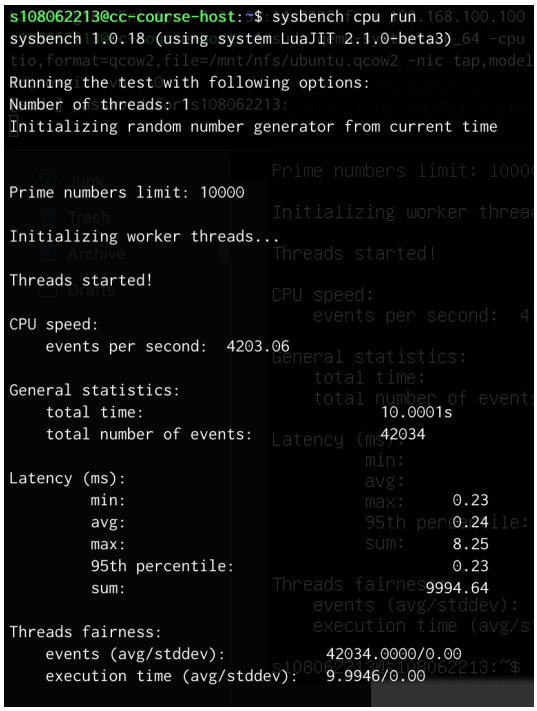
108062213 顏浩昀
Cloud Computing HW1
2022.04.10

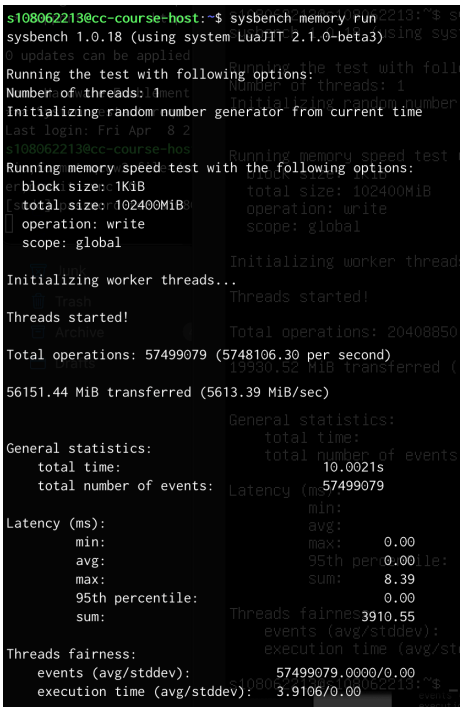
A. Please simply explain how do you set up your VM.

I set the VM with 10G disk, 1 core with 1 thread, 1Gigabytes ram, using Ubuntu os system(version:20.04.4) by starting QEMU with x86_64 architecture, enabling kvm to use CPU extension for virtualization, and connecting to internet by setup bridge network with TAP interface.

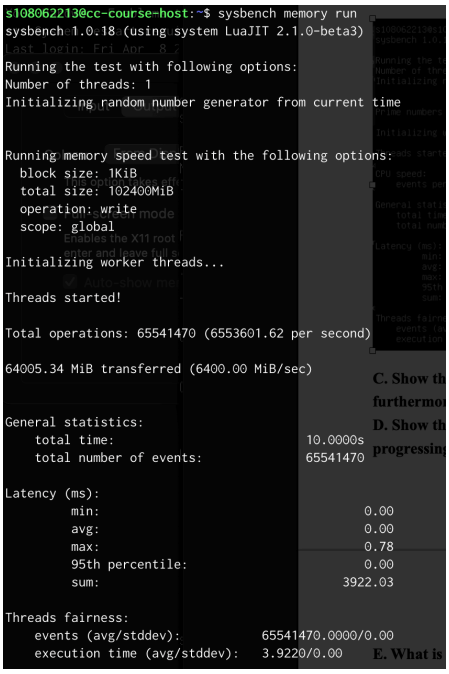
B. Show the performance testing results by sysbench with and without "-enable-kvm" on VM, and Host; furthermore compare among them and explain the result.

