Praveen Vandeyar
COEN 241
10/12/2021

<u>HW 1</u>

QEMU
Ubuntu 20.04_03 Server
10 GB Storage
2 GB Memory
Undefined | 1 CPU Core | 2 CPU Cores
No Hardware Acceleration

Docker
Allowed full resources
312 GB Storage
8 GB Memory
4 CPU Cores

sudo apt-get install qemu # Install QEMU

sudo qemu-img create ubuntu.img 10G # Allocate space for the VM

Download a boot file to create VM with, like the Ubuntu ISO file

sudo qemu-system-x86_64 -hda ubuntu.img -boot d -cdrom ./ubuntu-20.04.3-live-server-amd64.iso -m 1536 # Initial boot/install of the VM with the iso file to the img file

sudo qemu-system-x86_64 -hda ubuntu.img -smp cores=2 -m 2048 # Start VM from stored img file

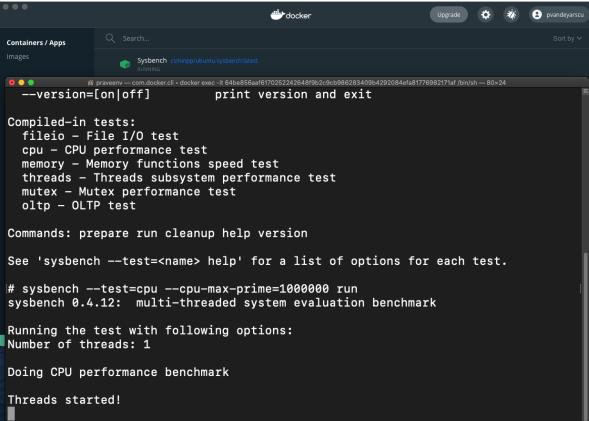Install Docker # Docker Desktop

Start Docker Engine # Start the container engine consisting of the docker service and containerd service

Pull Sysbench image # Install the application

Run Sysbench # Run the application within a Docker container

Open CLI for Sysbench container # Interact with the application

QEMU - Press Ctrl+Alt+G to release grab

Machine   View

```
General statistics:
    total time:                    10.7507s
    total number of events:        8

Latency (ms):
         min:                          1321.89
         avg:                          1342.43
         max:                          1377.09
         95th percentile:              1376.60
         sum:                         10739.48

Threads fairness:
    events (avg/stddev):          8.0000/0.00
    execution time (avg/stddev):  10.7395/0.00

pvandeyar@pvandeyarserver:~$ sysbench cpu --cpu-max-prime=1000000 run
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: 1000000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:     0.75

General statistics:
    total time:                    10.6946s
    total number of events:        8

Latency (ms):
         min:                          1315.93
         avg:                          1335.43
         max:                          1366.28
         95th percentile:              1376.60
         sum:                         10683.46

Threads fairness:
    events (avg/stddev):          8.0000/0.00
    execution time (avg/stddev):  10.6835/0.00

pvandeyar@pvandeyarserver:~$
```

Left terminal window:

```
Could not acce
qemu-system-x8
pvandeyar@Prav
8
pvandeyar@Prav
INFO: /dev/kvm
HINT:   sudo
modprobe: FATA
osoft
pvandeyar@Prav
Reading packag
Building depen
Reading state
cpu-checker is
cpu-checker se
0 upgraded, 0
pvandeyar@Prav
2048
WARNING: Image
.
         Autom
perations on b
         Speci
```


docker   Upgrade   pvandeyarscu

Containers / Apps   Search...   Sort by

Images

Sysbench csminpp/ubuntu-sysbench:latest
RUNNING

praveenv — com.docker.cli • docker exec -it 64be856aaf6170252242648f9b2c9cb986283409b4292084efa81776982171af /bin/sh — 80×24

```
  --version=[on|off]          print version and exit

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

Commands: prepare run cleanup help version

See 'sysbench --test=<name> help' for a list of options for each test.

# sysbench --test=cpu --cpu-max-prime=1000000 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!
```

