

Configurations of experimental setup:

Docker	CPU	cpus=1, m=16g	cpus=2, m=16g	cpus=2, m=32g
	File I/O	file-num=32	file-num=64	file-num=128
QEUM	CPU	cpus=1, m=16g	cpus=2, m=16g	cpus=2, m=32g
	File I/O	file-num=32	file-num=64	file-num=128

Main steps to enable a QEMU VM (with complex commands and VM configuration):



```
sandyhsueh@sandy-s-MacBook-Pro ~ % ls
Applications          Pictures          opt
Desktop               Public            package-lock.json
Documents             anaconda3        package.json
Downloads             bin               practice
IdeaProjects          crwn-clothing   python.py
Library               homebrew         src
Movies                lab1-checkoff  ubuntu_vm
Music                 learngit        ubuntu_vm
Oracle                node_modules
sandyhsueh@sandy-s-MacBook-Pro ~ % cd ubuntu_vm
sandyhsueh@sandy-s-MacBook-Pro ubuntu_vm % ls
read.txt              ubuntu-20.04.4-live-server-amd64.iso
start.sh              ubuntu_drive.qcow2
sandyhsueh@sandy-s-MacBook-Pro ubuntu_vm % ./start.sh

Ubuntu 20.04.4 LTS 241hw tty1
241hw login: sandyhsueh
Password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Sun 17 Apr 2022 04:07:26 AM UTC

 System load:          0.67
 Usage of /:           37.5% of 8.90GB
 Memory usage:         7%
 Swap usage:           0%
 Processes:            135
 Users logged in:     0
 IPv4 address for docker0: 172.17.0.1
 IPv4 address for enp0s2: 10.0.2.15
 IPv6 address for enp0s2: fec0::5054:ff:fe12:3456

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Thu Apr 14 05:33:48 UTC 2022 on tty1
sandyhsueh@241hw:~$
```

Main steps to enable Docker container (the operation I used):

Open the docker desktop, and we can do all the commands starting as Docker.

Proof of experiment (pics of Docker and VM running environments for each experimentation):

A. CPU (Docker)

CPU = 1, Memory = 16G, threads= 1, max-prime = 10000, time = 30s

```
sandyhsueh@sandy-s-MacBook-Pro ~ % docker run -it --cpus=1 -m 16G cyclonite/sysbench --test.cpu --cpu-max-prime=10000 --time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20-6ef8a4d4d7 (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: 10000
Initializing worker threads...

Threads started!

CPU speed:
  events per second: 2638.55

General statistics:
  total time:          30.0003s
  total number of events: 79162

Latency (ms):
  min:                 0.33
  avg:                 0.38
  max:                14.33
  95th percentile:    0.46
  sum:               29909.30

Threads fairness:
  events (avg/stddev): 79162.0000/0.00
  execution time (avg/stddev): 29.9093/0.00
```

Docker	CPU speed (events/s)	CPU utilization
1	2614.50	System:16.14% / User: 4.68%
2	2555.27	System:14.89% / User: 2.75%
3	2541.98	System:16.27% / User: 6.63%
4	2595.24	System:15.14% / User: 3.22%
5	2582.35	System:17.03% / User: 5.74%
avg	2577.87	System:15.89% / User: 4.60%

CPU = 2, Memory = 16G, threads= 1, max-prime = 10000, time = 30s

```
sandyhsueh@sandy-s-MacBook-Pro ~ % docker run -it --cpus=2 -m 16G zyclonite/sysbench --test=cpu --cpu-max-prime=10000 --time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20-6ef8a4d4d7 (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: 10000
Initializing worker threads...

Threads started!

CPU speed:
events per second: 2602.64

General statistics:
total time: 30.0005s
total number of events: 78085

Latency (ms):
min: 0.33
avg: 0.38
max: 8.91
95th percentile: 0.48
sum: 29890.46

Threads fairness:
events (avg/stddev): 78085.0000/0.00
execution time (avg/stddev): 29.8905/0.00
```

Docker	CPU speed (events/s)	CPU utilization
1	2653.50	System:15.61% / User: 4.62%
2	2540.85	System:18.54% / User: 6.16%
3	2593.78	System:17.97% / User: 7.87%
4	2585.33	System:16.00% / User: 4.85%
5	2578.96	System:17.03% / User: 5.74%
avg	2590.48	System:17.03% / User: 5.85%

