

COEN 241: CLOUD COMPUTING

ASSIGNMENT - 1

1. QEMU

1. We will go over the steps for installing QEMU in this section of the report before moving on to creating a QEMU Disk Image. Throughout the length of this report, we will conduct all of our analyses with this image. The installation will vary based on each person's system setup. I'll describe what I installed for a MacBook Pro with M1 chips in this article (Also known as Apple Silicon).
2. Since we will be using this image for all of our experiments, it is crucial that we get the appropriate ISO image for the configuration of our system. I created an Ubuntu QEMU disk image because I decided to use Ubuntu 20.04 for the assignment.
3. I downloaded the image from the following link:
<https://cdimage.ubuntu.com/releases/20.04/release/ubuntu-20.04.4-live-server-arm64.iso>
4. Next we need to install homebrew. You can install homebrew using the following link: <https://brew.sh/>
5. For installing Qemu, on the mac terminal run the command : **brew install qemu**
6. In order to later install your Ubuntu OS, create a QEMU empty image. Run the following command to do that:
qemu-img create -f raw ubuntu-latest.raw 40G
7. To install the VM, we need to run the following commands:
**qemu-system-aarch64 **
**-monitor stdio **
**-M virt,highmem=off **
**-accel hvf **
**-cpu host **
**-smp 4 **
**-m 3000 **
**-bios QEMU_EFI.fd **
**-device virtio-gpu-pci **
**-display default,show-cursor=on **

```
-device qemu-xhci \  
-device usb-kbd \  
-device usb-tablet \  
-device intel-hda \  
-device hda-duplex \  
-drive file=ubuntu-latest.raw,format=raw,if=virtio,cache=writethrough \  
-cdrom ubuntu-20.04.5-live-server-arm64.iso
```

8. Then, after carefully adhering to the installation instructions displayed on the screen, your system will have a QEMU Ubuntu VM image installed.
9. The same set of commands as before are used to run your image; the cdrom argument is simply omitted.
10. Important arguments :
 1. -m: Denotes memory. Since we specified 2G as the input value in the command above, we are allocating 2GB of RAM capacity for our VM.
 2. Hardware acceleration is denoted by the suffix "-accel."
 3. -smp: The number of cores is indicated. We have specified the value of the parameter as 2, which indicates that we have provided our VM 2 cores.
11. Thus, we have successfully installed QEMU and made a QEMU VM in our system

```
QEMU
Install complete! [ Help ]

configuring apt configuring apt
installing missing packages
Installing packages on target system: ['efibootmgr',
'grub-efi-arm64', 'grub-efi-arm64-signed', 'shim-signed']
configuring iscsi service
configuring raid (mdadm) service
installing kernel
setting up swap
apply networking config
writing etc/fstab
configuring multipath
updating packages on target system
configuring pollinate user-agent on target
updating initramfs configuration
configuring target system bootloader
installing grub to target devices
finalizing installation
running 'curtin hook'
curtin command hook
executing late commands
final system configuration
configuring cloud-init
calculating extra packages to install
downloading and installing security updates
curtin command in-target
restoring apt configuration
curtin command in-target
subiquity/Late/run

[ View full log ]
[ Reboot Now ]
```

2. Docker

1. Go to <https://docs.docker.com/desktop/mac/apple-silicon/> for downloading docker for Apple mac M1.
2. Run the installer and perform the following action:



