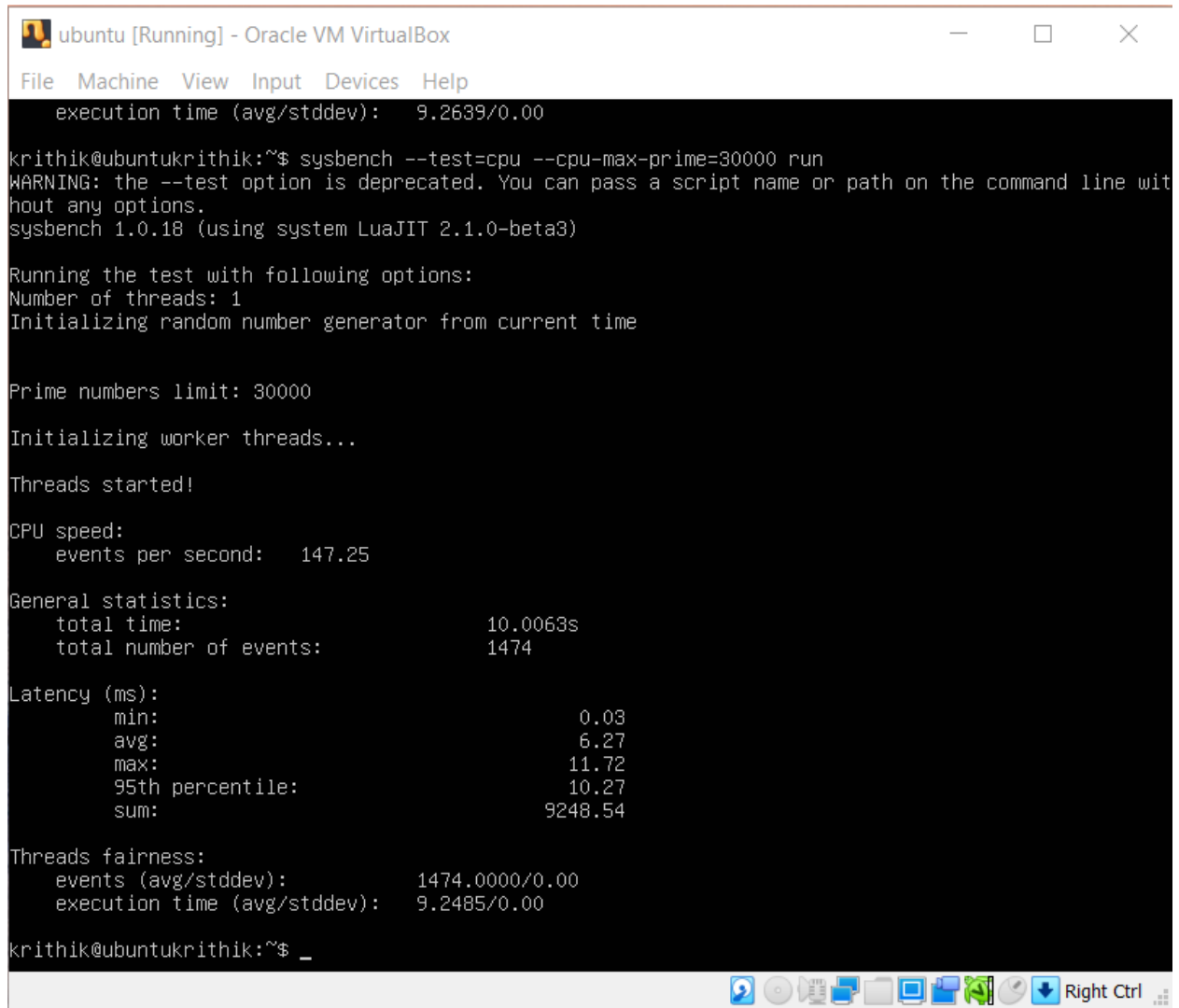
General statistics:
  total time:          10.0064s
  total number of events: 1553

Latency (ms):
  min:                 0.32
  avg:                 5.97
  max:                 12.80
  95th percentile:    10.27
  sum:                 9263.87