CPU = 2, Memory = 32G, threads= 1, max-prime = 10000, time = 30s

```
sandyhsueh@sandy-s-MacBook-Pro ~ % docker run -it --cpus=2 -m 32G zyclonite/sysbench --test=cpu --cpu-max-prime=10000 --time=30 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20-6ef8a4d4d7 (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: 10000
Initializing worker threads...

Threads started!

CPU speed:
events per second: 2579.10

General statistics:
total time: 30.0002s
total number of events: 77378

Latency (ms):
min: 0.32
avg: 0.39
max: 8.13
95th percentile: 0.49
sum: 29878.84

Threads fairness:
events (avg/stddev): 77378.0000/0.00
execution time (avg/stddev): 29.8788/0.00
```

Docker	CPU speed (events/s)	CPU utilization
1	2638.90	System:16.5% / User: 7.35%
2	2639.80	System:15.21% / User: 5.89%
3	2699.59	System:15.53% / User: 5.54%
4	2602.69	System:14.80% / User: 3.58%

5	2673.34	System:14.79% / User: 3.32%
avg	2650.86	System:15.37% / User: 5.14%

B. File O/I (Docker)

Threads = 1, file-num = 32, time = 30s

```
/ # sysbench fileio --file-num=32 prepare
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

32 files, 65536Kb each, 2048Mb total

/ # sysbench fileio --file-num=32 --file-test-mode=seqrewr --time=30 run
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

Extra file open flags: (none)
32 files, 64MiB each
2GiB 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:        17278.80
  fsyncs/s:        5530.25

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   269.98

General statistics:
  total time:           30.0063s
  total number of events: 684419

Latency (ms):
  min:                0.00
  avg:                0.04
  max:              2611.50
  95th percentile:    0.09
  sum:              29731.98

Threads fairness:
  events (avg/stddev): 684419.0000/0.00
  execution time (avg/stddev): 29.7320/0.00

/ # sysbench fileio cleanup
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

Removing test files...
```

Docker	written speed (MiB/s)	Latency sum(ms)	Disk utilization
--------	-----------------------	-----------------	------------------

1	270.99	29728.42	Reads in: 547,046 Writes out: 463,331 Reads in/sec: 11 Writes out/sec: 851		Data read: 6.18 GB Data written: 62.58 GB Data read/sec: 56.0 KB Data written/sec: 328 MB
2	269.98	29731.98	Reads in: 554,802 Writes out: 493,948 Reads in/sec: 15 Writes out/sec: 871		Data read: 6.21 GB Data written: 73.40 GB Data read/sec: 72.0 KB Data written/sec: 336 MB
3	360.21	29663.05	Reads in: 556,301 Writes out: 516,043 Reads in/sec: 12 Writes out/sec: 974		Data read: 6.22 GB Data written: 80.93 GB Data read/sec: 57.6 KB Data written/sec: 376 MB
4	206.67	30563.25	Reads in: 556,463 Writes out: 543,356 Reads in/sec: 2 Writes out/sec: 861		Data read: 6.22 GB Data written: 90.88 GB Data read/sec: 8.80 KB Data written/sec: 333 MB
5	273.60	29729.69	Reads in: 556,814 Writes out: 565,896 Reads in/sec: 1 Writes out/sec: 979		Data read: 6.22 GB Data written: 98.89 GB Data read/sec: 7.20 KB Data written/sec: 364 MB
avg	276.29	29883.28			write 907.2 out/s

Threads = 1, file-num = 64, time = 30s

```
sandyhsueh@Sandy-s-MacBook-Pro ~ % docker run --rm -it --entrypoint /bin/sh zyclonite/sysbench
/ # sysbench fileio --file-num=64 prepare
```

```
/ # sysbench fileio --file-num=64 --file-test-mode=seqrewr --time=30 run
/ # sysbench fileio cleanup
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)
Removing test files...
```

