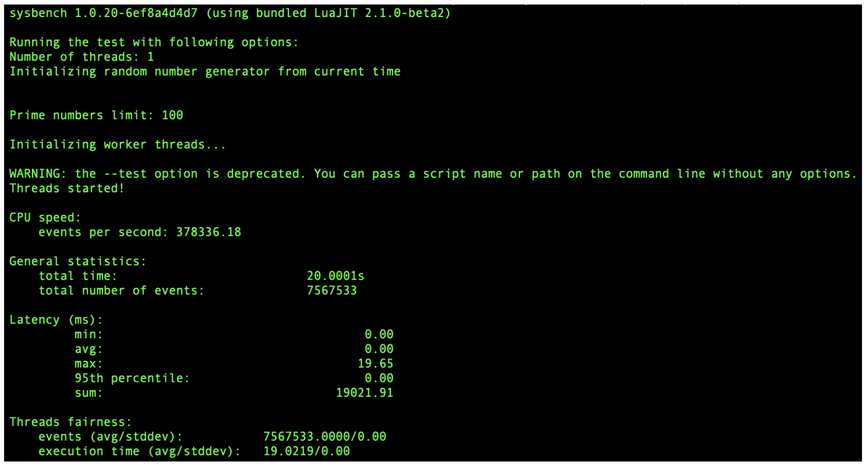
System Configuration
System: MacBook Pro
OS: macOS Catalina
Version: 10.15.7
Chip: 3.3 GHz Dual-Core Intel Core i5
Memory: 16 GB

Setup

After updating homebrew, I ran *brew install qemu* and after a few minutes, it completed successfully!!! I proceeded to install docker and sysbench with no issue. To verify that I had done everything correctly I ran the example command *docker run --rm*



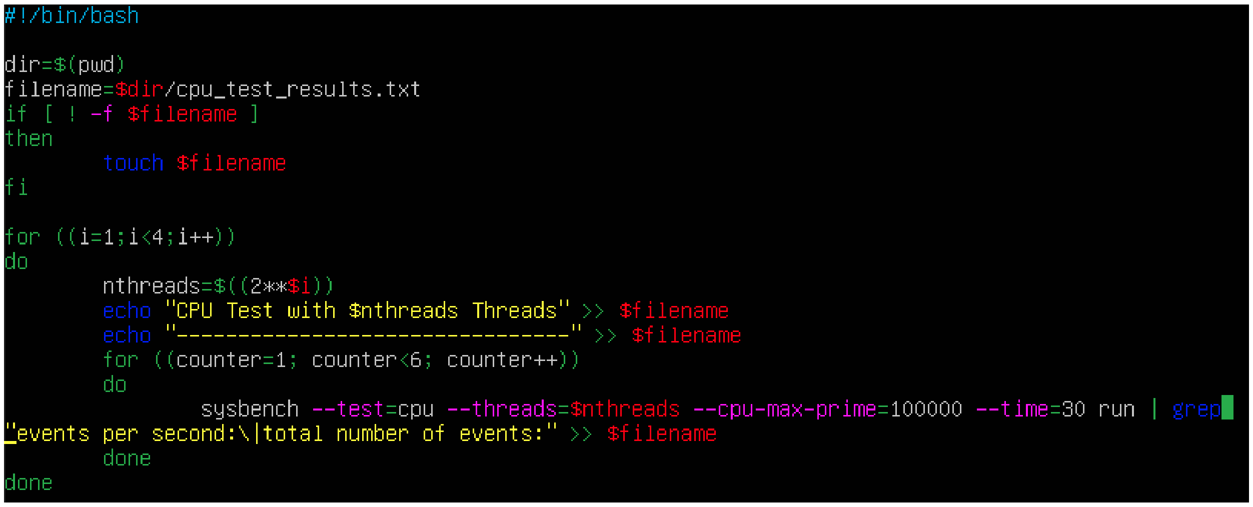
```
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

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


Prime numbers limit: 100

Initializing worker threads...

WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Threads started!

CPU speed:
    events per second: 378336.18

General statistics:
    total time:                          20.0001s
    total number of events:              7567533

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   19.65
         95th percentile:                        0.00
         sum:                                19021.91

Threads fairness:
    events (avg/stddev):           7567533.0000/0.00
    execution time (avg/stddev):   19.0219/0.00
```

zyclonite/sysbench --test=cpu --cpu-max-prime=100 --time=20 run with the following output:

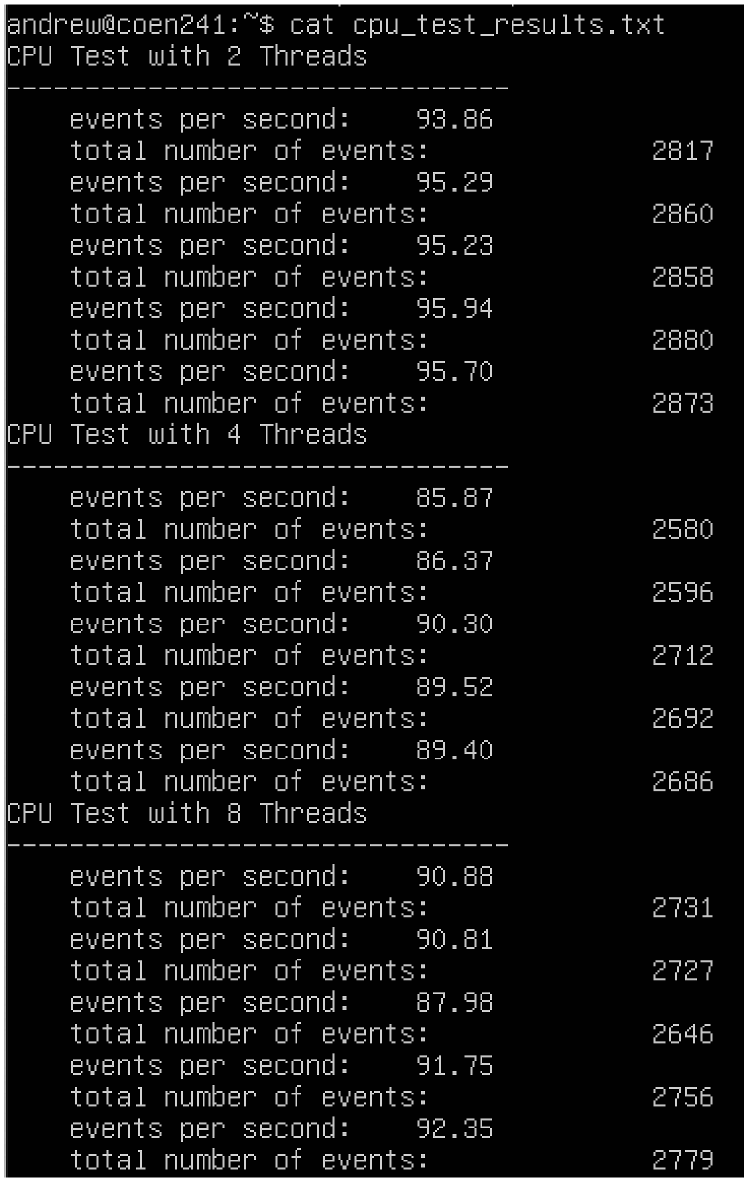
Qemu VM Performance Testing

With everything installed and working all that's left is to spin up the VM. This can be easily accomplished with the command *sudo qemu-system-x86_64 -m 4096 -boot d -accel hvf -cpu host -smp cores=2,threads=2 -boot strict=on -hda ubuntu.img*. Using the m flag I allocated 4 GB of RAM. I chose to use HVM (Hardware Virtual Machine) as the accelerator because KVM does not appear to work with Mac. With the smp flag I allocated 2 cores with a total of 4 threads.

The first tests I decided to run were cpu tests. To do this I created a bash script to run a total of 15 tests with 5 tests each for 2 threads, 4 threads and 8 threads. Each of these tests used –cpu-max-prime=100000 and –time=30. The script and test results can be seen in the two figures below.