Threads fairness:
  events (avg/stddev): 1553.0000/0.00
  execution time (avg/stddev): 9.2639/0.00

krithik@ubuntukrithik:~$ _
```


Test 5 (CPU Max Prime)



The screenshot shows a terminal window titled "ubuntu [Running] - Oracle VM VirtualBox". The terminal output displays the execution of the sysbench CPU test with the following details:

```
execution time (avg/stddev): 9.2639/0.00

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

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

Prime numbers limit: 30000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 147.25

General statistics:
  total time: 10.0063s
  total number of events: 1474

Latency (ms):
  min: 0.03
  avg: 6.27
  max: 11.72
  95th percentile: 10.27
  sum: 9248.54

Threads fairness:
  events (avg/stddev): 1474.0000/0.00
  execution time (avg/stddev): 9.2485/0.00

krithik@ubuntukrithik:~$ _
```

The terminal window includes a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The bottom status bar shows various icons and the text "Right Ctrl".

Test 1 (File I/O test with Random read/write)

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Creating file test_file.94
Creating file test_file.95
Creating file test_file.96
Creating file test_file.97
Creating file test_file.98
Creating file test_file.99
Creating file test_file.100
Creating file test_file.101
Creating file test_file.102
Creating file test_file.103
Creating file test_file.104
Creating file test_file.105
Creating file test_file.106
Creating file test_file.107
Creating file test_file.108
Creating file test_file.109
Creating file test_file.110
Creating file test_file.111
Creating file test_file.112
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
8589934592 bytes written in 30.32 seconds (270.19 MiB/sec).
krithik@ubuntukrithik:~$ sysbench --file-total-size=8G --file-num=128
--file-test-mode=rndrw prepare_
```

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
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:                638.63
  writes/s:               424.79
  fsyncs/s:              1550.29

Throughput:
  read, MiB/s:            9.98
  written, MiB/s:         6.64

General statistics:
  total time:              10.4362s
  total number of events:  25234

Latency (ms):
  min:                     0.00
  avg:                     6.16
  max:                     59.41
  95th percentile:       22.69
  sum:                    155333.88

Threads fairness:
  events (avg/stddev):    1577.1250/46.06
  execution time (avg/stddev): 9.7084/0.04

krithik@ubuntukrithik:~$
```

Test 2 (File I/O test with Random read/write)

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Creating file test_file.93
Creating file test_file.94
Creating file test_file.95
Creating file test_file.96
Creating file test_file.97
Creating file test_file.98
Creating file test_file.99
Creating file test_file.100
Creating file test_file.101
Creating file test_file.102
Creating file test_file.103
Creating file test_file.104
Creating file test_file.105
Creating file test_file.106
Creating file test_file.107
Creating file test_file.108
Creating file test_file.109
Creating file test_file.110
Creating file test_file.111
Creating file test_file.112
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
8589934592 bytes written in 31.12 seconds (263.23 MiB/sec).
krithik@ubuntukrithik:~$ _
```

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
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:          664.72
  writes/s:         442.70
  fsyncs/s:         1606.51

Throughput:
  read, MiB/s:      10.39
  written, MiB/s:    6.92

General statistics:
  total time:       10.5138s
  total number of events: 26495

Latency (ms):
  min:              0.01
  avg:              5.85
  max:              73.49
  95th percentile: 21.11
  sum:              154903.78

Threads fairness:
  events (avg/stddev): 1655.9375/66.60
  execution time (avg/stddev): 9.6815/0.04

krithik@ubuntukrithik:~$
```

Test 3 (File I/O test with Random read/write)

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Creating file test_file.93
Creating file test_file.94
Creating file test_file.95
Creating file test_file.96
Creating file test_file.97
Creating file test_file.98
Creating file test_file.99
Creating file test_file.100
Creating file test_file.101
Creating file test_file.102
Creating file test_file.103
Creating file test_file.104
Creating file test_file.105
Creating file test_file.106
Creating file test_file.107
Creating file test_file.108
Creating file test_file.109
Creating file test_file.110
Creating file test_file.111
Creating file test_file.112
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
9663676416 bytes written in 24.82 seconds (371.31 MiB/sec).
krithik@ubuntukrithik:~$
```

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
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:          643.05
  writes/s:         427.90
  fsyncs/s:         1565.70