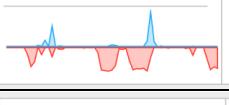
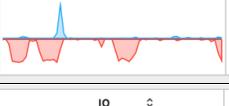
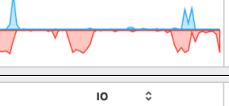
```
File operations:
  reads/s:                      0.00
  writes/s:                     12106.34
  fsyncs/s:                      7749.81

Throughput:
  read, MiB/s:                  0.00
  written, MiB/s:                189.16

General statistics:
  total time:                   30.7818s
  total number of events:       611218

Latency (ms):
  min:                           0.00
  avg:                           0.05
  max:                          6008.89
  95th percentile:                 0.09
  sum:                          30489.08

Threads fairness:
  events (avg/stddev):         611218.0000/0.00
  execution time (avg/stddev):  30.4891/0.00
```

Docker	written speed (MiB/s)	Latency sum(ms)	Disk utilization			
1	223.53	29687.03	Reads in: 517,294 Writes out: 346,544 Reads in/sec: 63 Writes out/sec: 651	IO 	Data read: 5.92 GB Data written: 26.19 GB Data read/sec: 430 KB Data written/sec: 225 MB	
2	217.81	29691.77	Reads in: 524,998 Writes out: 380,856 Reads in/sec: 10 Writes out/sec: 602	IO 	Data read: 5.96 GB Data written: 37.50 GB Data read/sec: 49.6 KB Data written/sec: 231 MB	
3	158.73	29767.15	Reads in: 527,500 Writes out: 395,764 Reads in/sec: 28 Writes out/sec: 632	IO 	Data read: 5.98 GB Data written: 41.54 GB Data read/sec: 298 KB Data written/sec: 189 MB	
4	236.19	29682.86	Reads in: 534,805 Writes out: 411,602 Reads in/sec: 13 Writes out/sec: 647	IO 	Data read: 6.05 GB Data written: 46.15 GB Data read/sec: 102 KB Data written/sec: 234 MB	
5	233.64	29689.99	Reads in: 535,949 Writes out: 436,486 Reads in/sec: 4 Writes out/sec: 623	IO 	Data read: 6.06 GB Data written: 55.04 GB Data read/sec: 19.2 KB Data written/sec: 239 MB	
avg	213.98	29703.76	write 631 out/s			

Threads = 1, file-num = 128, time = 30s

```

/ # sysbench fileio --file-num=128 prepare
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

128 files, 16384Kb each, 2048Mb total
Creating files for the test...
/ # sysbench fileio --file-num=128 --file-test-mode=seqrewr --time=30 run
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

Extra file open flags: (none)
128 files, 16MiB each
2GiB 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:        8502.59
  fsyncs/s:       10885.49

Throughput:
  read, MiB/s:      0.00
  written, MiB/s: 132.85

General statistics:
  total time:      30.0128s
  total number of events: 581793

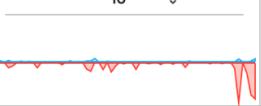
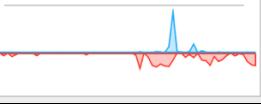
Latency (ms):
  min:              0.00
  avg:              0.05
  max:             369.96
  95th percentile: 0.10
  sum:            29688.76

Threads fairness:
  events (avg/stddev): 581793.0000/0.00
  execution time (avg/stddev): 29.6888/0.00

/ # sysbench fileio cleanup
sysbench 1.0.20-6ef8a4d4d7 (using bundled LuaJIT 2.1.0-beta2)

Removing test files...

```

Docker	written speed (MiB/s)	Latency sum(ms)	Disk utilization			
			Reads in:	Writes out:	IO	Data read:
1	132.85	29688.76	560,413	590,285		6.25 GB
2	72.68	29843.13	570,349	603,080		6.29 GB
3	99.35	29790.39	570,893	612,353		111.98 GB

4	103.96	29776.97	<p>Reads in: 571,085 Writes out: 620,807 Reads in/sec: 0 Writes out/sec: 338</p> <p>Data read: 6.29 GB Data written: 114.60 GB Data read/sec: 1.60 KB Data written/sec: 130 MB</p>
5	76.91	29830.28	<p>Reads in: 571,148 Writes out: 628,667 Reads in/sec: 0 Writes out/sec: 364</p> <p>Data read: 6.29 GB Data written: 117.15 GB Data read/sec: 2.40 KB Data written/sec: 138 MB</p>
avg	97.15	29785.9	write 388.8 out/s