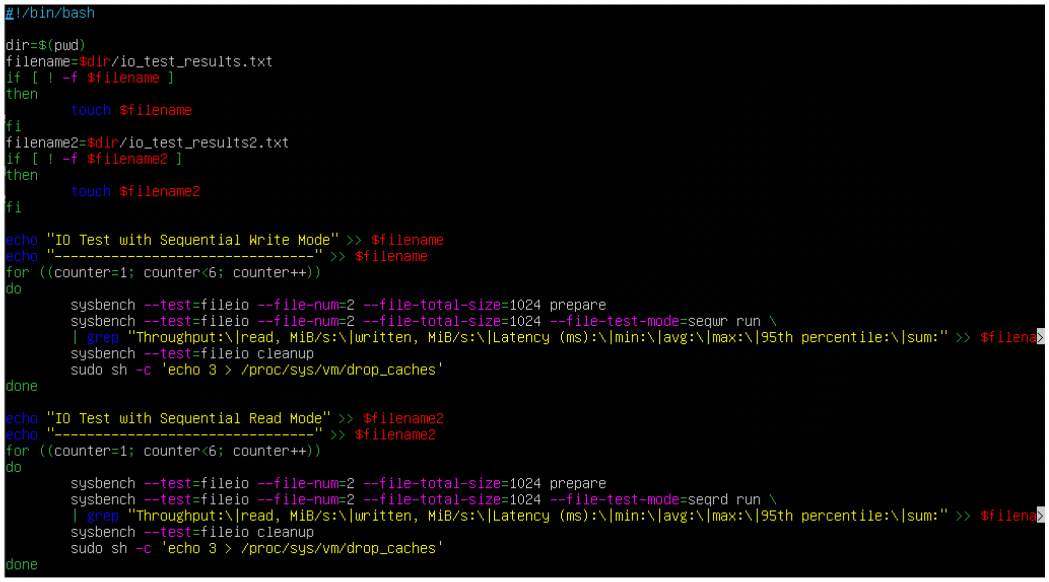


```
#!/bin/bash

dir=$(pwd)
filename=$dir/cpu_test_results.txt
if [ ! -f $filename ]
then
        touch $filename
fi

for ((i=1;i<4;i++))
do
        nthreads=$((2**$i))
        echo "CPU Test with $nthreads Threads" >> $filename
        echo "--------------------------------" >> $filename
        for ((counter=1; counter<6; counter++))
        do
                sysbench --test=cpu --threads=$nthreads --cpu-max-prime=100000 --time=30 run | grep
"events per second:\|total number of events:" >> $filename
        done
done
```

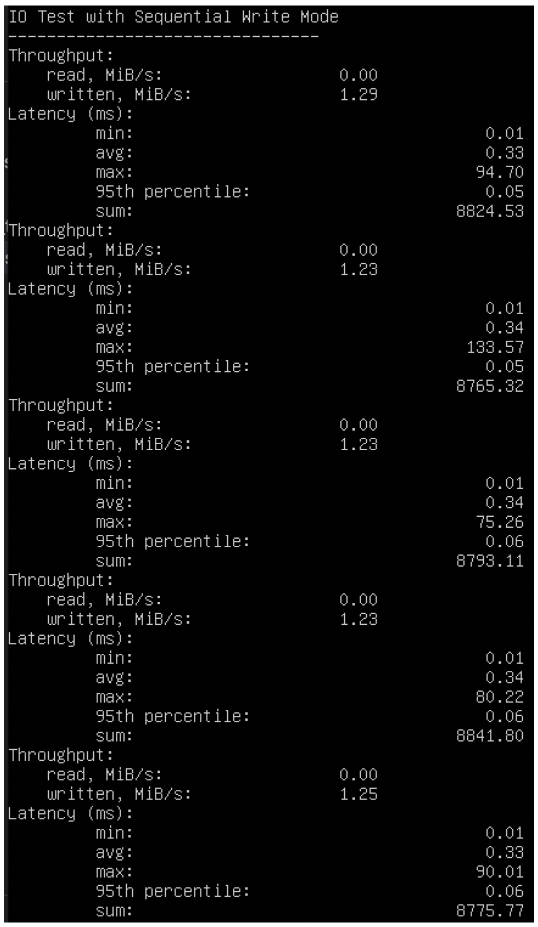


```
andrew@coen241:~$ cat cpu_test_results.txt
CPU Test with 2 Threads
--------------------------------
    events per second:     93.86
    total number of events:              2817
    events per second:     95.29
    total number of events:              2860
    events per second:     95.23
    total number of events:              2858
    events per second:     95.94
    total number of events:              2880
    events per second:     95.70
    total number of events:              2873
CPU Test with 4 Threads
--------------------------------
    events per second:     85.87
    total number of events:              2580
    events per second:     86.37
    total number of events:              2596
    events per second:     90.30
    total number of events:              2712
    events per second:     89.52
    total number of events:              2692
    events per second:     89.40
    total number of events:              2686
CPU Test with 8 Threads
--------------------------------
    events per second:     90.88
    total number of events:              2731
    events per second:     90.81
    total number of events:              2727
    events per second:     87.98
    total number of events:              2646
    events per second:     91.75
    total number of events:              2756
    events per second:     92.35
    total number of events:              2779
```