Throughput:
  read, MiB/s:      10.05
  written, MiB/s:    6.69

General statistics:
  total time:          10.4561s
  total number of events: 25526

Latency (ms):
  min:                0.00
  avg:                 6.05
  max:                69.88
  95th percentile:    21.50
  sum:                154489.03

Threads fairness:
  events (avg/stddev): 1595.3750/75.16
  execution time (avg/stddev): 9.6556/0.05

krithik@ubuntukrithik:~$
```

Test 4 (File I/O test with Random read/write)

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Creating file test_file.93
Creating file test_file.94
Creating file test_file.95
Creating file test_file.96
Creating file test_file.97
Creating file test_file.98
Creating file test_file.99
Creating file test_file.100
Creating file test_file.101
Creating file test_file.102
Creating file test_file.103
Creating file test_file.104
Creating file test_file.105
Creating file test_file.106
Creating file test_file.107
Creating file test_file.108
Creating file test_file.109
Creating file test_file.110
Creating file test_file.111
Creating file test_file.112
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
9663676416 bytes written in 33.42 seconds (275.77 MiB/sec).
krithik@ubuntukrithik:~$
```

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
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:          686.31
  writes/s:         456.78
  fsyncs/s:         1651.97

Throughput:
  read, MiB/s:      10.72
  written, MiB/s:    7.14

General statistics:
  total time:       10.4591s
  total number of events: 27189

Latency (ms):
  min:              0.00
  avg:              5.69
  max:              55.62
  95th percentile: 20.00
  sum:              154611.40

Threads fairness:
  events (avg/stddev): 1699.3125/52.35
  execution time (avg/stddev): 9.6632/0.04
krithik@ubuntukrithik:~$
```

Test 5 (File I/O test with Random read/write)

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Creating file test_file.93
Creating file test_file.94
Creating file test_file.95
Creating file test_file.96
Creating file test_file.97
Creating file test_file.98
Creating file test_file.99
Creating file test_file.100
Creating file test_file.101
Creating file test_file.102
Creating file test_file.103
Creating file test_file.104
Creating file test_file.105
Creating file test_file.106
Creating file test_file.107
Creating file test_file.108
Creating file test_file.109
Creating file test_file.110
Creating file test_file.111
Creating file test_file.112
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
9663676416 bytes written in 29.15 seconds (316.14 MiB/sec).
krithik@ubuntukrithik:~$ _
```

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
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:          673.48
  writes/s:         448.83
  fsyncs/s:        1622.40

Throughput:
  read, MiB/s:      10.52
  written, MiB/s:    7.01

General statistics:
  total time:       10.5128s
  total number of events: 26810

Latency (ms):
  min:              0.00
  avg:              5.78
  max:              60.83
  95th percentile: 20.74
  sum:              154842.11

Threads fairness:
  events (avg/stddev): 1675.6250/64.65
  execution time (avg/stddev): 9.6776/0.04
krithik@ubuntukrithik:~$ _
```

4. Benchmarking OS Virtualization using Docker Containers

Test 1 (CPU Max Prime)

 root@b86d9593594f: /

```
PS C:\Users\VENKAT\Desktop\Krithik> docker p
docker: 'p' is not a docker command.
See 'docker --help'
PS C:\Users\VENKAT\Desktop\Krithik> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
PS C:\Users\VENKAT\Desktop\Krithik> docker run -it csmnpp/ubuntu-sysbench
root@b86d9593594f:/# sysbench --test=cpu --cpu-max-prime=30000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

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

Doing CPU performance benchmark

Threads started!

Done.

Maximum prime number checked in CPU test: 30000

Test execution summary:
total time:                               58.2731s
total number of events:                   10000
total time taken by event execution: 58.2691
per-request statistics:
  min:                                   3.13ms
  avg:                                   5.83ms
  max:                                   12.73ms
  approx. 95 percentile:                 7.30ms

Threads fairness:
  events (avg/stddev):                   10000.0000/0.00
  execution time (avg/stddev):           58.2691/0.00

root@b86d9593594f:/#
root@b86d9593594f:/#
```

Test 2 (CPU Max Prime)

```
root@b86d9593594f: /
root@b86d9593594f:/# sysbench --test=cpu --cpu-max-prime=30000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

