HOMEWORK 1

○ Detailed configurations (CPU, Mem, etc...) of experimental setup:

Virtual Box:

OS Name: Ubuntu

Base memory 2048 MB

Storage: 30.23GB

No. of Processors: 1

Host Machine:

OS Name: Microsoft Windows 10 Home

Version: 10.0.19042 Build 19042

Processor Intel(R) Core (TM) i7-10710U CPU @ 1.10GHz, 1608 Mhz, 6 Core(s)

RAM: 16GB

Storage: 512GB SSD

○ Main steps to enable VM:

I installed Virtual Box instead of QEMU as I have windows machine. The main steps to enable Virtual Box is as follows:

1. Download the latest version of virtual box (6.1.26) for the required host.

2. Go to the location to which the VirtualBox EXE file downloaded and double-click the file. Doing so will open the VirtualBox installation window.

3. Install the software.

4. Add VirtualBox to the PATH which is the list of directories of which windows can run executables from. It can be done using following command:

$env:PATH = $env:PATH + ";C:\Program Files\Oracle\VirtualBox"

○ QEMU commands and VM configurations:

Detailed VirtualBox Commands and VM configurations are as follow:

1. To create a new virtual machine from the command line and immediately register it with Oracle VM VirtualBox, use VBoxManage createvm with the --register option, as follows:

VBoxManage createvm --name tanmay --register

2. To select or modify the operating system, use:

VBoxManage modifyvm tanmay --ostype Ubuntu_64

3. To set the RAM of the virtual machine, use:

VBoxManage modifyvm tanmay --memory 2048 --vram 16

4. To assign the number of CPU cores:

VBoxManage modifyvm demovm --cpus 1

5. To create virtual storage and set size for it:

VBoxManage createhd --filename tanmay.vdi --size 32768

6. To add storage controller and attach hard disk + ISO Image to boot

- VBoxManage storagectl tanmay --name "SATA Controller" --add sata --controller IntelAHCI
- VBoxManage storageattach tanmay --storagectl "SATA Controller" --port 0 --device 0 --type hdd --medium tanmay.vdi
- VBoxManage storageattach tanmay --storagectl "SATA Controller" --port 1 --device 0 --type dvddrive --medium "C:\Users\Tanmay Agrawal\Downloads\ubuntu-20.04.3-desktop-amd64"

7. To set boot order:

VBoxManage modifyvm demovm --boot1 dvd --boot2 disk --boot3 none --boot4 none

8. To list all the virtual machines:

vboxmanage list vms

9. To start the virtual machine:

Vboxmanage startvm “tanmay”

○ Steps to enable Docker container and operations to manage Docker containers:

To enable Docker container download and install Docker Desktop for windows.

In order to use the Docker image called csminpp/ubuntu-sysbench, which has sysbench preinstalled, I used the following command:

docker pull csminpp/ubuntu-sysbench

Following are some other operations which I think are also important:

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 running container.

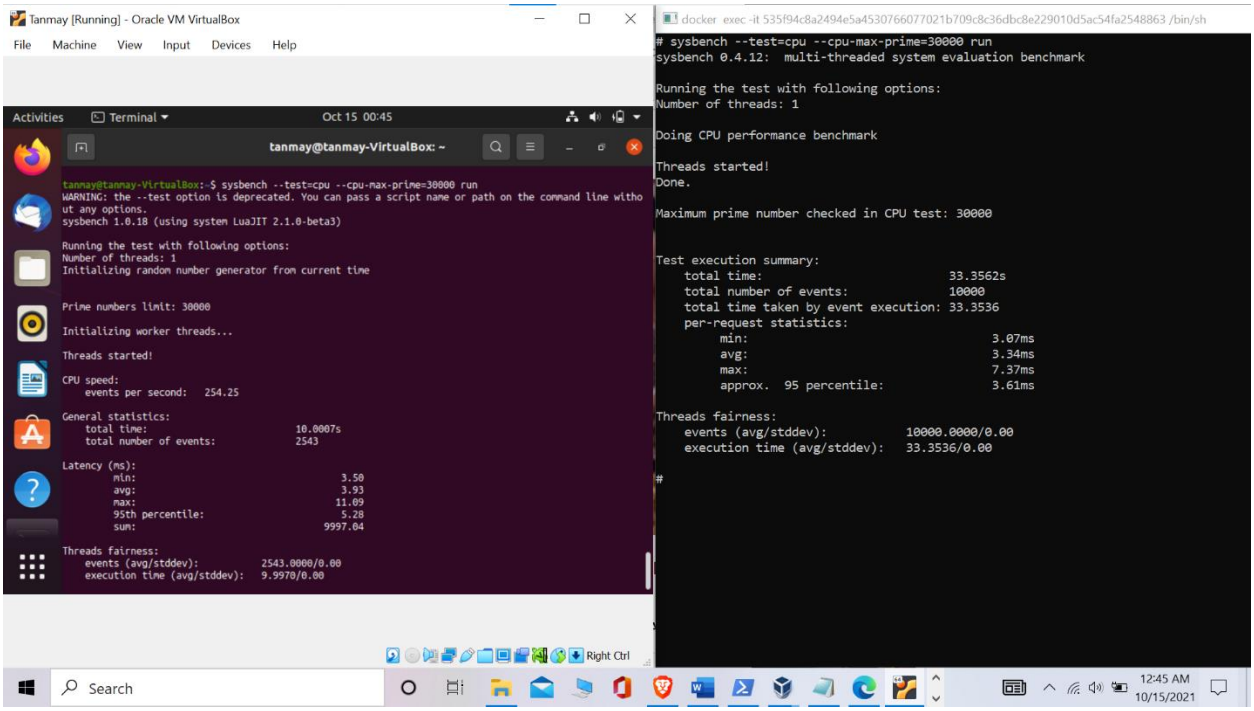
kill —Stop main process in container abruptly.

rm— Delete a stopped container.