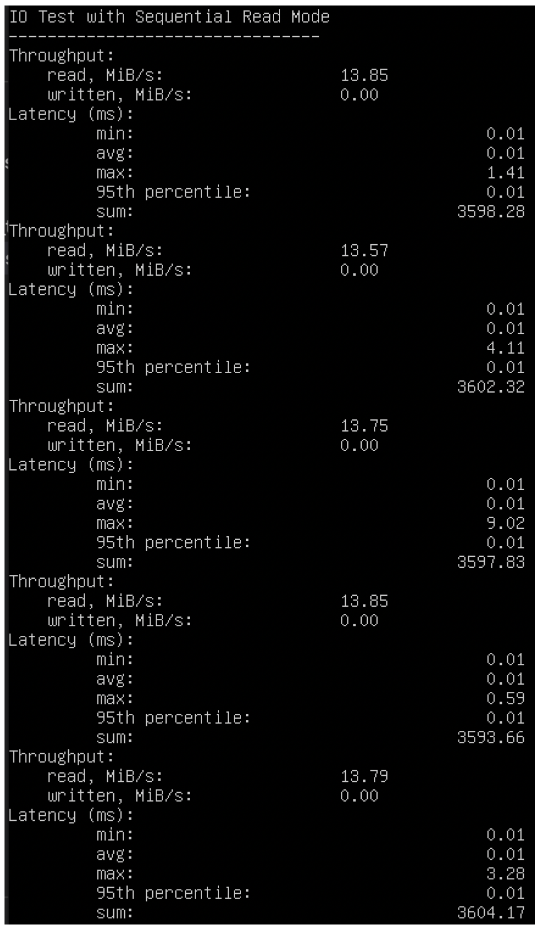
Next I ran IO tests. I created a new script to run 5 tests using sequential write mode, and 5 tests using sequential read mode. I ran these tests with a varying number of files and total file sizes, to varying degrees of success. I settled on using just 2 files of size 1024 due to storage and memory limitations on my computer.



```
#!/bin/bash

dir=$(pwd)
filename=$dir/io_test_results.txt
if [ ! -f $filename ]
then
        touch $filename
fi
filename2=$dir/io_test_results2.txt
if [ ! -f $filename2 ]
then
        touch $filename2
fi

echo "IO Test with Sequential Write Mode" >> $filename
echo "--------------------------------" >> $filename
for ((counter=1; counter<6; counter++))
do
        sysbench --test=fileio --file-num=2 --file-total-size=1024 prepare
        sysbench --test=fileio --file-num=2 --file-total-size=1024 --file-test-mode=seqwr run \
        | grep "Throughput:\|read, MiB/s:\|written, MiB/s:\|Latency (ms):\|min:\|avg:\|max:\|95th percentile:\|sum:" >> $filena>
        sysbench --test=fileio cleanup
        sudo sh -c 'echo 3 > /proc/sys/vm/drop_caches'
done

echo "IO Test with Sequential Read Mode" >> $filename2
echo "--------------------------------" >> $filename2
for ((counter=1; counter<6; counter++))
do
        sysbench --test=fileio --file-num=2 --file-total-size=1024 prepare
        sysbench --test=fileio --file-num=2 --file-total-size=1024 --file-test-mode=seqrd run \
        | grep "Throughput:\|read, MiB/s:\|written, MiB/s:\|Latency (ms):\|min:\|avg:\|max:\|95th percentile:\|sum:" >> $filena>
        sysbench --test=fileio cleanup
        sudo sh -c 'echo 3 > /proc/sys/vm/drop_caches'
done
```



```
IO Test with Sequential Write Mode
--------------------------------
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.29
Latency (ms):
         min:                                   0.01
         avg:                                   0.33
         max:                                  94.70
         95th percentile:                       0.05
         sum:                                8824.53
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.23
Latency (ms):
         min:                                   0.01
         avg:                                   0.34
         max:                                 133.57
         95th percentile:                       0.05
         sum:                                8765.32
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.23
Latency (ms):
         min:                                   0.01
         avg:                                   0.34
         max:                                  75.26
         95th percentile:                       0.06
         sum:                                8793.11
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.23
Latency (ms):
         min:                                   0.01
         avg:                                   0.34
         max:                                  80.22
         95th percentile:                       0.06
         sum:                                8841.80
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.25
Latency (ms):
         min:                                   0.01
         avg:                                   0.33
         max:                                  90.01
         95th percentile:                       0.06
         sum:                                8775.77
```