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

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 30000

Test execution summary:
total time: 46.9993s
total number of events: 10000
total time taken by event execution: 46.9968
per-request statistics:
  min: 3.13ms
  avg: 4.70ms
  max: 9.05ms
  approx. 95 percentile: 7.17ms

Threads fairness:
  events (avg/stddev): 10000.0000/0.00
  execution time (avg/stddev): 46.9968/0.00

root@b86d9593594f:/#
```


Test 3 (CPU Max Prime)

```
root@b86d9593594f: /  
root@b86d9593594f:/# sysbench --test=cpu --cpu-max-prime=30000 run  
sysbench 0.4.12: multi-threaded system evaluation benchmark  
  
Running the test with following options:  
Number of threads: 1  
  
Doing CPU performance benchmark  
  
Threads started!  
Done.  
  
Maximum prime number checked in CPU test: 30000  
  
Test execution summary:  
total time: 49.2084s  
total number of events: 10000  
total time taken by event execution: 49.2056  
per-request statistics:  
min: 3.13ms  
avg: 4.92ms  
max: 9.17ms  
approx. 95 percentile: 7.19ms  
  
Threads fairness:  
events (avg/stddev): 10000.0000/0.00  
execution time (avg/stddev): 49.2056/0.00  
root@b86d9593594f:/#
```

Test 4 (CPU Max Prime)

```
root@b86d9593594f: /
root@b86d9593594f:/# sysbench --test=cpu --cpu-max-prime=30000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

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

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 30000

Test execution summary:
total time: 46.3772s
total number of events: 10000
total time taken by event execution: 46.3747
per-request statistics:
  min: 3.13ms
  avg: 4.64ms
  max: 11.33ms
  approx. 95 percentile: 7.17ms

Threads fairness:
  events (avg/stddev): 10000.0000/0.00
  execution time (avg/stddev): 46.3747/0.00

root@b86d9593594f:/#
```

Test 5 (CPU Max Prime)

```
root@b86d9593594f: /
root@b86d9593594f:/# sysbench --test=cpu --cpu-max-prime=30000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

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

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 30000

Test execution summary:
total time: 49.2478s
total number of events: 10000
total time taken by event execution: 49.2450
per-request statistics:
  min: 3.13ms
  avg: 4.92ms
  max: 10.61ms
  approx. 95 percentile: 7.20ms

Threads fairness:
  events (avg/stddev): 10000.0000/0.00
  execution time (avg/stddev): 49.2450/0.00

root@b86d9593594f:/#
```

Test 1 (File I/O test with Random read/write)

```
root@b86d9593594f: /
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 72Mb each
9Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
Done.

Operations performed: 6037 Read, 4022 Write, 12800 Other = 22859 Total
Read 94.328Mb Written 62.844Mb Total transferred 157.17Mb (135.12Mb/sec)
8647.62 Requests/sec executed

Test execution summary:
total time: 1.1632s
total number of events: 10059
total time taken by event execution: 0.1110
per-request statistics:
  min: 0.00ms
  avg: 0.01ms
  max: 0.38ms
  approx. 95 percentile: 0.02ms

Threads fairness:
  events (avg/stddev): 628.6875/121.56
  execution time (avg/stddev): 0.0069/0.00

root@b86d9593594f:/#
```

Test 2 (File I/O test with Random read/write)

```
root@b86d9593594f: /
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 72Mb each
9Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
Done.

Operations performed: 6032 Read, 4024 Write, 12800 Other = 22856 Total
Read 94.25Mb Written 62.875Mb Total transferred 157.12Mb (135.09Mb/sec)
8645.88 Requests/sec executed

Test execution summary:
total time: 1.1631s
total number of events: 10056
total time taken by event execution: 0.1030
per-request statistics:
  min: 0.00ms
  avg: 0.01ms
  max: 0.22ms
  approx. 95 percentile: 0.02ms

Threads fairness:
  events (avg/stddev): 628.5000/112.17
  execution time (avg/stddev): 0.0064/0.00

root@b86d9593594f:/#
```

Test 3 (File I/O test with Random read/write)

```
root@b86d9593594f: /
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw cleanup
sysbench 0.4.12: multi-threaded system evaluation benchmark

Removing test files...
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw prepare
sysbench 0.4.12: multi-threaded system evaluation benchmark

128 files, 73728Kb each, 9216Mb total
Creating files for the test...
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 72Mb each
9Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
Done.