3. Open the Docker application in the application folder and the Docker engine starts running.
4. Install sysbench using:

```
$ sudo apt update
```

```
$ sudo apt install sysbench
```

```
manasibendale — root@3f5a2e03e760: / — com.docker.cli < docker run -it --entrypoint /bin/...
Fetched 19.1 MB in 4s (5090 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@3f5a2e03e760:/# apt install sysbench
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  krb5-locales libaio1 libasn1-8-heimdal libgssapi-krb5-2 libgssapi3-heimdal
  libhcrypto4-heimdal libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-heimdal libk5crypto3
  libkeyutils1 libkrb5-26-heimdal libkrb5-3 libkrb5support0 libldap-2.4-2 libldap-common
  liblua5.1-2 liblua5.1-common libmysqlclient21 libpq5 libroken18-heimdal libsasl2-2
  libsasl2-modules libsasl2-modules-db libsqlite3-0 libssl1.1 libwind0-heimdal mysql-common
Suggested packages:
  krb5-doc krb5-user libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal
  libsasl2-modules-ldap libsasl2-modules-otp libsasl2-modules-sql
The following NEW packages will be installed:
  krb5-locales libaio1 libasn1-8-heimdal libgssapi-krb5-2 libgssapi3-heimdal
  libhcrypto4-heimdal libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-heimdal libk5crypto3
  libkeyutils1 libkrb5-26-heimdal libkrb5-3 libkrb5support0 libldap-2.4-2 libldap-common
  liblua5.1-2 liblua5.1-common libmysqlclient21 libpq5 libroken18-heimdal libsasl2-2
  libsasl2-modules libsasl2-modules-db libsqlite3-0 libssl1.1 libwind0-heimdal mysql-common
  sysbench
0 upgraded, 29 newly installed, 0 to remove and 0 not upgraded.
Need to get 5015 kB of archives.
After this operation, 20.2 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Experiments:

1. CPU TESTING

We will employ the following 3 test cases to evaluate CPU performance between QEMU and Docker. The following sysbench command and the related test cases will be used for our testing:

```
sysbench cpu --cpu-max-prime={some_value} --num-threads={some_value} --time={some_value} run
```

a) Docker Ubuntu VM

i) Experiment -1 sysbench cpu --threads=1 --cpu-max-prime=50000
--time=10 run

```

root@3f5a2e03e760:/# sysbench cpu --threads=1 --cpu-max-prime=50000 --time=10 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: 50000
Initializing worker threads...

Threads started!

CPU speed:
  events per second: 1222.63

General statistics:
  total time:          10.0007s
  total number of events: 12228

Latency (ms):
  min:                 0.80
  avg:                 0.82
  max:                 2.07
  95th percentile:    0.83
  sum:                9991.95

Threads fairness:
  events (avg/stddev): 12228.0000/0.00
  execution time (avg/stddev): 9.9920/0.00

```

After 5 iterations:

Test Run	Total Time	CPU Speed	Avg Latency
1	10.0007	12228	0.82
2	10.0007	12220	0.82
3	10.0007	12214	0.82
4	10.0009	12087	0.83
5	10.0005	12093	0.83
Minimum	10.0005	12087	0.82
Maximum	10.0009	12228	0.83
Average	10.00076	12228.4	0.82
StdDev	0.00223607	9.327379	0.013198

ii) Experiment -1 sysbench cpu --threads=1 --cpu-max-prime=250000
--time=10 run

```

[root@3f5a2e03e760:/# sysbench cpu --threads=1 --cpu-max-prime=250000 --time=10 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: 250000
Initializing worker threads...
Threads started!

CPU speed:
  events per second:   139.19

General statistics:
  total time:          10.0066s
  total number of events: 1393

Latency (ms):
  min:                 7.04
  avg:                 7.18
  max:                 18.22
  95th percentile:    7.30
  sum:                 9995.90

Threads fairness:
  events (avg/stddev): 1393.0000/0.00
  execution time (avg/stddev): 9.9959/0.00

```

After 5 iterations:

Test Run	Total Time	CPU Speed	Avg Latency
1	10.0002	1388	7.20
2	10.0065	1397	7.16
3	10.0032	1374	7.05
4	10.0064	1355	7.38
5	10.0029	1382	7.23
Minimum	10.0002	1355	7.05
Maximum	10.0065	1397	7.38
Average	10.0404	1381.2	7.204
Standard Deviation	0.018807	7.823809	0.033927

iii) Experiment -1 sysbench cpu --threads=1 --cpu-max-prime=350000 --time=10
run

```

root@3f5a2e03e760:/# sysbench cpu --threads=1 --cpu-max-prime=350000 --time=10 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: 350000
Initializing worker threads...

Threads started!