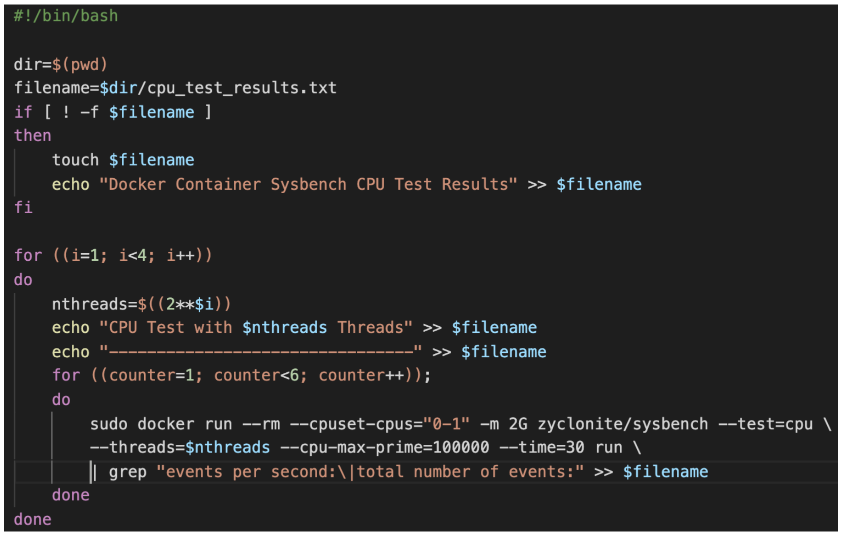


```
IO Test with Sequential Read Mode
--------------------------------
Throughput:
    read, MiB/s:                  13.85
    written, MiB/s:               0.00
Latency (ms):
         min:                                   0.01
         avg:                                   0.01
         max:                                   1.41
         95th percentile:                       0.01
         sum:                                3598.28
Throughput:
    read, MiB/s:                  13.57
    written, MiB/s:               0.00
Latency (ms):
         min:                                   0.01
         avg:                                   0.01
         max:                                   4.11
         95th percentile:                       0.01
         sum:                                3602.32
Throughput:
    read, MiB/s:                  13.75
    written, MiB/s:               0.00
Latency (ms):
         min:                                   0.01
         avg:                                   0.01
         max:                                   9.02
         95th percentile:                       0.01
         sum:                                3597.83
Throughput:
    read, MiB/s:                  13.85
    written, MiB/s:               0.00
Latency (ms):
         min:                                   0.01
         avg:                                   0.01
         max:                                   0.59
         95th percentile:                       0.01
         sum:                                3593.66
Throughput:
    read, MiB/s:                  13.79
    written, MiB/s:               0.00
Latency (ms):
         min:                                   0.01
         avg:                                   0.01
         max:                                   3.28
         95th percentile:                       0.01
         sum:                                3604.17
```