Operations performed: 6038 Read, 4026 Write, 12800 Other = 22864 Total
Read 94.344Mb Written 62.906Mb Total transferred 157.25Mb (131.88Mb/sec)
8440.11 Requests/sec executed

Test execution summary:
total time: 1.1924s
total number of events: 10064
total time taken by event execution: 0.1036
per-request statistics:
  min: 0.00ms
  avg: 0.01ms
  max: 0.18ms
  approx. 95 percentile: 0.02ms

Threads fairness:
  events (avg/stddev): 629.0000/140.58
  execution time (avg/stddev): 0.0065/0.00

root@b86d9593594f:/#
```

Test 4 (File I/O test with Random read/write)

```
root@b86d9593594f: /
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw cleanup
sysbench 0.4.12: multi-threaded system evaluation benchmark

Removing test files...
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw prepare
sysbench 0.4.12: multi-threaded system evaluation benchmark

128 files, 73728Kb each, 9216Mb total
Creating files for the test...
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 72Mb each
9Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
Done.

Operations performed: 6039 Read, 4029 Write, 12800 Other = 22868 Total
Read 94.359Mb Written 62.953Mb Total transferred 157.31Mb (133.29Mb/sec)
8530.60 Requests/sec executed

Test execution summary:
total time: 1.1802s
total number of events: 10068
total time taken by event execution: 0.1086
per-request statistics:
  min: 0.00ms
  avg: 0.01ms
  max: 0.17ms
  approx. 95 percentile: 0.02ms

Threads fairness:
  events (avg/stddev): 629.2500/137.05
  execution time (avg/stddev): 0.0068/0.00

root@b86d9593594f:/#
```

Test 5 (File I/O test with Random read/write)

```
root@b86d9593594f: /
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw cleanup
sysbench 0.4.12: multi-threaded system evaluation benchmark

Removing test files...
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw prepare
sysbench 0.4.12: multi-threaded system evaluation benchmark

128 files, 73728Kb each, 9216Mb total
Creating files for the test...
root@b86d9593594f:/# sysbench --num-threads=16 --test=fileio --file-total-size=9G --file-num=128
--file-test-mode=rndrw run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 72Mb each
9Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
Done.

Operations performed: 6030 Read, 4023 Write, 12800 Other = 22853 Total
Read 94.219Mb Written 62.859Mb Total transferred 157.08Mb (118.86Mb/sec)
7606.78 Requests/sec executed

Test execution summary:
total time: 1.3216s
total number of events: 10053
total time taken by event execution: 0.1201
per-request statistics:
  min: 0.00ms
  avg: 0.01ms
  max: 0.81ms
  approx. 95 percentile: 0.03ms

Threads fairness:
  events (avg/stddev): 628.3125/115.99
  execution time (avg/stddev): 0.0075/0.00