CPU speed:
events per second:      87.27

General statistics:
total time:              10.0025s
total number of events:  873

Latency (ms):
min:                     11.10
avg:                     11.45
max:                     71.26
95th percentile:        12.08
sum:                     9995.91

Threads fairness:
events (avg/stddev):     873.0000/0.00
execution time (avg/stddev): 9.9959/0.00

```

Test Run	Total Time	CPU Speed	Avg Latency
1	10.0025	873	11.45
2	10.0109	888	11.26
3	10.0111	885	11.30
4	10.0074	885	11.30
5	10.0067	886	11.29
Minimum	10.0025	873	11.26
Maximum	10.0109	888	11.30
Average	10.0086	884	11.31
Standard Deviation	0.005019	3.567	0.092

Conclusion: CPU speed decreases as we increase cpu max prime.

b) QEMU Ubuntu VM

i) Experiment -1 sysbench cpu --threads=1 --cpu-max-prime=50000 --time=10 run

```

QEMU
Threads fairness:
  events (avg/stddev):       12233.0000/0.00
  execution time (avg/stddev): 9.9750/0.00

manasibendale@manasibendale:~$ sysbench cpu --threads=1 --cpu-max-prime=50000 --time=10 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: 50000
Initializing worker threads...

Threads started!

CPU speed:
  events per second: 1218.67

General statistics:
  total time:          10.0009s
  total number of events: 12189

Latency (ms):
  min:                 0.79
  avg:                 0.82
  max:                 2.53
  95th percentile:    0.86
  sum:                9977.74

Threads fairness:
  events (avg/stddev):       12189.0000/0.00
  execution time (avg/stddev): 9.9777/0.00

manasibendale@manasibendale:~$ _

```

After 5 iteration:

Test Run	Total Time	CPU Speed	Avg Latency
1	10.0004	12233	0.82
2	10.0009	12189	0.82
3	10.0006	12237	0.82
4	10.0004	12249	0.81
5	10.0008	12244	0.81
Minimum	10.0004	12189	0.81
Maximum	10.0009	12249	0.82
Average	10.0006	12229	0.82
Standard Deviation	0.0001	20.80	0.0035

ii) Experiment -1 sysbench cpu --threads=1 --cpu-max-prime=250000 --time=10 run


```

QEMU
Threads fairness:
  events (avg/stddev):       12244.0000/0.00
  execution time (avg/stddev): 9.9786/0.00

manasibendale@manasibendale:~$ sysbench cpu --threads=1 --cpu-max-prime=250000 --time=10 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: 250000
Initializing worker threads...

Threads started!

CPU speed:
  events per second:   140.52

General statistics:
  total time:          10.0049s
  total number of events: 1406

Latency (ms):
  min:                 7.02
  avg:                 7.10
  max:                 13.15
  95th percentile:    7.17
  sum:                 9988.35

Threads fairness:
  events (avg/stddev):       1406.0000/0.00
  execution time (avg/stddev): 9.9883/0.00

manasibendale@manasibendale:~$

```

After 5 iterations:

Test Run	Total Time	CPU Speed	Avg Latency
1	10.0049	1406	7.10
2	10.0029	1406	7.10
3	10.0009	1401	7.13
4	10.0060	1402	7.13
5	10.0043	1403	7.12

iii) Experiment -1 sysbench cpu --threads=1 --cpu-max-prime=350000 --time=10 run

```

QEMU
Threads fairness:
  events (avg/stddev):      1403.0000/0.00
  execution time (avg/stddev):  9.9929/0.00

manasibendale@manasibendale:~$ sysbench cpu --threads=1 --cpu-max-prime=350000 --time=10 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: 350000
Initializing worker threads...

Threads started!

CPU speed:
  events per second:      89.27

General statistics:
  total time:              10.0025s
  total number of events:   893

Latency (ms):
  min:                     11.06
  avg:                     11.19
  max:                     24.04
  95th percentile:        11.24
  sum:                     9991.45

Threads fairness:
  events (avg/stddev):      893.0000/0.00
  execution time (avg/stddev):  9.9915/0.00