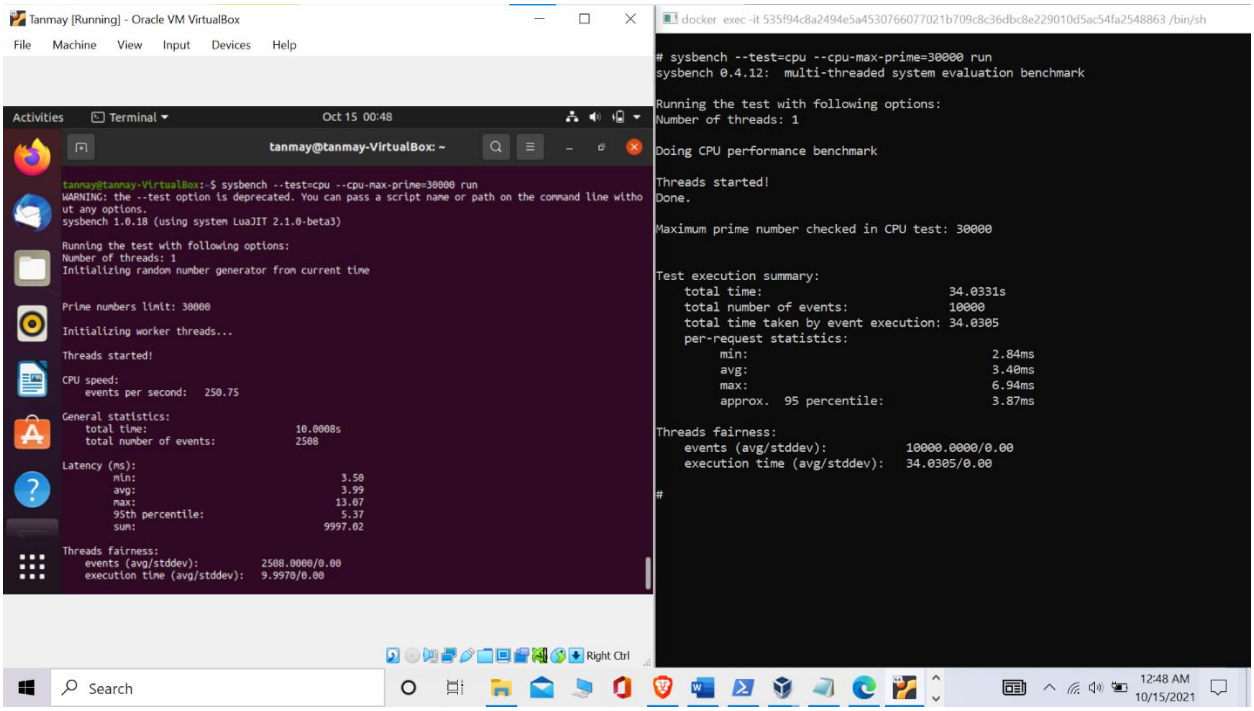
○ Proof of experiment

CPU TEST MODE

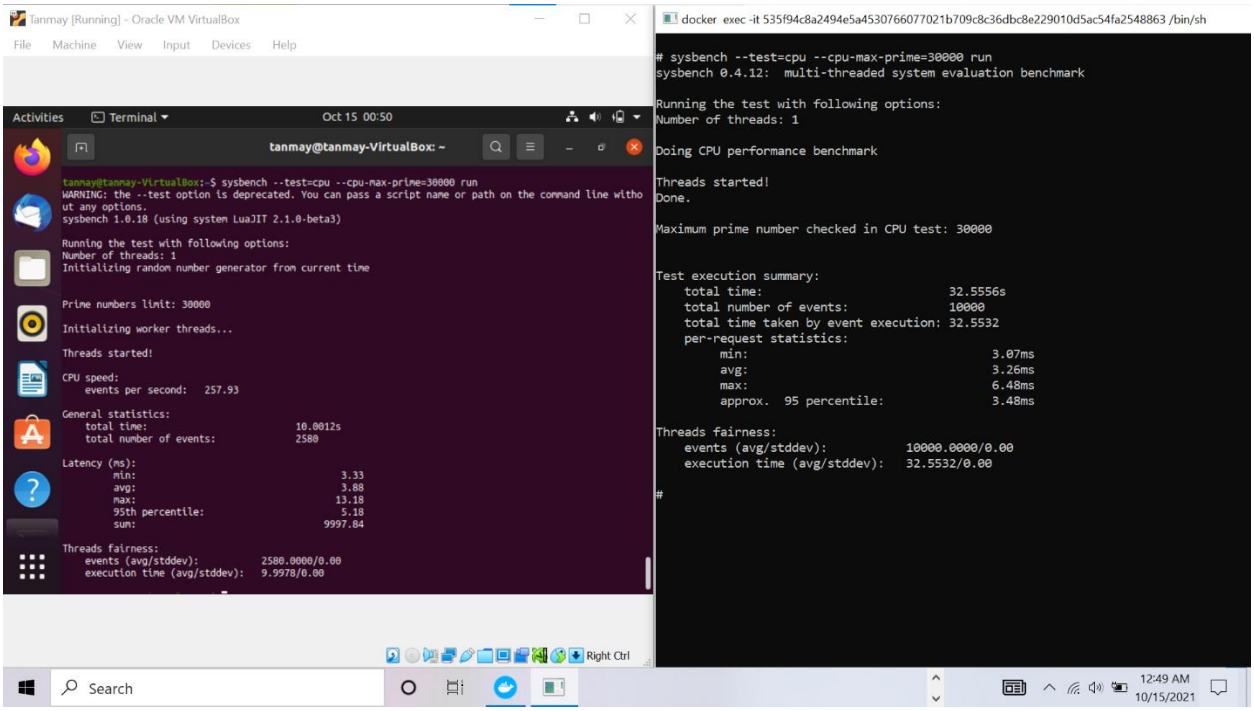
RUN 1:



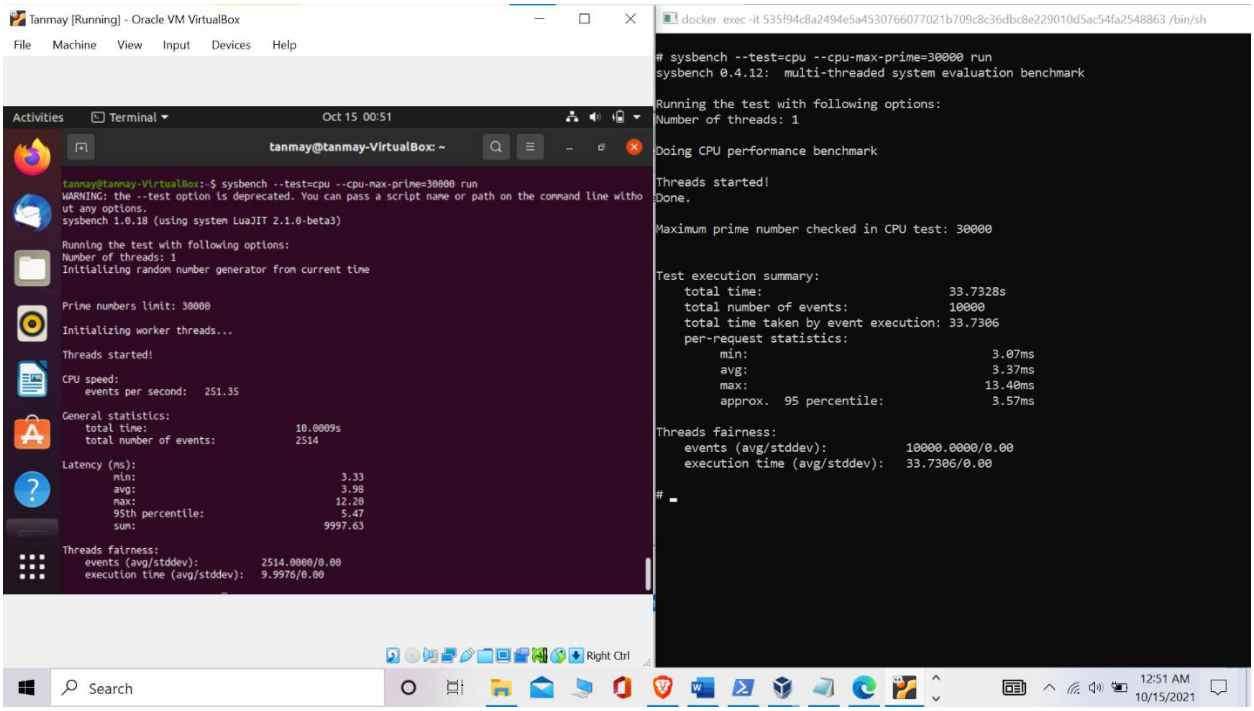
RUN 2:


Tanmay [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 15 00:48
tanmay@tanmay-VirtualBox: ~
tanmay@tanmay-VirtualBox:~$ 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 witho
ut 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: 250.75
General statistics:
total time: 10.0008s
total number of events: 2508
Latency (ms):
min: 3.50
avg: 3.99
max: 13.07
95th percentile: 5.37
sum: 9997.02
Threads fairness:
events (avg/stddev): 2508.0000/0.00
execution time (avg/stddev): 9.9970/0.00
docker exec -it 535f94c8a2494e5a4530766077021b709c8c36dbc8e229010d5ac54fa2548863 /bin/sh
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: 34.0331s
total number of events: 10000
total time taken by event execution: 34.0305
per-request statistics:
min: 2.84ms
avg: 3.40ms
max: 6.94ms
approx. 95 percentile: 3.87ms
Threads fairness:
events (avg/stddev): 10000.0000/0.00
execution time (avg/stddev): 34.0305/0.00
#
Right Ctrl
Search
12:48 AM
10/15/2021

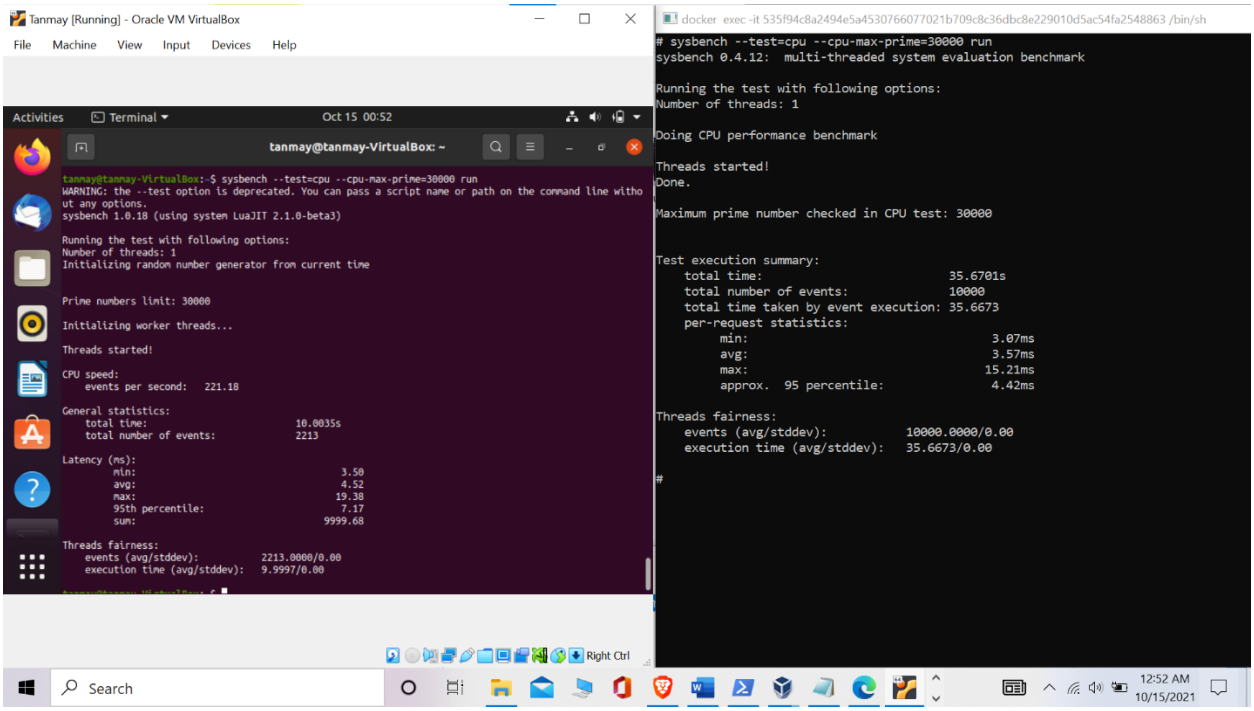
RUN 3:


Tanmay [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 15 00:50
tanmay@tanmay-VirtualBox: ~
tanmay@tanmay-VirtualBox:~$ 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 witho
ut 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: 257.93
General statistics:
total time: 10.0012s
total number of events: 2580
Latency (ms):
min: 3.33
avg: 3.88
max: 13.18
95th percentile: 5.18
sum: 9997.84
Threads fairness:
events (avg/stddev): 2580.0000/0.00
execution time (avg/stddev): 9.9978/0.00
docker exec -it 535f94c8a2494e5a4530766077021b709c8c36dbc8e229010d5ac54fa2548863 /bin/sh
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: 32.5556s
total number of events: 10000
total time taken by event execution: 32.5532
per-request statistics:
min: 3.07ms
avg: 3.26ms
max: 6.48ms
approx. 95 percentile: 3.48ms
Threads fairness:
events (avg/stddev): 10000.0000/0.00
execution time (avg/stddev): 32.5532/0.00
#
Right Ctrl
Search
12:49 AM
10/15/2021

RUN 4:

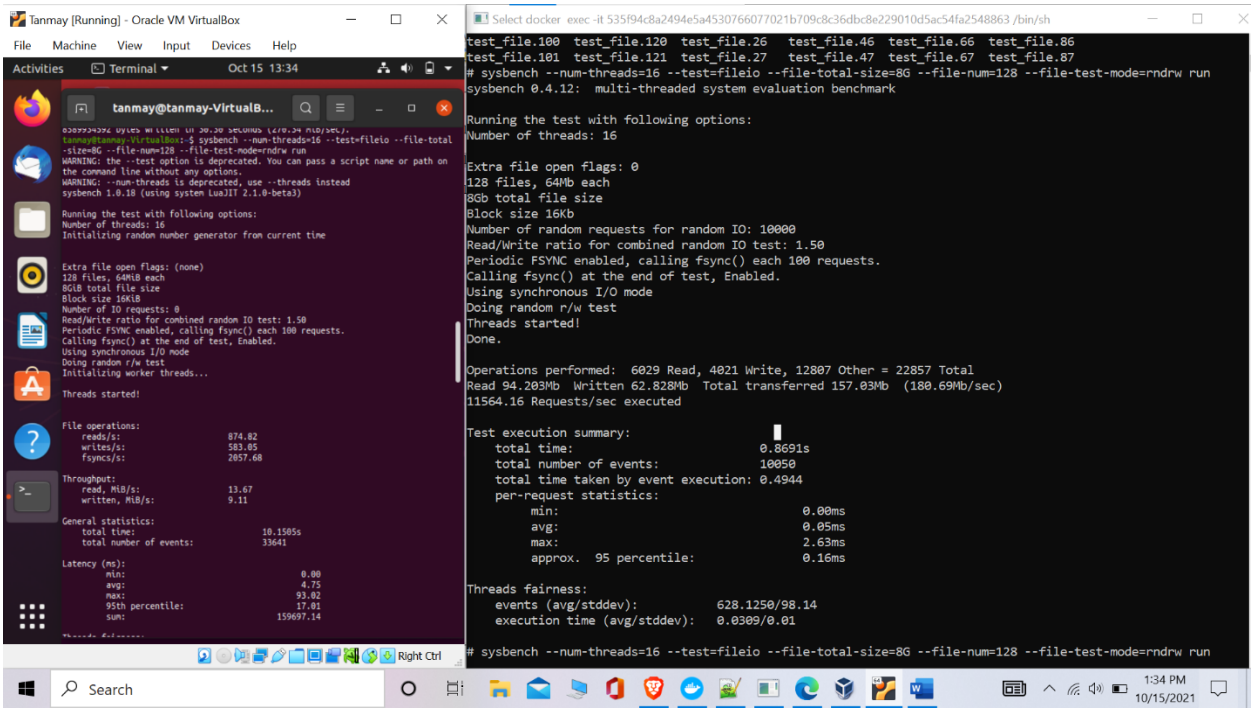

Tanmay [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 15 00:51
tanmay@tanmay-VirtualBox: ~
tanmay@tanmay-VirtualBox:~$ sysbench --test=cpu --cpu-max-prime=30000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command 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: 251.35
General statistics:
total time: 10.0009s
total number of events: 2514
Latency (ms):
min: 3.33
avg: 3.98
max: 12.20
95th percentile: 5.47
sum: 9997.63
Threads fairness:
events (avg/stddev): 2514.0000/0.00
execution time (avg/stddev): 9.9976/0.00
docker exec -it 535f94c8a2494e5a4530766077021b709c8c36dbc8e229010d5ac54fa2548863 /bin/sh
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: 33.7328s
total number of events: 10000
total time taken by event execution: 33.7306
per-request statistics:
min: 3.07ms
avg: 3.37ms
max: 13.40ms
approx. 95 percentile: 3.57ms
Threads fairness:
events (avg/stddev): 10000.0000/0.00
execution time (avg/stddev): 33.7306/0.00
#
Right Ctrl
Search
12:51 AM
10/15/2021

RUN 5:

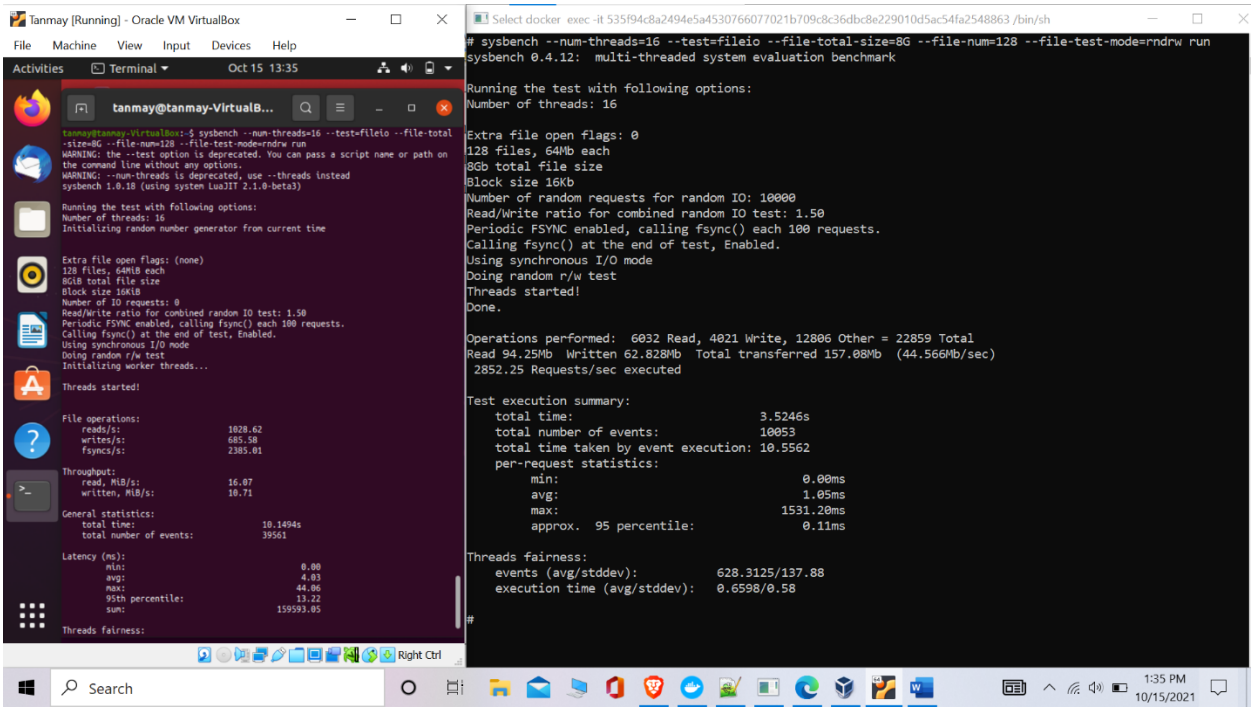

Tanmay [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 15 00:52
tanmay@tanmay-VirtualBox: ~
tanmay@tanmay-VirtualBox:~$ sysbench --test=cpu --cpu-max-prime=30000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command 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: 221.18
General statistics:
total time: 10.0035s
total number of events: 2213
Latency (ms):
min: 3.50
avg: 4.52
max: 19.38
95th percentile: 7.17
sum: 9999.68
Threads fairness:
events (avg/stddev): 2213.0000/0.00
execution time (avg/stddev): 9.9997/0.00
docker exec -it 535f94c8a2494e5a4530766077021b709c8c36dbc8e229010d5ac54fa2548863 /bin/sh
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: 35.6701s
total number of events: 10000
total time taken by event execution: 35.6673
per-request statistics:
min: 3.07ms
avg: 3.57ms
max: 15.21ms
approx. 95 percentile: 4.42ms
Threads fairness:
events (avg/stddev): 10000.0000/0.00
execution time (avg/stddev): 35.6673/0.00
#
Right Ctrl
Search
12:52 AM
10/15/2021

FILEIO TEST MODE

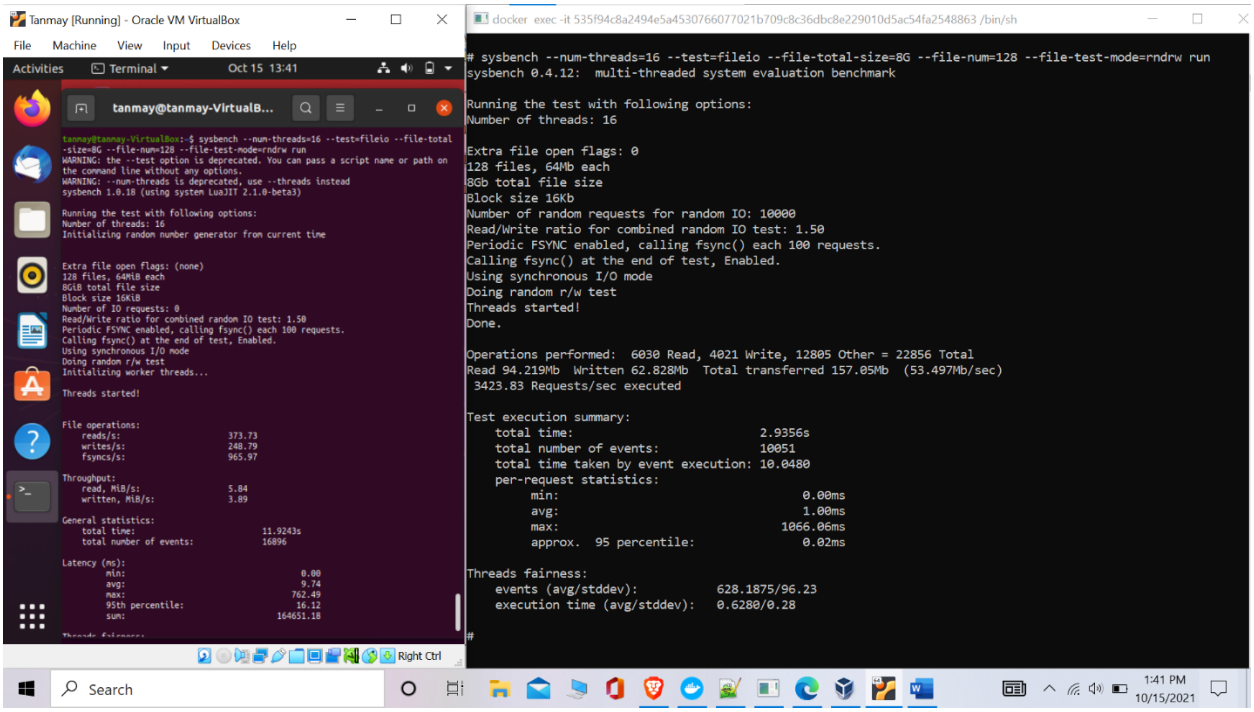
RUN 1:



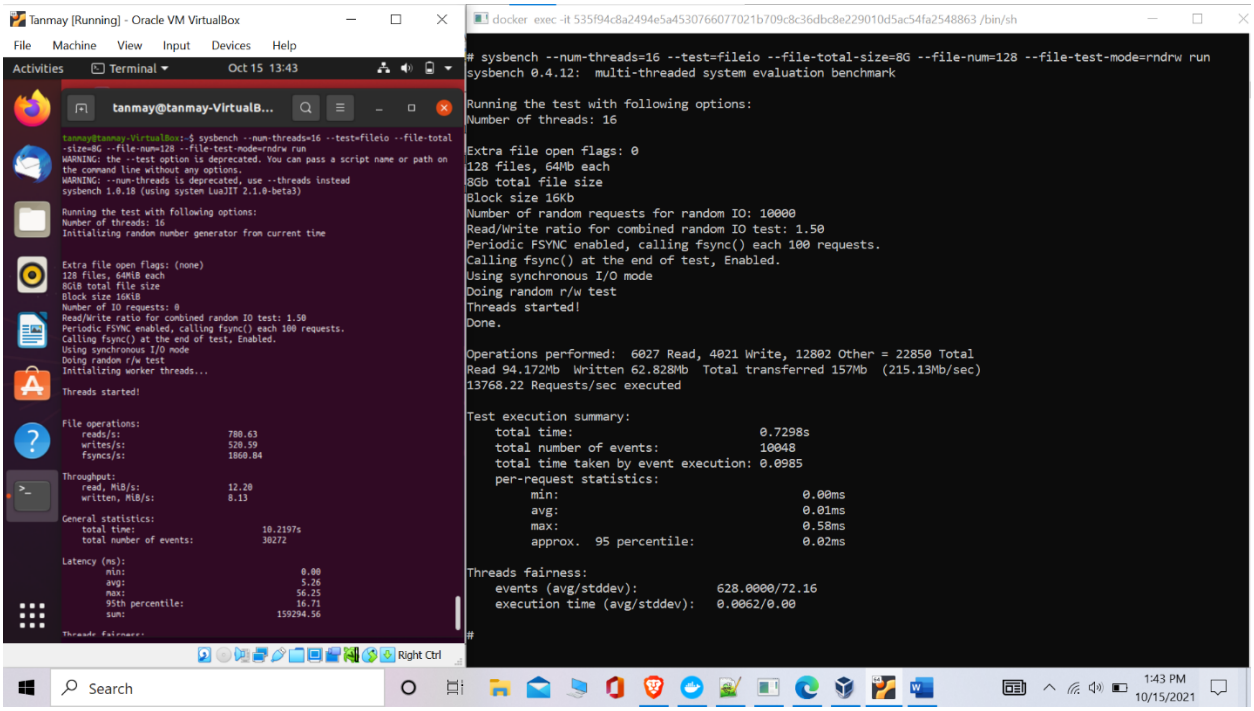
RUN 2:



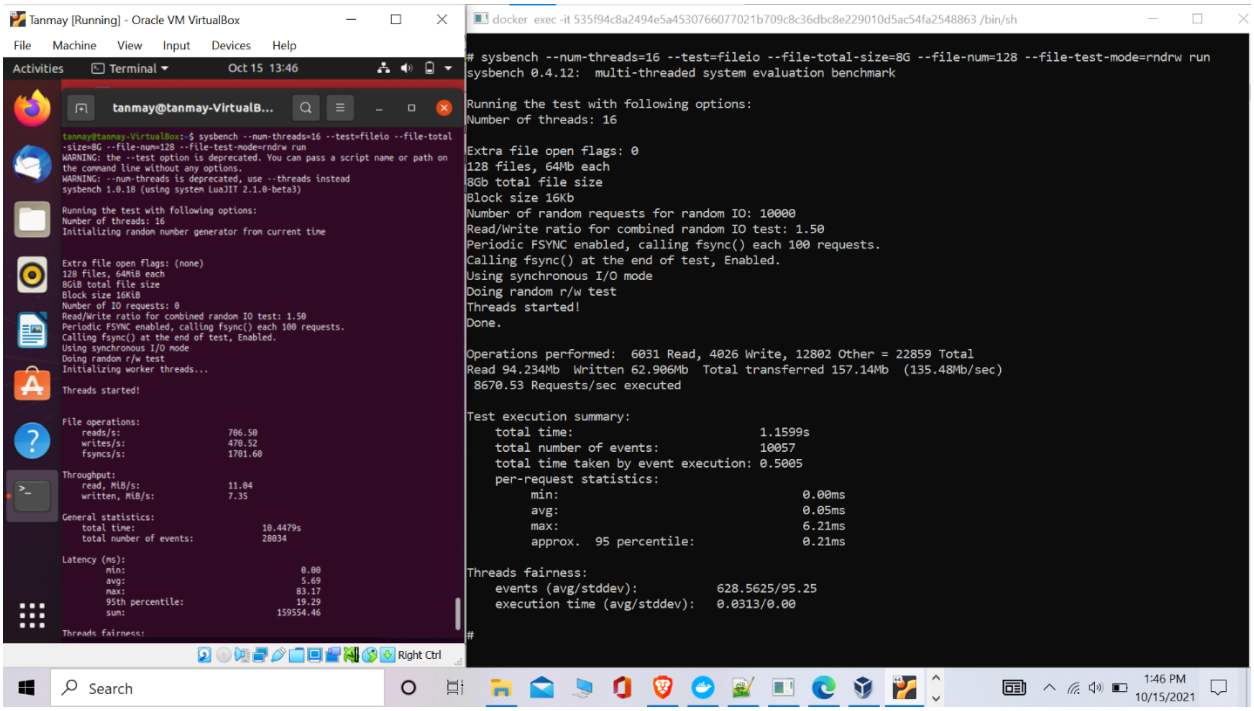
RUN 3:


Tanmay [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 15 13:41
tanmay@tanmay-VirtualB...
docker exec -it 535f94c8a2494e5a4530766077021b709c8c36dbc8e229010d5ac54fa2548863 /bin/sh
sysbench --num-threads=16 --test=fileio --file-total-size=8G --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, 64Mb each
8Gb 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, 4021 Write, 12805 Other = 22856 Total
Read 94.219Mb Written 62.828Mb Total transferred 157.05Mb (53.497Mb/sec)
3423.83 Requests/sec executed
Test execution summary:
total time: 2.9356s
total number of events: 10051
total time taken by event execution: 10.0480
per-request statistics:
min: 0.00ms
avg: 1.00ms
max: 1066.06ms
approx. 95 percentile: 0.02ms
Threads fairness:
events (avg/stddev): 628.1875/96.23
execution time (avg/stddev): 0.6280/0.28
Right Ctrl
Search
1:41 PM
10/15/2021

RUN 4:


Tanmay [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 15 13:43
tanmay@tanmay-VirtualB...
docker exec -it 535f94c8a2494e5a4530766077021b709c8c36dbc8e229010d5ac54fa2548863 /bin/sh
sysbench --num-threads=16 --test=fileio --file-total-size=8G --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, 64Mb each
8Gb 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: 6027 Read, 4021 Write, 12802 Other = 22850 Total
Read 94.172Mb Written 62.828Mb Total transferred 157Mb (215.13Mb/sec)
13768.22 Requests/sec executed
Test execution summary:
total time: 0.7298s
total number of events: 10048
total time taken by event execution: 0.0985
per-request statistics:
min: 0.00ms
avg: 0.01ms
max: 0.58ms
approx. 95 percentile: 0.02ms
Threads fairness:
events (avg/stddev): 628.0000/72.16
execution time (avg/stddev): 0.0062/0.00
Right Ctrl
Search
1:43 PM
10/15/2021

RUN 5:



○ Steps to conduct measurements in each virtualization technology:

In case of system virtualization, I installed Sybench on Ubuntu using following commands:

$sudo apt update

$sudo apt install sysbench

Then, for CPU test mode I ran the following command:

sysbench --test=cpu --cpu-max-prime=30000 run

After that I noted the CPU speed, i.e., the number of events being executed per second.

While, in case of OS virtualization, I used the Docker image called csminpp/ubuntu-sysbench, which has sysbench preinstalled. To find the image, I ran the following command:

docker pull csminpp/ubuntu-sysbench

Then, for CPU test mode I ran the following command in Docker CLI:

sysbench --test=cpu --cpu-max-prime=30000 run

After that I noted the CPU speed, i.e., the number of events being executed per second.

On comparing, CPU speed for the scenario I concluded that CPU speed in OS virtualization is faster when compared against system virtualization.

○ Shell Script for running CPU test mode (5 time):

```
for i in {1..5};

do

sysbench --test=cpu --cpu-max-prime=30000 run;

done
```

○ Shell Script for running Fileio test mode (5 time):

```
for i in {1..5};

do

sysbench --num-threads=16 --test=fileio --file-total-size=8G --file-num=128 --file-test-mode=rndrw prepare


sysbench --num-threads=16 --test=fileio --file-total-size=8G --file-num=128 --file-test-mode=rndrw run


sysbench --num-threads=16 --test=fileio --file-total-size=8G --file-num=128 --file-test-mode=rndrw cleanup;

done
```

○ Steps to use Performance tools along with Performance Data

I ran the following command on the terminal of my VM and also from the Docker CLI for CPU test mode:

```
sysbench --test=cpu --cpu-max-prime=30000 run;
```

After this I noted down the performance data is below:

RUN	CPU Test Mode (System Virtualization)						
	Total number of events	Total time	CPU Speed	Latency(in ms)			
				min	avg	max	95th percentile
1	2543	10.0007	254.25	3.5	3.93	11.09	5.28
2	2508	10.0008	250.75	3.5	3.99	13.07	5.37
3	2580	10.0012	257.93	3.33	3.88	13.18	5.18
4	2514	10.0009	251.35	3.33	3.98	12.2	5.47
5	2213	10.0035	221.18	3.5	4.52	19.38	7.17

RUN	CPU Test Mode (OS Virtualization)						
	Total number of events	Total time	CPU Speed	Latency(in ms)			
				min	avg	max	95th percentile
1	10000	33.3562	299.7943411	3.07	3.34	7.37	3.61
2	10000	34.0331	293.8315934	2.84	3.4	6.94	3.87
3	10000	32.5556	307.1668162	3.07	3.26	6.48	3.48
4	10000	33.7328	296.4473747	3.07	3.37	13.4	3.57
5	10000	35.6701	280.3468451	3.07	3.57	15.21	4.42

From the above data, we can easily conclude that CPU speed in case of OS virtualization.

In case of system virtualization, I ran the following command on the terminal of my VM and also from the Docker CLI:

At the prepare stage SysBench creates a specified number of files with a specified total size

sysbench --num-threads=16 --test=fileio --file-total-size=8G --file-num=128 --file-test-mode=rndrw prepare

Then at the run stage, each thread performs specified I/O operations on this set of files

sysbench --num-threads=16 --test=fileio --file-total-size=8G --file-num=128 --file-test-mode=rndrw run

After this I noted down the performance data is below:

RUN	Fileio Test Mode (System Virtualization)						
	Throughput Mb/sec			Latency(in ms)			
	Read	Write	Total Transferred	min	avg	max	95th percentile
1	13.67	9.11	22.78	0	4.75	93.02	17.01
2	16.07	10.71	26.78	0	4.03	44.06	13.22
3	5.84	3.89	9.73	0	9.74	762.49	16.12
4	12.2	8.13	20.33	0	5.26	56.25	16.71
5	11.04	7.35	18.39	0	5.69	83.17	19.29
RUN	Fileio Test Mode (OS Virtualization)						
	Throughput Mb/sec			Latency(in ms)			
	Read	Write	Total Transferred	min	avg	max	95th percentile
1	94.203	62.828	157.031	0	0.05	2.63	0.16
2	94.25	62.828	157.078	0	1.05	1531.2	0.11
3	94.219	62.828	157.047	0	1	1066.06	0.02
4	94.172	62.828	157	0	0.01	0.58	0.02
5	94.234	62.906	157.14	0	0.05	6.21	0.21

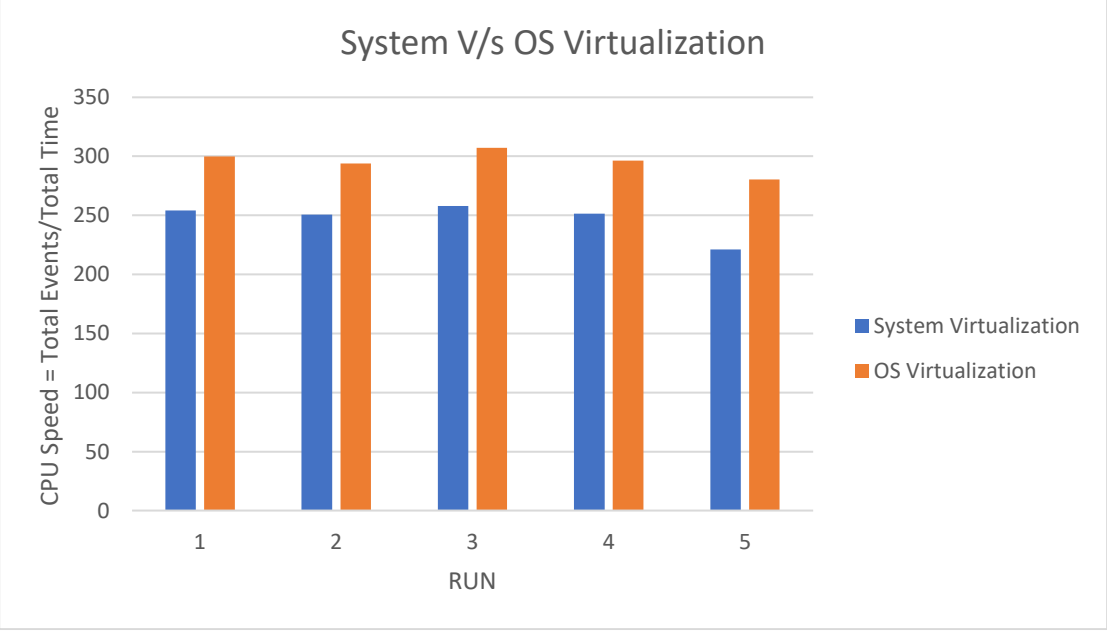
Finally, at the cleanup stage, all the files used for the test are removed.

sysbench --num-threads=16 --test=fileio --file-total-size=8G --file-num=128 --file-test-mode=rndrw cleanup

After analyzing all the parameters like throughput, latency and disk utilization, we can conclude that fileio operations takes faster in OS virtualization as compared to system virtualization.

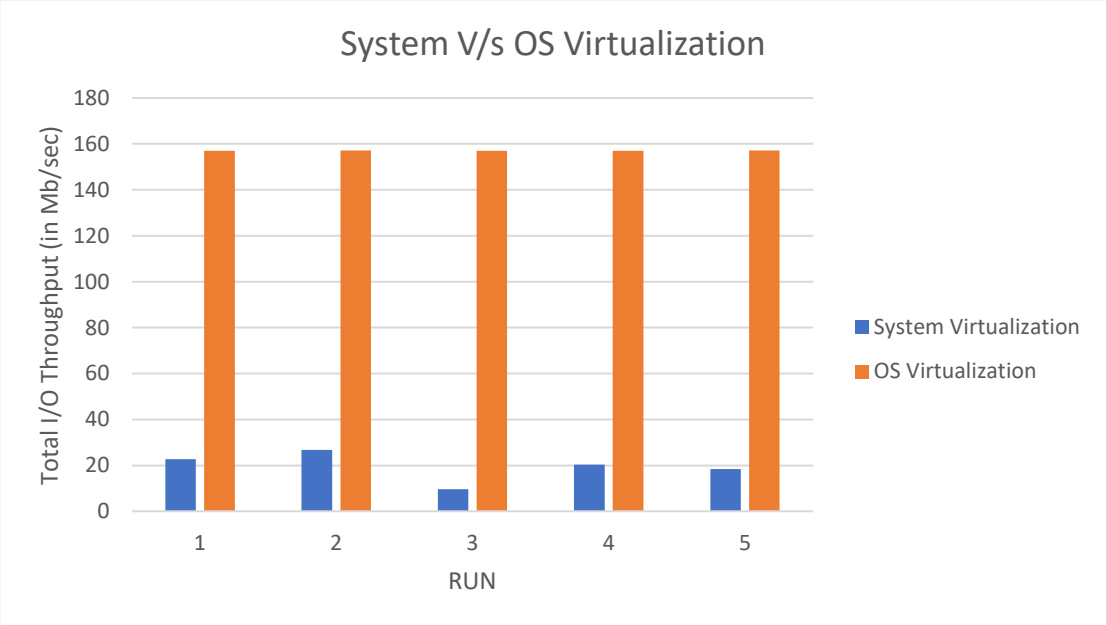
○ Presentation and analysis of the performance data:

RUN	CPU Test Mode (System Virtualization)		
	Total number of events	Total time	CPU Speed
1	2543	10.0007	254.25
2	2508	10.0008	250.75
3	2580	10.0012	257.93
4	2514	10.0009	251.35
5	2213	10.0035	221.18
RUN	CPU Test Mode (OS Virtualization)		
	Total number of events	Total time	CPU Speed
1	10000	33.3562	299.7943411
2	10000	34.0331	293.8315934
3	10000	32.5556	307.1668162
4	10000	33.7328	296.4473747
5	10000	35.6701	280.3468451



On analyzing CPU speed in both the types of virtualizations, we find that OS virtualization performs better than the system virtualization. It is quite evident from the graph above.

RUN	Fileio Test Mode (System Virtualization)		
	Throughput Mb/sec		
	Read	Write	Total Transferred
1	13.67	9.11	22.78
2	16.07	10.71	26.78
3	5.84	3.89	9.73
4	12.2	8.13	20.33
5	11.04	7.35	18.39
RUN	Fileio Test Mode (OS Virtualization)		
	Throughput Mb/sec		
	Read	Write	Total Transferred
1	94.203	62.828	157.031
2	94.25	62.828	157.078
3	94.219	62.828	157.047
4	94.172	62.828	157
5	94.234	62.906	157.14



On analyzing total throughput in both the types of virtualizations, we find that OS virtualization handles the file I/O workloads much better than system virtualization. It is quite evident from the graph above.

○ Git Repository Information:

https://github.com/tanmay24497/hw1.git

○ Vagrant file:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure("2") do |config|
 config.vm.define "tanmay"
 config.vm.hostname = "tanmay"
 config.vm.box = "bento/ubuntu-16.04"
 config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
 # Argument 1: path on the host to the actual folder.
 #Argument 2: path on the guest to mount the folder.
 config.vm.synced_folder "../data", "/vagrant_data"
  config.vm.provider "virtualbox" do |vb|
    vb.name = "tanmay"
    vb.gui = false
    vb.memory = "2048"
    vb.disk :disk, size: "10GB", primary: true
    vb.disk :dvd, name: "ubuntu iso file", file: "./ ubuntu-20.04.3-desktop-amd64 "
  end
  config.vm.provision "shell", inline: <<-SHELL
    apt-get update
    apt-get install -y sysbench
  SHELL
     config.vm.provision "shell", run:"always", inline: <<-SHELL
            echo "Hi VM is successfully configured using vagrant"
           SHELL
End
```

○ **Dockerfile:**

```
FROM csminpp/ubuntu-sysbench

MAINTAINER tanmay agrawal <tagrawal@scu.edu>

RUN apt-get update

CMD ["echo", "Hi this is my first docker image atumatically built using dockerfile"]
```