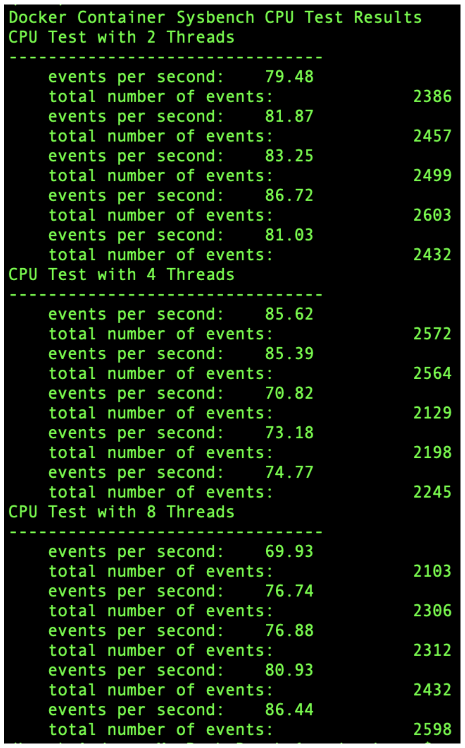
Docker Container Performance Testing

Spinning up a docker container is easy once it has been installed. We can run sudo docker run --rm -it --cpuset-cpus="0-1" -m 2G --entrypoint /bin/sh zyclonite/sysbench. This will start a container using the zyclonite/sysbench image, and allocate a single cpu core with 2 threads and 2 gigabytes of ram. The -rm flag ensures that the container is removed when the shell is closed. The -it flag specifies it being interactive with shell type being specified by the -entrypoint flag.

Similar to the qemu tests, I started out with the cpu tests. To test I used the following command: sudo docker run --rm --cpuset-cpus="0-1" -m 2G zyclonite/sysbench --test=cpu --threads=2 --cpu-max-prime=100000 --time=30 run | grep "events per second:\|total number of events:" >> $filename. Again I ran tests using 2, 4 and 8 threads, and ran each test 5 times. The script used and the output are shown below.

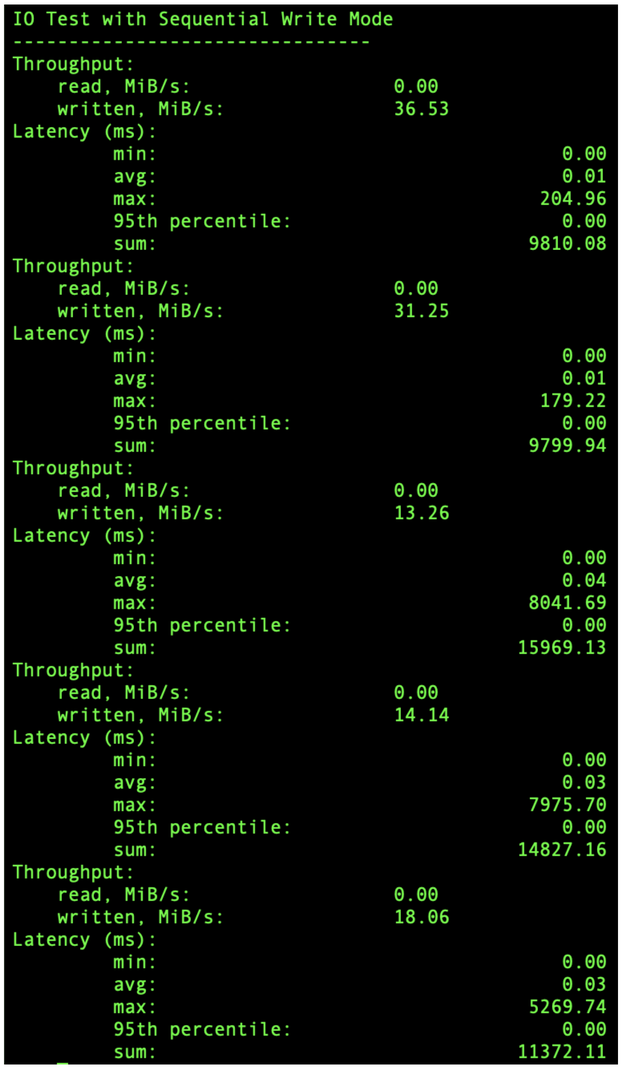


```
#!/bin/bash

dir=$(pwd)
filename=$dir/cpu_test_results.txt
if [ ! -f $filename ]
then
    touch $filename
    echo "Docker Container Sysbench CPU Test Results" >> $filename
fi

for ((i=1; i<4; i++))
do
    nthreads=$((2**$i))
    echo "CPU Test with $nthreads Threads" >> $filename
    echo "--------------------------------" >> $filename
    for ((counter=1; counter<6; counter++));
    do
        sudo docker run --rm --cpuset-cpus="0-1" -m 2G zyclonite/sysbench --test=cpu \
        --threads=$nthreads --cpu-max-prime=100000 --time=30 run \
        | grep "events per second:\|total number of events:" >> $filename
    done
done
```