manasibendale@manasibendale:~$ _

```

After 5 iterations:

Test Run	Total Time	CPU Speed	Avg Latency
1	10.0025	893	11.19
2	10.0030	892	11.20
3	10.0028	892	11.20
4	10.0057	893	11.19
5	10.0061	893	11.19

2. FILE I/O TESTING

For File I/O testing, I will use two modes which sysbench supports, which are:

- Sequential Rewrite (seqrewr)
- Combined random read/write (rndrw)

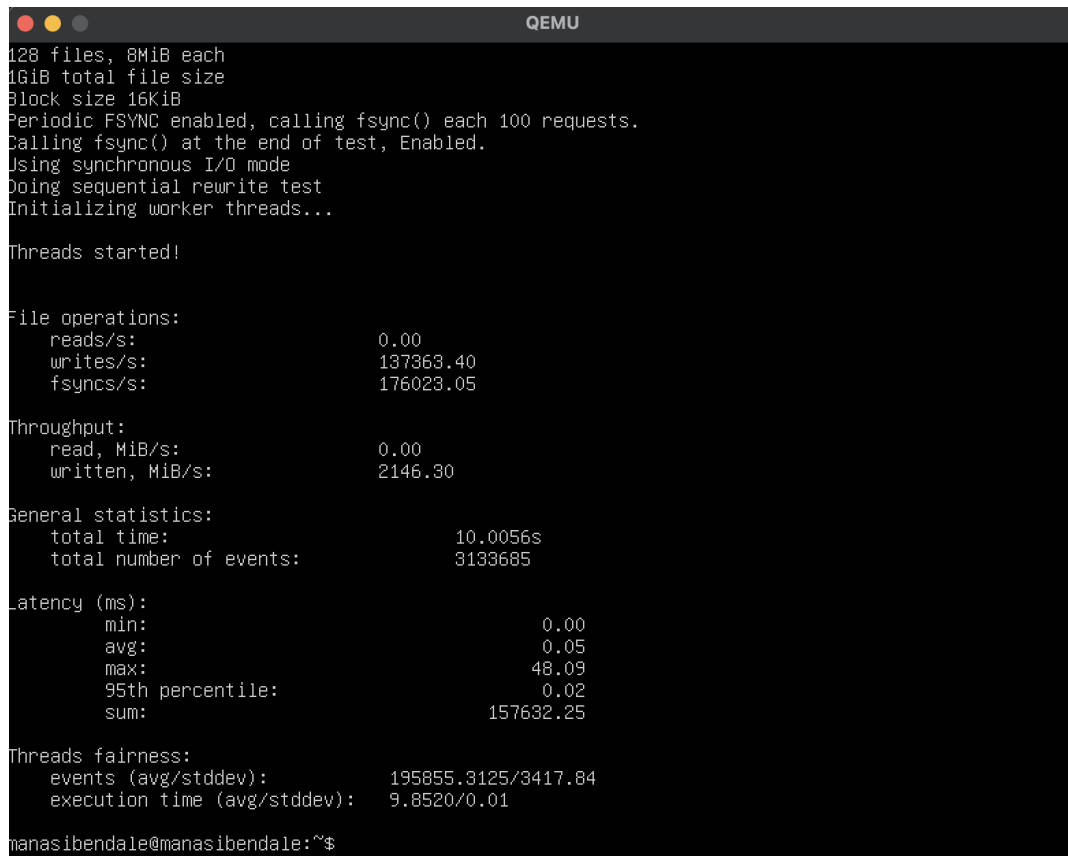
We will employ the following 3 test cases to evaluate file i/o performance between QEMU and Docker.

I. Sequential Rewrite

QEMU ubuntu VM

i) Experiment -1

**sysbench --num-threads=16 --test=fileio --file-total-size=1G
--file-test-mode=seqrewr run**



```
QEMU
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         137363.40
  fsyncs/s:         176023.05

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:    2146.30

General statistics:
  total time:       10.0056s
  total number of events: 3133685

Latency (ms):
  min:              0.00
  avg:              0.05
  max:              48.09
  95th percentile: 0.02
  sum:              157632.25

Threads fairness:
  events (avg/stddev): 195855.3125/3417.84
  execution time (avg/stddev): 9.8520/0.01
manasibendale@manasibendale:~$
```

After running for 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	0	2146.30	10.0056
2	0	2152.67	10.0632
3	0	2219.13	10.0031
4	0	2248.02	10.0071
5	0	2195.13	10.0111
Minimum	0	2146.30	10.0031
Maximum	0	2248.02	10.0632
Average	0	2203.52	10.0144
Standard Deviation	0	52.36	0.0273

ii) Experiment -2

**sysbench --num-threads=16 --test=fileio --file-total-size=3G
--file-test-mode=seqrewr run**

