COEN 241 HomeWork 1 (1609266)

Configuration of Experiment:

I conducted the experiment using Qemu and Docker. I'm using MacOs with the M1 processor. For setting up the VMs I used Qemu and Docker. The configuration of the VMs I set up for this experiment is:

Docker:
1. 1 GB Ram 1 CPU
2. 2 GB Ram 2 CPU
3. 4 GB Ram 4 CPU

Qemu:
1. 1GB Ram 1 CPU
2. 2GB Ram 2 CPU
3. 3GB Ram 3 CPU

Qemu Installation:
1. Install qemu using: "brew install qemu"
2. Create empty image using: "$ qemu-img create -f raw ~/qemu/ubuntu-latest.raw 40G"
3. Download prebuilt EDK2 UEFI image form "https://gist.github.com/theboreddev/5f79f86a0f163e4a1f9df919da5eea20#:~:text=QEMU_EFI%2Dcb438b9%2Dedk2%2Dstable202011%2Dwith%2Dextra%2Dresolutions.tar.gz"
4. Decompress the content and copy QEMU_EFI.fd QEMU_VARS.fd to your qemu directory.
5. Download ubuntu iso image and install it using:

```
qemu-system-aarch64 \
    -serial 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-22.04.1-live-server-arm64.iso

Docker:
1. Download docker from "https://www.docker.com/" and install it.
2.

```
(base) shubhamkamdi@Shubhams-MacBook-Pro-2 ~ % docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
8b150fd943bc: Pull complete
Digest: sha256:9a0bdde4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
(base) shubhamkamdi@Shubhams-MacBook-Pro-2 ~ %
```

   Use "docker pull ubuntu" to download ubuntu.
3. Next run "docker -run -t -i ubuntu /bin/bash" this will start bash terminal for you.
4.

```
(base) shubhamkamdi@Shubhams-MacBook-Pro-2 ~ % docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED          STATUS            PORTS     NAMES
bcd4d3a216d4   ubuntu         "/bin/bash"   About a minute ago   Up About a minute           silly_wescoff
5408326dd507   c39b4f7af9a0   "bash"        2 days ago        Up 2 days                   charming_goldstine
```

   On another terminal run docker ps to see the running docker container. Here we will commit the docker container.

5. I have already committed my container and so I run "docker run -it –memory="1G" –cpus="1.0" my_ubuntu ". This command specifies the memory, cpus of your vm. My_ubuntu in the command is the commit message for the container.

6. Next install vim editor on your ubuntu instance, we will need to use and editor to type our script.

Screenshots of the instances are on separate files, please refer to them.

Experiment conducting:
We need to install sysbench as well, so run this command, "apt-install sysbench"
Let first start with Qemu, we first conduct the CPU test using the command "sysbench cpu --cpu-max-prime=5000 --num-threads=1 --time=35 run"

I ran a bash script for this command in three scenarios:
1. Cpu_max_prime = 5000 and time = 35s
2. Cpu_max_prime = 10000 and time = 30s
3. Cpu_max_prime = 150000 and time = 50s

All of the above cases were executed 5 times per VM configuration(as mentioned above)
Results:

Config	Max	Min	Avg
1 GB Ram 1 CPU + Test 1 (5 Runs)	21053.64	20978	21019.54
1 GB Ram 1 CPU + Test 2 (5 Runs)	8653.49	8607.62	8629.39
1 GB Ram 1 CPU + Test 3 (5 Runs)	268.03	267.3	267.75
2 GB Ram 2 CPU + Test 1 (5 Runs)	21072.49	20945.99	20985.972
2 GB Ram 2 CPU + Test 2 (5 Runs)	8661.71	8645.18	8654.02
2 GB Ram 2 CPU + Test 3 (5 Runs)	267.95	267.58	267.81
3 GB Ram 4 CPU + Test 1 (5 Runs)	21175.08	20966.21	21046.39
3 GB Ram 4 CPU + Test 2 (5 Runs)	8626.86	8613.17	8626.004
3 GB Ram 4 CPU + Test 3 (5 Runs)	267.85	267.03	267.412

These are the results from qemu VM. The screenshots of these are in the QemuResult file in the repository. Please refer to them. I kept them separate in order to keep the size of the report short.

Now I conducted the same tests on Docker and its results are noted down:

Configuration	Max	min	Average
1 GB Ram 1 CPU + Test 1 (5 Runs)	21094.47	21069.15	21081.1
1 GB Ram 1 CPU + Test 2 (5 Runs)	8679.34	8662	8671.792
1 GB Ram 1 CPU + Test 3 (5 Runs)	267.88	267.3	267.746
2 GB Ram 2 CPU + Test 1 (5 Runs)	21112.29	21087.93	21102.84
2 GB Ram 2 CPU + Test 2 (5 Runs)	8690.12	8675.06	8684.028
2 GB Ram 2 CPU + Test 2 (5 Runs)	268.18	268.08	268.144
4GB Ram 4 CPU + Test 1 (5 Runs)	21097.5	20993.97	21038.54
4GB Ram 4 CPU + Test 2 (5 Runs)	8660.86	8471.08	8555.97
4GB Ram 4 CPU + Test 3 (5 Runs)	266.78	265.07	265.912

Shell Script:

CPU Test:

```bash
for i in {1..5}
do
        echo "Config 1, iteration $i"
        echo "CPU = 5000", Thread = 1, Time = 35
        sysbench cpu --cpu-max-prime=5000  --num-threads=1 --time=35 run
done

for j in {1..5}
do
    echo "Config 1, iteration $j"
    echo "CPU = 10000", Thread = 1, Time = 30
    sysbench cpu --cpu-max-prime=10000  --num-threads=1 --time=30 run
done

for k in {1..5}
do
    echo "Config 1, iteration $k"
    echo "CPU = 150000", Thread = 1, Time = 50
    sysbench cpu --cpu-max-prime=150000  --num-threads=1 --time=50 run

done
```

FileIO Test:

```sh
#!/bin/sh

#  fileio.sh

#

#

#  Created by Shubham Kamdi on 2/2/23.

#

file_size=3G


# Prepare the test file

sysbench fileio --file-total-size=$file_size prepare


# Run the read test
```

```
        sysbench fileio --file-total-size=$file_size --file-test-mode=rndrw
        --time=30 --threads=1 run



        # Clean up the test file

        sysbench fileio --file-total-size=$file_size cleanup
```

Conclusion: There is no significant change in CPU performance after adding additional resources as well. Similar performance is observed on both Docker and qemu.

Sample Outputs:

CPU Test:

```
                        shubhamkamdi — root@5408326dd507: / — com.docker.cli ‹ sudo — 115×40
        sum:                                49994.94

Threads fairness:
    events (avg/stddev):           13309.0000/0.00
    execution time (avg/stddev):   49.9949/0.00

Config 3, iteration 5
CPU = 150000, Thread = 1, Time = 50
sysbench 1.0.20 (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: 150000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:    265.07

General statistics:
    total time:                    50.0009s
    total number of events:        13254

Latency (ms):
        min:                               3.71
        avg:                               3.77
        max:                               8.85
        95th percentile:                   3.89
        sum:                           49993.97

Threads fairness:
    events (avg/stddev):           13254.0000/0.00
    execution time (avg/stddev):   49.9940/0.00
```

FileIO test:

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


Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      12741.50
    writes/s:                     8494.33
    fsyncs/s:                     27184.21

Throughput:
    read, MiB/s:                  199.09
    written, MiB/s:               132.72

General statistics:
    total time:                        30.0075s
    total number of events:            1452861

Latency (ms):
         min:                                  0.00
         avg:                                  0.02
         max:                                 11.19
         95th percentile:                      0.08
         sum:                              29658.86

Threads fairness:
    events (avg/stddev):           1452861.0000/0.00
    execution time (avg/stddev):   29.6589/0.00

sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)
```

Please refer to the screenshot files for all the screenshots.