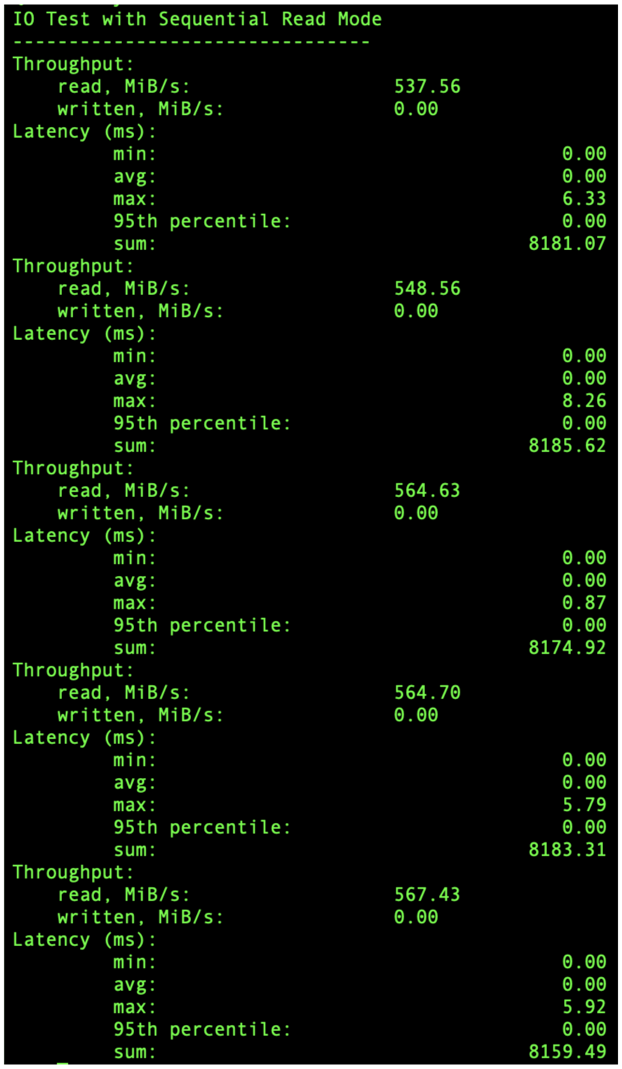


```
Docker Container Sysbench CPU Test Results
CPU Test with 2 Threads
--------------------------------
    events per second:    79.48
    total number of events:              2386
    events per second:    81.87
    total number of events:              2457
    events per second:    83.25
    total number of events:              2499
    events per second:    86.72
    total number of events:              2603
    events per second:    81.03
    total number of events:              2432
CPU Test with 4 Threads
--------------------------------
    events per second:    85.62
    total number of events:              2572
    events per second:    85.39
    total number of events:              2564
    events per second:    70.82
    total number of events:              2129
    events per second:    73.18
    total number of events:              2198
    events per second:    74.77
    total number of events:              2245
CPU Test with 8 Threads
--------------------------------
    events per second:    69.93
    total number of events:              2103
    events per second:    76.74
    total number of events:              2306
    events per second:    76.88
    total number of events:              2312
    events per second:    80.93
    total number of events:              2432
    events per second:    86.44
    total number of events:              2598
```

Next I run io tests. I followed the same conditions as I did with the qemu tests. To run the tests, I opened a docker container with Docker run --rm -it -m=2G --cpuset-cpus="0-1" --entrypoint /bin/sh zyclonite/sysbench. I then ported over the same script as used in the qemu test. The results of the test are shown below.



```
IO Test with Sequential Write Mode
--------------------------------
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               36.53
Latency (ms):
         min:                                  0.00
         avg:                                  0.01
         max:                                204.96
         95th percentile:                      0.00
         sum:                               9810.08
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               31.25
Latency (ms):
         min:                                  0.00
         avg:                                  0.01
         max:                                179.22
         95th percentile:                      0.00
         sum:                               9799.94
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               13.26
Latency (ms):
         min:                                  0.00
         avg:                                  0.04
         max:                               8041.69
         95th percentile:                      0.00
         sum:                              15969.13
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               14.14
Latency (ms):
         min:                                  0.00
         avg:                                  0.03
         max:                               7975.70
         95th percentile:                      0.00
         sum:                              14827.16
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               18.06
Latency (ms):
         min:                                  0.00
         avg:                                  0.03
         max:                               5269.74
         95th percentile:                      0.00
         sum:                              11372.11
```



```
IO Test with Sequential Read Mode
--------------------------------
Throughput:
    read, MiB/s:                  537.56
    written, MiB/s:               0.00
Latency (ms):
         min:                                  0.00
         avg:                                  0.00
         max:                                  6.33
         95th percentile:                      0.00
         sum:                               8181.07
Throughput:
    read, MiB/s:                  548.56
    written, MiB/s:               0.00
Latency (ms):
         min:                                  0.00
         avg:                                  0.00
         max:                                  8.26
         95th percentile:                      0.00
         sum:                               8185.62
Throughput:
    read, MiB/s:                  564.63
    written, MiB/s:               0.00
Latency (ms):
         min:                                  0.00
         avg:                                  0.00
         max:                                  0.87
         95th percentile:                      0.00
         sum:                               8174.92
Throughput:
    read, MiB/s:                  564.70
    written, MiB/s:               0.00
Latency (ms):
         min:                                  0.00
         avg:                                  0.00
         max:                                  5.79
         95th percentile:                      0.00
         sum:                               8183.31
Throughput:
    read, MiB/s:                  567.43
    written, MiB/s:               0.00
Latency (ms):
         min:                                  0.00
         avg:                                  0.00
         max:                                  5.92
         95th percentile:                      0.00
         sum:                               8159.49
```

QEMU vs Docker Test Result Comparison

CPU Test with 2 Threads				
	QEMU		Docker	
	Events Per Second	Total Number of Events	Events Per Second	Total Number of Events
Average	95.204	2857.6	82.47	2475.4
Min	93.86	2817	79.48	2432
Max	95.94	2880	86.72	2603

CPU Test with 4 Threads				
	QEMU		Docker	
	Events Per Second	Total Number of Events	Events Per Second	Total Number of Events
Average	88.292	2653.2	77.956	2341.6
Min	85.87	2580	70.82	2129
Max	90.30	2712	85.62	2572

CPU Test with 8 Threads				
	QEMU		Docker	
	Events Per Second	Total Number of Events	Events Per Second	Total Number of Events
Average	90.754	2727.8	78.184	2350.2
Min	87.98	2646	69.93	2103
Max	92.35	2779	86.44	2598

IO Test with Sequential Write Mode				
	Qemu		Docker	
	Written MiB/s	Latency avg (ms)	Written MiB/s	Latency avg (ms)
Average	1.246	.336	22.648	.024
Min	1.23	.33	13.26	.01
Max	1.29	.34	36.53	.04

IO Test with Sequential Read Mode				
	Qemu		Docker	
	Read MiB/s	Latency avg (ms)	Read MiB/s	Latency avg (ms)
Average	13.762	0.01	556.576	0.0
Min	13.57	0.01	537.56	0.0
Max	13.85	0.01	567.43	0.0

From these results we can tell the QEMU performed better in each of the CPU tests for both events per second and total number of events. However, when looking at the results from IO tests, we can see that Docker performed for both reading and writing.