```

QEMU
128 files, 24MiB each
3GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         99414.70
  fsyncs/s:         127445.67

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   1553.35

General statistics:
  total time:                10.0039s
  total number of events:    2267537

Latency (ms):
  min:                      0.00
  avg:                       0.07
  max:                      37.91
  95th percentile:         0.04
  sum:                      158342.66

Threads fairness:
  events (avg/stddev):       141721.0625/3339.81
  execution time (avg/stddev): 9.8964/0.01

manasibendale@manasibendale:~$

```

After running for 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	0	1553.35	10.0039
2	0	1187.45	10.0043
3	0	1260.32	10.0033
4	0	1220.87	10.0036
5	0	1255.71	10.0055
Minimum	0	1187.45	10.0033
Maximum	0	1553.35	10.0055
Average	0	1299.13	173.47
Standard Deviation	0	10.0040	0.0009

iii) Experiment -3

**sysbench --num-threads=16 --test=fileio --file-total-size=5G
--file-test-mode=seqrewr run**

```

QEMU
128 files, 40MiB each
5GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         70849.19
  fsyncs/s:         90883.34

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:    1107.02

General statistics:
  total time:        10.0048s
  total number of events: 1616129

Latency (ms):
  min:                0.00
  avg:                 0.10
  max:                43.82
  95th percentile:    0.05
  sum:               158831.89

Threads fairness:
  events (avg/stddev): 101008.0625/3276.15
  execution time (avg/stddev): 9.9270/0.01

manasibendale@manasibendale:~$

```

After running for 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	0	1245.04	10.0041
2	0	1107.02	10.0048
3	0	1083.09	10.0095
4	0	1221.17	10.0023
5	0	1122.11	10.0030
Minimum	0	1083.09	10.0023
Maximum	0	1245.04	10.0095
Average	0	1150.65	10.0045
Standard Deviation	0	98.41	0.0027

Docker Ubuntu VM

i) Experiment -1

**sysbench --num-threads=16 --test=fileio --file-total-size=1G
--file-test-mode=seqrewr run**

```
root@3f5a2e03e760:/# sysbench --num-threads=16 --test=fileio --file-total-size=1G --time=30 --file-test-mode=seqrewr run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.15 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
Initializing random number generator from current time

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

Threads started!

File operations:
  reads/s:           0.00
  writes/s:          47619.76
  fsyncs/s:          61017.76

Throughput:
  read, MiB/s:        0.00
  written, MiB/s:      744.06

General statistics:
  total time:          30.0120s
  total number of events: 3258463

Latency (ms):
  min:                 0.00
  avg:                  0.15
  max:                  8.70
  95th percentile:     0.73
  sum:                 478817.50

Threads fairness:
  events (avg/stddev): 203653.9375/834.64
  execution time (avg/stddev): 29.9261/0.00

root@3f5a2e03e760:/# sysbench --num-threads=16 --test=fileio --file-total-size=1G --time=30 --file-test-mode=seqrewr cleanup
```

After running for 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	0	744.06	30.0120
2	0	741.98	30.0115
3	0	664.15	30.0096
4	0	717.58	30.0114
5	0	683.99	30.0161
Minimum	0	664.15	30.0096
Maximum	0	741.98	30.0161
Average	0	708.996	30.0118
Standard Deviation	0	80.5706	0.01

ii) Experiment -2

**sysbench --num-threads=16 --test=fileio --file-total-size=3G
--file-test-mode=seqrewr run**

```

root@3f5a2e03e760:/# sysbench --num-threads=16 --test=fileio --file-total-size=3G --time=30 --file-test-mode=seqrewr run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 24MiB each
3GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         33173.52
  fsyncs/s:         42529.48

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:    518.34

General statistics:
  total time:        30.0111s
  total number of events: 2269942

Latency (ms):
  min:                0.01
  avg:                0.21
  max:                21.91
  95th percentile:    0.99
  sum:               479073.63

Threads fairness:
  events (avg/stddev): 141871.3750/1069.83
  execution time (avg/stddev): 29.9421/0.00

```

After running for 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	0	518.34	30.0111
2	0	467.55	30.0158
3	0	468.22	30.0138
4	0	462.47	30.0109
5	0	573.01	30.0106
Minimum	0	462.47	30.0106
Maximum	0	573.01	30.0158
Average	0	507.218	30.0124
Standard Deviation	0	47.7456	0.01

iii) Experiment -3

```

sysbench --num-threads=16 --test=fileio --file-total-size=5G
--file-test-mode=seqrewr run

```



```
root@3f5a2e03e760:/# sysbench --num-threads=16 --test=fileio --file-total-size=5G --time=30
WARNING: the --test option is deprecated. You can pass a script name or path on the command
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 40MiB each
5GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         33318.25
  fsyncs/s:         42713.07

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:    520.60

General statistics:
  total time:       30.0097s
  total number of events: 2279696

Latency (ms):
  min:              0.01
  avg:              0.21
  max:              26.75
  95th percentile: 0.97
  sum:              479089.51

Threads fairness:
  events (avg/stddev): 142481.0000/1016.65
  execution time (avg/stddev): 29.9431/0.00
```

After 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	0	520.6	30.0097
2	0	541.92	30.0111
3	0	522.67	30.0117
4	0	542.62	30.0102
5	0	539.62	30.0147
Minimum	0	520.6	30.0097
Maximum	0	542.62	30.0147
Average	0	533.506	30.0114
Standard Deviation	0	9.0576	0.01

II. Combined random read/write

QEMU ubuntu VM

i) Experiment -1

**sysbench --num-threads=16 --test=fileio --file-total-size=1G
--file-test-mode=rndwr run**

```
QEMU
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 write test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         32089.91
  fsyncs/s:         41274.05

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:    501.40

General statistics:
  total time:        10.0056s
  total number of events: 732051

Latency (ms):
  min:                0.00
  avg:                0.22
  max:                7.70
  95th percentile:    0.95
  sum:               159672.22

Threads fairness:
  events (avg/stddev): 45753.1875/566.04
  execution time (avg/stddev): 9.9795/0.00
manasibendale@manasibendale:~$ _
```

After running for 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	0	501.40	10.0056
2	0	508.34	10.0044
3	0	510.78	10.0085
4	0	502.82	10.0070
5	0	516.17	10.0074

Minimum	0	501.40	10.0044
Maximum	0	516.17	10.0085
Average	0	508.70	10.0065
Standard Deviation	0	6.28	0.0022

ii) Experiment -1

**sysbench --num-threads=16 --test=fileio --file-total-size=3G
--file-test-mode=rndwr run**

```

QEMU
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 write test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         27581.39
  fsyncs/s:         35502.54

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   430.96

General statistics:
  total time:       10.0080s
  total number of events: 629329

Latency (ms):
  min:              0.00
  avg:              0.25
  max:              21.75
  95th percentile: 1.10
  sum:              159713.69

Threads fairness:
  events (avg/stddev): 39333.0625/633.92
  execution time (avg/stddev): 9.9821/0.00
manasibendale@manasibendale:~$

```

After running for 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	0	430.96	10.0080
2	0	416.28	10.0025
3	0	438.68	10.0152
4	0	430.73	10.0187
5	0	441.72	10.0064
Minimum	0	416.28	10.0025
Maximum	0	441.72	10.0187
Average	0	432.86	10.0101
Standard Deviation	0	7.71	0.0061

iii) Experiment -3

**sysbench --num-threads=16 --test=fileio --file-total-size=5G
--file-test-mode=rndwr run**

```

QEMU
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 write test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         27581.39
  fsyncs/s:         35502.54

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   430.96

General statistics:
  total time:       10.0080s
  total number of events: 629329

Latency (ms):
  min:              0.00
  avg:              0.25
  max:              21.75
  95th percentile: 1.10
  sum:              159713.69

Threads fairness:
  events (avg/stddev): 39333.0625/633.92
  execution time (avg/stddev): 9.9821/0.00
manasibendale@manasibendale:~$

```

After running for 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	0	447.51	10.0097
2	0	448.45	10.0112
3	0	428.72	10.0073
4	0	440.68	10.0068
5	0	462.03	10.0096
Minimum	0	428.72	10.0068
Maximum	0	462.03	10.0112
Average	0	443.77	10.0089
Standard Deviation	0	11.56	0.0019

Docker Ubuntu VM

i) Experiment -1

**sysbench --num-threads=1 --test=fileio --file-total-size=3G
--file-test-mode=rndwr run**