root@b86d9593594f:/#
```


5. Comparing the Results between them both

System Virtualization Results:

A. CPU Testing

The tests were designed to find a maximum prime number that is below a given threshold, which in this case is 30,000. The results varied a lot over the set of 5 tests. The table below gives a brief overview of the testing results.

Test number	Total number of events	Time taken for the test (seconds)	Events per second (Comparison Metric)	Difference from before
1	1431	10.0040	143.03	-
2	1474	10.0026	147.32	4.29 (+)
3	1695	10.0058	169.34	22.02 (+)
4	1553	10.0064	155.17	14.17 (-)
5	1474	10.0063	147.25	7.92

The maximum number of observed events per second is 169.34

The minimum number of observed events per second is 143.03

The average of the 5 tests was observed to be 152.42

B. File I/O Testing

The tests for file I/O were to prepare the set of files which will be accessed during the test. I decided to run a script that would create a set of 128 files combining to a total size of 9GB for the random read/write test. This was conducted across all 5 tests. (I cleaned up the files from the machine and cleared the cache too.) The below table gives a brief overview of the testing results.

Test number	Prepare Time (Seconds)	Prepare Speed (Mbit/s)	Run Read throughput	Run Write Throughput	Run Events per second
1	30.32	270.19	9.98	6.64	2417.93
2	31.12	263.23	10.39	6.92	2520.02
3	24.82	371.31	10.05	6.69	2441.25
4	33.42	275.77	10.72	7.14	2559.55
5	29.15	316.14	10.52	7.01	2550.22

The maximum number of observed events per second is 2559.55

The minimum number of observed events per second is 2417.93

The average of the 5 tests was observed to be 2497.79

OS Virtualization Results:

A. CPU Testing:

The tests were designed to find a maximum prime number that is below a given threshold, which in this case is 30,000. (The same number was kept to have the testing measures constant). The results varied a lot over the set of 5 tests. The table below gives a brief overview of the testing results.

Test number	Total number of events	Time taken for the test (seconds)	Events per second (Comparison Metric)
1	10000	58.2731	171.60
2	10000	46.9993	212.77

3	10000	49.2084	203.21
4	10000	46.3772	215.62
5	10000	49.2478	203.05

The maximum number of observed events per second is 215.62

The minimum number of observed events per second is 171.60

The average of the 5 tests was observed to be 201.25

B. File I/O Testing:

The tests for file I/O were to prepare the set of files which will be accessed during the test. I decided to run a script that would create a set of 128 files combining to a total size of 9GB for the random read/write test. This was conducted across all 5 tests. (I cleaned up the files from the machine and cleared the cache too. The same file size and mode of the test were kept to keep the testing criteria same) The below table gives a brief overview of the testing results.

Test number	Total Time Taken (Seconds)	Total Events occurred	Run Events per second
1	1.1632	10059	8647.70
2	1.1631	10056	8645.86
3	1.1924	10064	8440.12
4	1.1802	10068	8530.76
5	1.3216	10053	7606.69

The maximum number of observed events per second is 7606.69

The minimum number of observed events per second is 8647.70

The average of the 5 tests was observed to be 8374.22

6. Results of the comparison

The final results speak a lot about the difference of metrics over the type of virtualization implemented.

Let's compare the **CPU Max Prime** test results between both.

The system virtualization showed poor performance than OS virtualization as predicted since containers are lightweight than an entire VM which runs the entire operating system. The maximum events per second achieved by the system virtualization (VM = 169.34 events/s) were lower than the minimum events per second achieved by OS virtualization (Container = 171.60 events/s).

The average values differed between System Virtualization and OS virtualization by 50 events/s (48.83 Events/s exactly).

This goes to show that OS Virtualization was able to handle a lot more events per second on average than System Virtualization could.

When it comes to the File I/O test, the differences were far too great, that they pop right out.

System Virtualization

Test number	Prepare Time (Seconds)	Prepare Speed (Mbit/s)	Run Read throughput	Run Write Throughput	Run Events per second
1	30.32	270.19	9.98	6.64	2417.93
2	31.12	263.23	10.39	6.92	2520.02
3	24.82	371.31	10.05	6.69	2441.25
4	33.42	275.77	10.72	7.14	2559.55
5	29.15	316.14	10.52	7.01	2550.22

The maximum number of observed events per second is 2559.55

The minimum number of observed events per second is 2417.93

The average of the 5 tests was observed to be 2497.79

OS Virtualization

Test number	Total Time Taken (Seconds)	Total Events occurred	Run Events per second
1	1.1632	10059	8647.70
2	1.1631	10056	8645.86
3	1.1924	10064	8440.12
4	1.1802	10068	8530.76
5	1.3216	10053	7606.69

The maximum number of observed events per second is 7606.69

The minimum number of observed events per second is 8647.70

The average of the 5 tests was observed to be 8374.22

OS virtualization demolished the System Virtualization in terms of events per second alone. The amount of time it took for System Virtualization (VM) to perform the same test (in preparing time and run time combined) was exponentially larger than what it took for OS virtualization (Containers).

7. Conclusion

The conclusion is pretty more obvious in the case of CPU testing rather than File I/O since file I/O testing could vary from OS to OS. It could be understood from these tests conducted above that System Virtualization is slower when compared to OS virtualization with respect to the overhead of resources. But the security benefits more than makeup for the difference in speed and resource management.