C. CPU (QEMU)

CPU = 1, Memory = 16G, threads= 1, max-prime = 10000, time = 30s

```
sandyhsueh@sandy-s-MacBook-Pro:~/ubuntu_vm$ qemu-system-x86_64 \
    -machine type=q35,accel=hvf \
    -smp 1 \
    -cpu host \
    -hda ubuntu_drive.qcow2 \
    -m 16G \
    -vga virtio \
    -usb \
    -device usb-tablet \
    -display default,show-cursor-on -cdrom ./ubuntu-20.04.4-live-server-amd64.iso
```

```
sandyhsueh@241hw:~$ sysbench --test=cpu --cpu-max-prime=10000 --time=30 run
```

QEMU	CPU speed (events/s)	CPU utilization		
1	2575.02	<p>System: 18.33% User: 7.63% Idle: 74.04%</p> <p>Threads: 1,827 Processes: 495</p>		
2	2600.41	<p>System: 14.99% User: 5.67% Idle: 79.33%</p> <p>Threads: 1,793 Processes: 499</p>		
3	2537.04	<p>System: 15.43% User: 7.81% Idle: 76.75%</p> <p>Threads: 1,825 Processes: 492</p>		
4	2599.24	<p>System: 13.48% User: 3.93% Idle: 82.59%</p> <p>Threads: 1,733 Processes: 495</p>		
5	2615.45	<p>System: 14.71% User: 4.90% Idle: 80.39%</p> <p>Threads: 1,739 Processes: 493</p>		
avg	2585.43	System: 15.39% / User: 5.99%		

Latency	min	avg	max	95th	sum
1	0.29	0.36	8.72	0.49	28002.39
2	0.29	0.36	8.33	0.48	28020.49
3	0.29	0.37	5.20	0.51	27976.06
4	0.29	0.36	6.43	0.49	28026.10
5	0.29	0.36	3.96	0.48	28027.43

avg	0.29	0.36	6.53	0.49	28010.49
-----	------	------	------	------	----------

CPU = 2, Memory = 16G, threads= 1, max-prime = 10000, time = 30s

```
sandyhsueh@sandy-s-MacBook-Pro:~/ubuntu-vm% qemu-system-x86_64 \
-m 16G \
-smp 2 \
-cpu host \
-hda ubuntu_drive.qcow2 \
-memtune none \
-vga virtio \
-usb \
-device usb-tablet \
-display default,show-cursor=on -cdrom ./ubuntu-20.04.4-live-server-amd64.iso
sandyhsueh@sandy-s-MacBook-Pro:~/ubuntu-vm% sysbench --test=cpu --cpu-max-prime=10000 --time=30 run
```

QEMU	CPU speed (events/s)	CPU utilization		
1	2884.26	System: 20.75% User: 14.71% Idle: 64.54%	CPU LOAD	Threads: 1,742 Processes: 509
2	2898.18	System: 15.13% User: 14.29% Idle: 70.58%	CPU LOAD	Threads: 1,689 Processes: 498
3	2842.81	System: 15.25% User: 16.35% Idle: 68.40%	CPU LOAD	Threads: 1,712 Processes: 495
4	2862.80	System: 15.35% User: 14.65% Idle: 70.00%	CPU LOAD	Threads: 1,722 Processes: 497
5	2965.19	System: 14.56% User: 15.83% Idle: 69.61%	CPU LOAD	Threads: 1,644 Processes: 498
avg	2870.65	System: 15.6% / User: 15.17%		

Latency	min	avg	max	95th	sum
1	0.28	0.35	2.11	0.44	29908.41
2	0.32	0.34	4.94	0.42	29923.95
3	0.30	0.35	3.54	0.46	29904.09
4	0.30	0.35	3.36	0.44	29921.43
5	0.28	0.34	2.69	0.43	29927.16
avg	0.30	0.35	3.33	0.44	29917.01

CPU = 2, Memory = 32G, threads= 1, max-prime = 10000, time = 30s

```
sandyhsueh@sandy-s-MacBook-Pro:~/ubuntu_vm% qemu-system-x86_64 \
-m 32G \PU \
-smp 2 \
-cpu host \
-hda ubuntu_drive.qcow2 \
-m 32G \PU \
-vga virtio \
-usb \
-device usb-tablet \
-d display default,show-cursor=on -cdrom ./ubuntu-20.04.4-live-server-amd64.iso
sandyhsueh@241hw:~$ sysbench --test=cpu --cpu-max-prime=10000 --time=30 run
```