```

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
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:          20929.30
  writes/s:         13952.92
  fsyncs/s:         44714.62

Throughput:
  read, MiB/s:      327.02
  written, MiB/s:   218.01

General statistics:
  total time:       30.0116s
  total number of events: 2386846

Latency (ms):
  min:              0.00
  avg:              0.20
  max:              23.00
  95th percentile: 0.69
  sum:              478995.27

Threads fairness:
  events (avg/stddev): 149177.8750/590.97
  execution time (avg/stddev): 29.9372/0.00

root@3f5a2e03e760:/# sysbench --num-threads=16 --test=fileio --file-total-size=3G --time=30
up

```

After running for 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	327.02	218.01	30.0116
2	261.42	174.28	30.0153
3	344.99	229.99	30.0153
4	359.96	239.98	30.0125
5	351.82	234.55	30.0168
Minimum	261.42	174.28	30.0116
Maximum	351.82	239.98	30.0168
Average	327.072	214.242	30.0144
Standard Deviation	65.9295	23.0854	0.0103

ii) Experiment -2

**sysbench --num-threads=1 --test=fileio --file-total-size=1G
--file-test-mode=rndwr run**

```
root@3f5a2e03e760:/# sysbench --num-threads=16 --test=fileio --file-total-size=1G --time=30 --file-test-mode=rndrw run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
10iB 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:                28044.28
  writes/s:               18696.25
  fsyncs/s:               59893.26

Throughput:
  read, MiB/s:            438.19
  written, MiB/s:         292.13

General statistics:
  total time:              30.0155s
  total number of events:  3198679

Latency (ms):
  min:                     0.00
  avg:                     0.15
  max:                     33.70
  95th percentile:        0.49
  sum:                     478718.08

Threads fairness:
  events (avg/stddev):    199917.4375/1175.84
  execution time (avg/stddev): 22.9122/0.00
```

After 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	440.03	293.36	30.0118
2	438.19	292.13	30.0155
3	465.24	310.16	30.0099
4	416.50	277.67	30.0106
5	391.54	261.03	30.0105
Minimum	391.54	261.03	30.0155
Maximum	465.24	310.16	30.0099
Average	430.85	284.86	30.0113

iii) Experiment -3

**sysbench --num-threads=1 --test=fileio --file-total-size=5G
--file-test-mode=rndwr run**

```
manasibendale — root@3f5a2e03e760: / — com.docker.cli - docker run -it --entrypoint /bin/bash arm64v8/ubuntu:20.04
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 40MiB each
5GiB 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:          14356.32
  writes/s:         9570.61
  fsyncs/s:        30690.70

Throughput:
  read, MiB/s:      224.32
  written, MiB/s:   149.54

General statistics:
  total time:       30.0156s
  total number of events: 1637376

Latency (ms):
  min:             0.00
  avg:             0.29
  max:            21.94
  95th percentile: 0.97
  sum:            479294.41

Threads fairness:
  events (avg/stddev): 102336.0000/390.07
  execution time (avg/stddev): 29.9559/0.00

root@3f5a2e03e760:/# sysbench --num-threads=16 --test=fileio --file-total-size=5G --time=30 --file-test-mode=rndwr cl
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
```

After 5 iterations:

Test Run	Read Throughput	Write Throughput	Total Time
1	220.21	146.8	30.0139
2	225.10	150.06	30.0159
3	224.32	149.54	30.0156
4	222.19	148.13	30.0154
5	272.92	181.94	30.0149
Minimum	222.19	146.8	30.0139
Maximum	272.92	150.06	30.0159
Average	220.57	152.33	30.0153
Standard Deviation	26.49	17.44	0.0009

Conclusion:

MANASI BENDALE (W1630617)

Comparing both QEMU and Docker, both perform similarly for all the 3 test cases. This can be a result of the system configurations of the machines. In terms of file i/o QEMU performed better than docker.

Github Repository

Account Name: ManasiBendale

Repository - Cloud_Computing

Folder for assignment 1: Assignment_1

Link: https://github.com/ManasiBendale/Cloud_Computing