CPU

	Total Time	Min	Avg	Max	95th	Event STD

1000000

	Total Time	Min	Avg	Max	95th	Event STD
2 GB	10.8627	1318.24	1356.48	1388.75	1376.60	8
1 CPU	10.9088	1350.74	1362.14	1369.99	1376.60	8
2 CPU	10.8781	1354.21	1359.16	1363.72	1352.03	8
Docker	10.0154	370.49	385.18	385.18	397.39	26

4000000

	Total Time	Min	Avg	Max	95th	Event STD
2 GB	19.0182	9481.26	9503.71	9526.16	9452.83	2
1 CPU	19.0820	9451.63	9534.68	9617.74	9624.59	2
2 CPU	19.0476	9453.13	9521.46	9589.80	9624.59	2
Docker	10.9091	2650.44	2727.09	2770.88	2778.39	4

9000000

	Total Time	Min	Avg	Max	95th	Event STD
2 GB	29.6797	29669.28	29669.28	29669.28	29926.15	1
1 CPU	30.0932	30082.30	30082.30	30082.30	29926.15	1
2 CPU	29.9495	29936.75	29936.75	29936.75	29926.15	1
Docker	17.2665	8616.82	8632.89	8648.97	8638.96	2

Memory instead of FileIO due to errors on QEMU machine

	Total Time	Min	Avg	Max	Event STD

5 GB Read

	Total Time	Min	Avg	Max	Event STD
2 GB	4.5837	0	0	1.37	5242880
1 CPU	4.7980	0.00	0.00	0.67	5242880
2 CPU	10.0009	0.00	0.00	10.89	1913983
Docker	10.0001	0.00	0.00	0.12	4619489

10 GB Read

	Total Time	Min	Avg	Max	Event STD
2 GB	9.2217	0	0	1.31	10485760
1 CPU	9.2312	0.00	0.00	0.68	10485760
2 CPU	10.0011	0.00	0.00	4.77	1920542
Docker	10.0001	0.00	0.00	1.53	4427045

5 GB Write

	Total Time	Min	Avg	Max	Event STD
2 GB	5.2739	0	0	14.96	5242880
1 CPU	5.0981	0.00	0.00	19.06	5242880
2 CPU	10.0026	0.00	0.00	1.40	1765795
Docker	10.0001	0.00	0.00	6.80	4383380

Scripts

```
#!/bin/bash

sysbench cpu --cpu-max-prime=1000000 run

sysbench cpu --cpu-max-prime=4000000 run

sysbench cpu --cpu-max-prime=9000000 run
```

```
#!/bin/bash

#Keep getting a fatal error for fileio in QEMU, one of the prepared files is said to be less than
the minimum which changes when I change or remove an option

sysbench --test=fileio --file-total-size=4G prepare

sysbench --test=fileio --file-total-size=4G --file-test-mode=rndrw --max-time=180
--max-requests=0 --file-extra-flags=direct run

sysbench --test=fileio --file-total-size=4G cleanup
```

```
#!/bin/bash

sysbench --test=memory --memory-block-size=1K --memory-scope=global
--memory-total-size=5G --memory-oper=read run

sysbench --test=memory --memory-block-size=1K --memory-scope=global
--memory-total-size=10G --memory-oper=read run

sysbench --test=memory --memory-block-size=1K --memory-scope=global
--memory-total-size=5G --memory-oper=write run
```

With the performance results, it can be seen that a container has better performance than a virtual machine all around. Unfortunately I couldn't get fileio to work due to the prepared file size fatal error, but it was interesting to see the memory results. It seems that having access to higher CPU cores gives a worse performance. For Docker, the lower load read time was the same as the higher load read time, whereas the QEMU virtual machine had an increase in time. For higher loads, it is very likely that the time for Docker would be lower than the QEMU virtual machine, and the current data is just a result of a bad test case as it seems that there is a floor for Docker memory test.

https://github.com/pv-gitjob/coen241.git
https://github.com/pv-gitjob/coen241