QEMU	CPU speed (events/s)	CPU utilization			
		System:	User:	CPU LOAD	Threads: 1,703 Processes: 491
1	2618.20	16.97%	4.91%		
2	2620.68	13.44%	2.63%		Threads: 1,651 Processes: 496
3	2545.02	15.48%	8.39%		Threads: 1,733 Processes: 496
4	2645.74	13.48%	2.70%		Threads: 1,647 Processes: 495
5	2599.95	13.48%	2.83%		Threads: 1,641 Processes: 497
avg	2585.43	System: 14.55% / User: 4.29%			

Latency	min	avg	max	95th	sum
1	0.28	0.36	2.28	0.48	28011.72
2	0.29	0.36	3.55	0.49	28023.66
3	0.29	0.37	5.46	0.51	27964.08
4	0.29	0.35	5.72	0.47	28039.31
5	0.29	0.36	4.45	0.49	28009.92
avg	0.29	0.36	4.29	0.49	28009.74

D. File I/O (QEMU)

Threads = 1, file-num = 32, time = 30s

```
sysbench fileio --file-num=32 prepare
sysbench fileio --file-num=32 --file-test-mode=seqrewr --time=30 run
sysbench --test=fileio cleanup
```

```

File operations:
  reads/s:          0.00
  writes/s:        4498.17
  fsyncs/s:        1439.45

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   70.28

General statistics:
  total time:       30.0107s
  total number of events: 178169

Latency (ms):
  min:                0.01
  avg:                0.14
  max:               49.97
  95th percentile:    0.43
  sum:            25437.76

Threads fairness:
  events (avg/stddev): 178169.0000/0.00
  execution time (avg/stddev): 25.4378/0.00

```

QEMU	written speed (MiB/s)	Latency sum(ms)	Disk utilization			
			Reads in:	Writes out:	IO	Data read:
1	70.28	25437.76	923,152	937,926		Data read: 9.15 GB Data written: 133.94 GB Data read/sec: 198 KB Data written/sec: 71.1 MB
2	69.83	25447.20	928,766	950,679		Data read: 9.21 GB Data written: 136.36 GB Data read/sec: 114 KB Data written/sec: 67.6 MB
3	69.02	25452.60	928,852	956,010		Data read: 9.21 GB Data written: 137.65 GB Data read/sec: 0 bytes Data written/sec: 77.2 MB
4	69.76	25445.20	929,032	964,983		Data read: 9.21 GB Data written: 140.17 GB Data read/sec: 49.6 KB Data written/sec: 71.4 MB
5	70.95	25396.20	935,428	970,982		Data read: 9.24 GB Data written: 141.59 GB Data read/sec: 0 bytes Data written/sec: 71.5 MB
avg	69.97	25435.87	write 235.6 out/sec			

Threads = 1, file-num = 64, time = 30s

```

sysbench fileio --file-num=64 prepare
sysbench fileio --file-num=64 --file-test-mode=seqrewr --time=30 run
sysbench --test=fileio cleanup

```

QEMU	written speed (MiB/s)	Latency sum(ms)	Disk utilization	
			Reads in:	Writes out:

1	42.67	26476.05	Reads in: 936,916 Writes out: 986,180 Reads in/sec: 1 Writes out/sec: 182		Data read: 9.24 GB Data written: 147.02 GB Data read/sec: 17.6 KB Data written/sec: 44.2 MB
2	43.65	26410.08	Reads in: 940,352 Writes out: 992,085 Reads in/sec: 32 Writes out/sec: 119		Data read: 9.27 GB Data written: 148.21 GB Data read/sec: 141 KB Data written/sec: 44.7 MB
3	43.87	26364.25	Reads in: 943,500 Writes out: 997,877 Reads in/sec: 4 Writes out/sec: 172		Data read: 9.29 GB Data written: 149.35 GB Data read/sec: 23.2 KB Data written/sec: 46.9 MB
4	43.86	26446.55	Reads in: 944,506 Writes out: 1,003,681 Reads in/sec: 10 Writes out/sec: 120		Data read: 9.30 GB Data written: 150.98 GB Data read/sec: 48.8 KB Data written/sec: 45.0 MB
5	45.24	26319.83	Reads in: 944,806 Writes out: 1,008,754 Reads in/sec: 0 Writes out/sec: 120		Data read: 9.30 GB Data written: 152.52 GB Data read/sec: 1.60 KB Data written/sec: 46.3 MB
avg	43.86	26403.35		write 142.6 out/sec	