-with enable-kvm on host

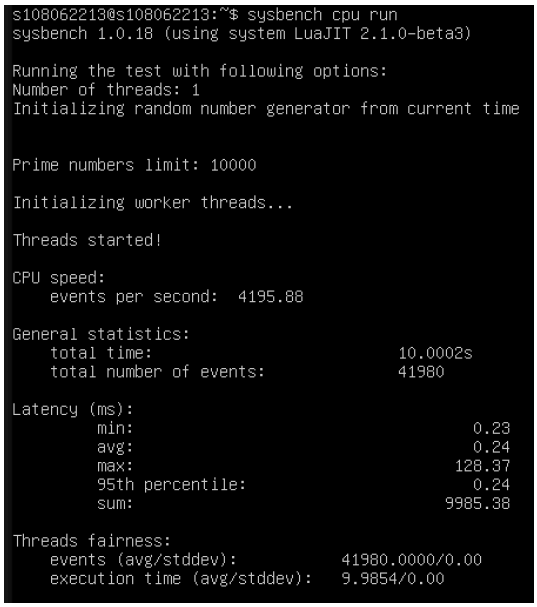


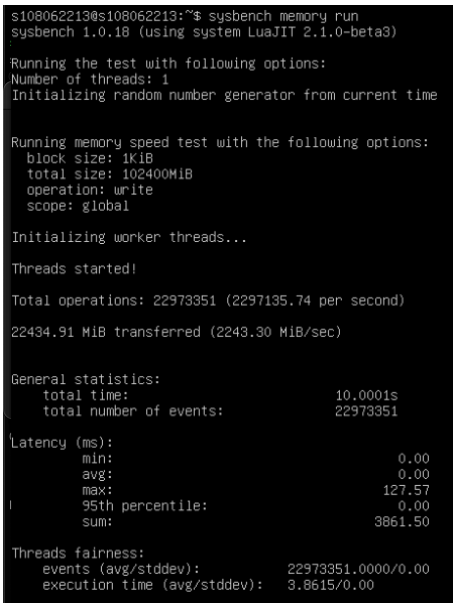


-without enable-kvm on host

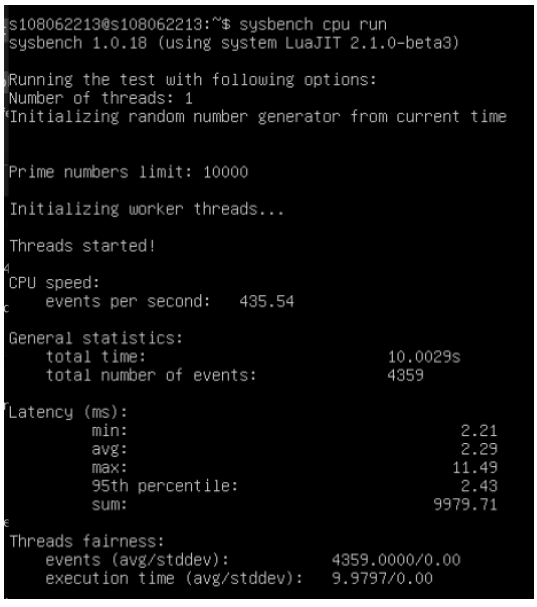


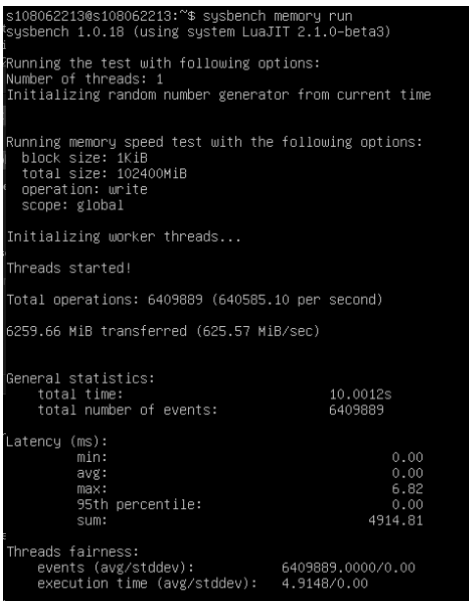
-with enable-kvm on VM





-without enable-kvm on VM





Obviously, there is significantly effect on VM. CPU speed of a VM with enable-kvm can run 10 times events more than a VM of without enable-kvm. Moreover, the number of event in 10 seconds process by memory previous one is about 3 times larger than a VM without enable-kvm.
KVM is a module in linux kernel space. It can virtualize CPU and memory and accelerate it by hardware support, thus CPU and memory speed sharply increase.

C. Show the performance testing results by iperf with and without "virtio" on VM, and Host; furthermore compare among them and explain the results.

-with virtio on host

```
^Cs108062213@cc-course-host:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 192.168.124.164 port 5001 connected with 192.168.252.94 port 39426
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec  4.39 GBytes  3.77 Gbits/sec
```

-without virtio on host

```
^Cs108062213@cc-course-host:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 192.168.124.164 port 5001 connected with 192.168.252.94 port 37460
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec  1.08 GBytes   930 Mbits/sec
```

-with virtio on VM

```
s108062213@s108062213:~$ iperf -c 192.168.124.164
------------------------------------------------------------
Client connecting to 192.168.124.164, TCP port 5001
TCP window size:  510 KByte (default)
------------------------------------------------------------
[  3] local 192.168.252.94 port 39426 connected with 192.168.124.164 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  4.39 GBytes  3.77 Gbits/sec
```

-without virtio on VM

```
s108062213@s108062213:~$ iperf -c 192.168.124.164
------------------------------------------------------------
Client connecting to 192.168.124.164, TCP port 5001
TCP window size:  255 KByte (default)
------------------------------------------------------------
[  3] local 192.168.252.94 port 37460 connected with 192.168.124.164 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  1.08 GBytes   932 Mbits/sec
```

Without "virtio" the bandwidth of network is down from 3.7Gbits to 930Mbits per second.

-nic is a command in qemu for network setting. "virtio" is proposed by IBM in 2005. It uses virtqueue to transfer package between frontend and backend and reduces complex I/O operation during transfer data with e1000.

D. Show the performance measurements by iperf and sysbench during the live migration is progressing; furthermore, describe your observations.

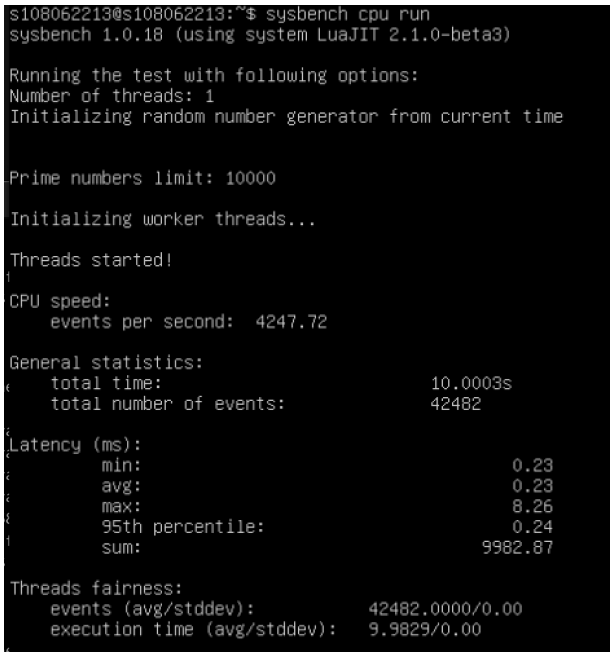
-on host

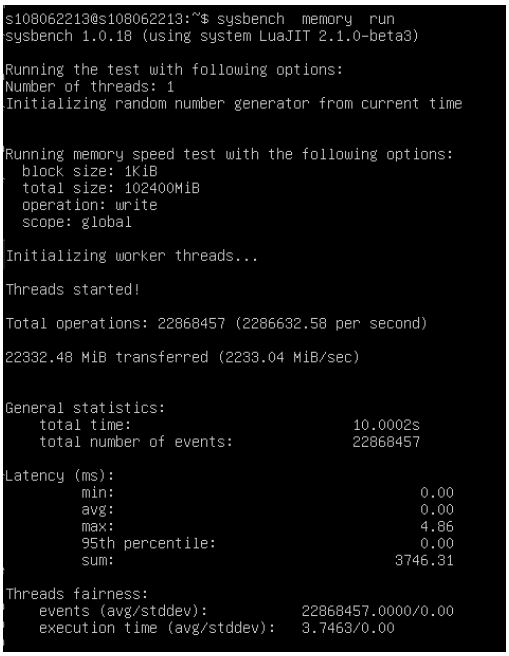
```
s108062213@cc-course-host:~$ sysbench cpu run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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


Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  2391.17

General statistics:
    total time:                          10.0035s
    total number of events:              23922

Latency (ms):
         min:                                    0.23
         avg:                                    0.42
         max:                                   12.24
         95th percentile:                        0.24
         sum:                                 9995.84

Threads fairness:
    events (avg/stddev):           23922.0000/0.00
    execution time (avg/stddev):   9.9958/0.00
```

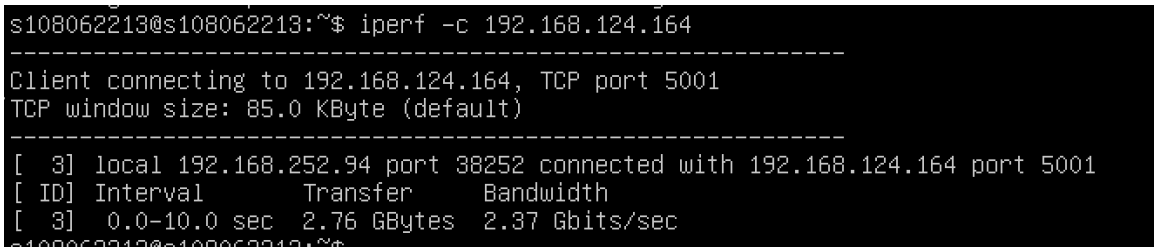
```
s108062213@cc-course-host:~$ sysbench memory run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

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


Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 36067834 (3606487.56 per second)

35222.49 MiB transferred (3521.96 MiB/sec)


General statistics:
    total time:                          10.0000s
    total number of events:              36067834

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                    9.09
         95th percentile:                        0.00
         sum:                                 3856.01

Threads fairness:
    events (avg/stddev):           36067834.0000/0.00
    execution time (avg/stddev):   3.8560/0.00
```

```
^Cs108062213@cc-course-host:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 192.168.124.164 port 5001 connected with 192.168.252.94 port 38252
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec  2.76 GBytes  2.37 Gbits/sec
```

-on VM







On my observation, when I do migration the performance on host sharply decreasing. Its CPU speed go from 4203.06 events per second to 2391.17 events per second, and its memory process 3606487.56 operations per seconds from 5748106.3 operations per seconds. Both CPU and memory decrease almost half performance ratio.

E. What is “live migration” and why we need it.

Live migration is the process of transferring a live VM from a host to another one without disrupting its normal operation. In practice, when a physical computer or server needs manipulation, updating, or to be switched between different hosts, and we wonder the VM on host wouldn’t be disrupted, then we can perform live migration.

F. Please describe how to maintain the network connection when the VM is being migrated.

To do live migration, I need a mirror VM on host 2, and transfer data from VM on host1 to the mirror VM. With my setting, I use bridge network to connect VM and host not only host 1 but host 2, so the mirror VM should own network with host 2 internet setting. The whole live migration process is using monitor to connect source(host 1) and destination(host 2), copy data and create mirror one on mirror VM, save status and transfer to destination, end connecting and delete source data.

G. QEMU has a fault tolerance feature called COLO.

a. What is fault-tolerance in cloud system and why we require it?

Fault tolerance in cloud system is designing blueprint for continuing the ongoing work whenever a few parts are unavailable. We need fault tolerance since we don't wonder get fail because of a few error(it's really difficulty to make everything always fine.). With fault tolerance we can keep system in running mode at a useable at least.

b. What are the relationships between live migration and fault-tolerance?

Because cloud computing infrastructure increasing, real time system be also critically safe and robust. Live migration for the purpose for fault tolerance has a considerable role to play in cloud computing system.

Live migration can be said as a kind of implement of fault-tolerance.