Threads = 1, file-num = 128, time = 30s

```
sysbench fileio --file-num=128 prepare
sysbench fileio --file-num=128 --file-test-mode=seqrewr --time=30 run
sysbench --test=fileio cleanup
```

QEMU	written speed (MiB/s)	Latency sum(ms)	Disk utilization			
			Reads in:	Writes out:	Data read:	Data written:
1	25.54	27063.27	950,533	1,017,509	9.32 GB	155.60 GB
2	26.10	26973.71	951,840	1,022,578	9.33 GB	156.28 GB
3	26.27	26939.21	952,170	1,025,213	9.33 GB	156.74 GB
4	25.05	27082.77	954,119	1,029,249	9.34 GB	157.53 GB
5	25.32	27054.69	954,506	1,032,128	9.34 GB	158.26 GB
avg	25.66	27022.73		write 100.2 out/sec		

Conclusion:

Docker	CPU	cpus=1, m=16g	cpus=2, m=16g	cpus=2, m=32g
	File I/O	file-num=32	file-num=64	file-num=128
QEMU	CPU	cpus=1, m=16g	cpus=2, m=16g	cpus=2, m=32g
	File I/O	file-num=32	file-num=64	file-num=128

CPU				
Docker			QEMU	
2577.87(events/s)	System:15.89% / User: 4.60%		2585.43(events/s)	System:15.39% / User: 5.99%
2590.48(events/s)	System:17.03% / User: 5.85%		2870.65(events/s)	System:15.6% / User: 5.17%
2650.86(events/s)	System:15.37% / User: 5.14%		2585.43(events/s)	System:14.55% / User: 4.29%

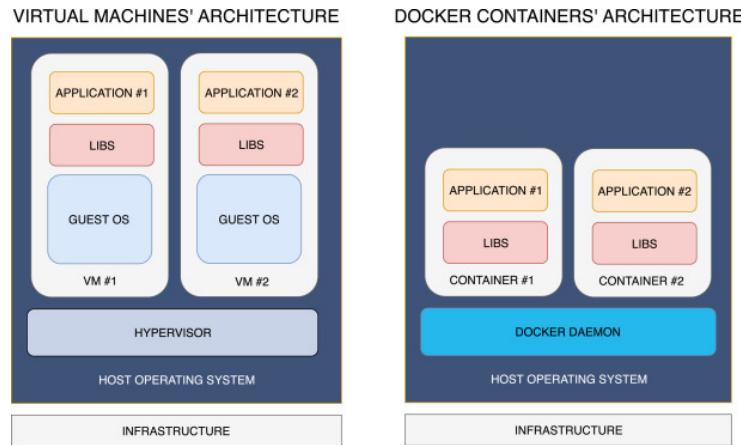
File I/O					
Docker			QEMU		
276.29(MiB/s)	29883.28(ms)	write 907.2 (out/sec)	69.97(MiB/s)	25435.87(ms)	write 235.6 (out/sec)
213.98(MiB/s)	29703.76(ms)	write 631 (out/sec)	43.86(MiB/s)	26403.35(ms)	write 142.6 (out/sec)
97.15(MiB/s)	29785.9 (ms)	write 388.8 (out/sec)	25.66(MiB/s)	27022.73(ms)	write 100.2 (out/sec)

Observation from the experiences:

1. Overall performance:

	Docker	QEMU
The overall performance of the CPU	slower	faster
The overall performance of File I/O	faster	slower

The result may be due to the different architecture between Docker and QEMU. For CPU, QEMU is faster than Docker because the hypervisor can manage the virtual CPU, and the path from VM to hypervisor is shorter than the path in Docker from container to hardware. However, for file I/O, Docker is faster than QEMU because the route from container to hardware is shorter than the path for QEMU from VM to hardware.



2. For CPU performance, both Docker and QEMU have a similar result in a different scenario. The reason for it is probably because the thread for all of it is 1. Also, the memory did not affect the result.
3. For File I/O performance, for both Docker and QEMU, the speed decreased, and the write-out per second dropped when the file number increased. For QEMU, the latency increased when the file number increased.

**Present and analysis the performance data:
Understandability/ Neatness of my report:**

Git repository information: