

Cloud Computing Homework 1 Report

Part 1. System Spec

1. Device specifications

裝置規格

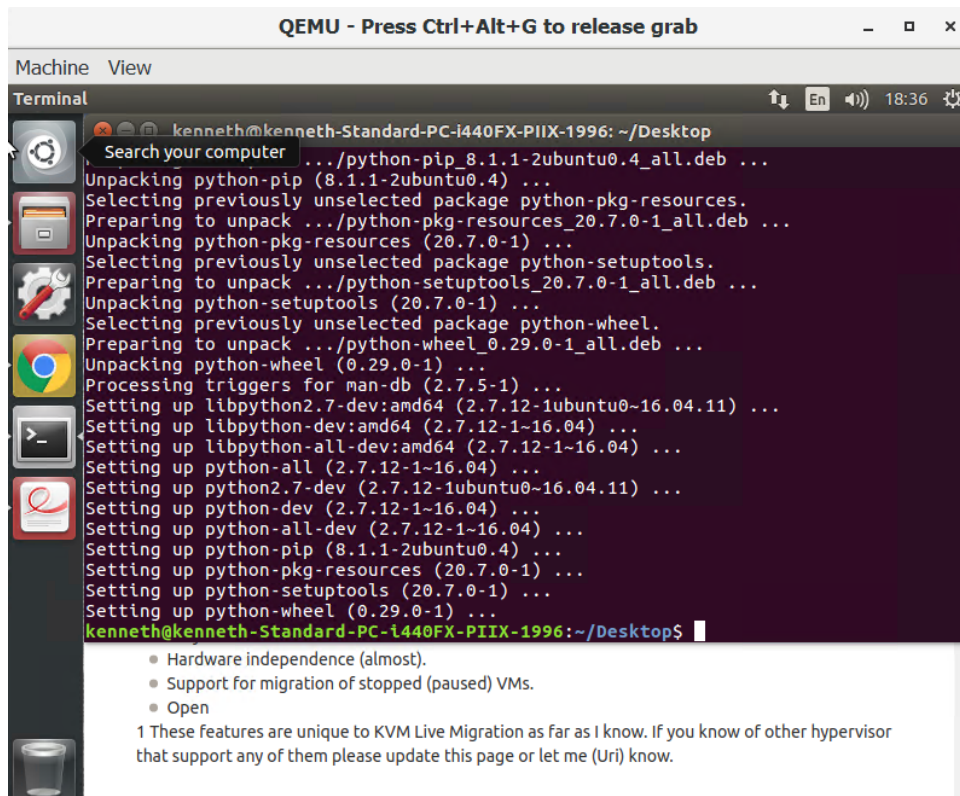
裝置名稱	DESKTOP-KTNGSFP
處理器	Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz 2.70 GHz
已安裝記憶體(RAM)	16.0 GB (15.9 GB 可用)
裝置識別碼	5711972E-F122-4921-899E-5FADC9520E3D
產品識別碼	00331-10000-00001-AA692
系統類型	64 位元作業系統，x64 型處理器
手寫筆與觸控	此顯示器不提供手寫筆或觸控式輸入功能

2. Windws specifications

Windows 規格

版本	Windows 10 專業版
版本	1903
安裝於	2020/4/16
OS 組建	18362.778

Part 2. Proof of using Qemu



Part 3. Modules modified

First, I listed all the services by command `systemd-analyze blame`.

```

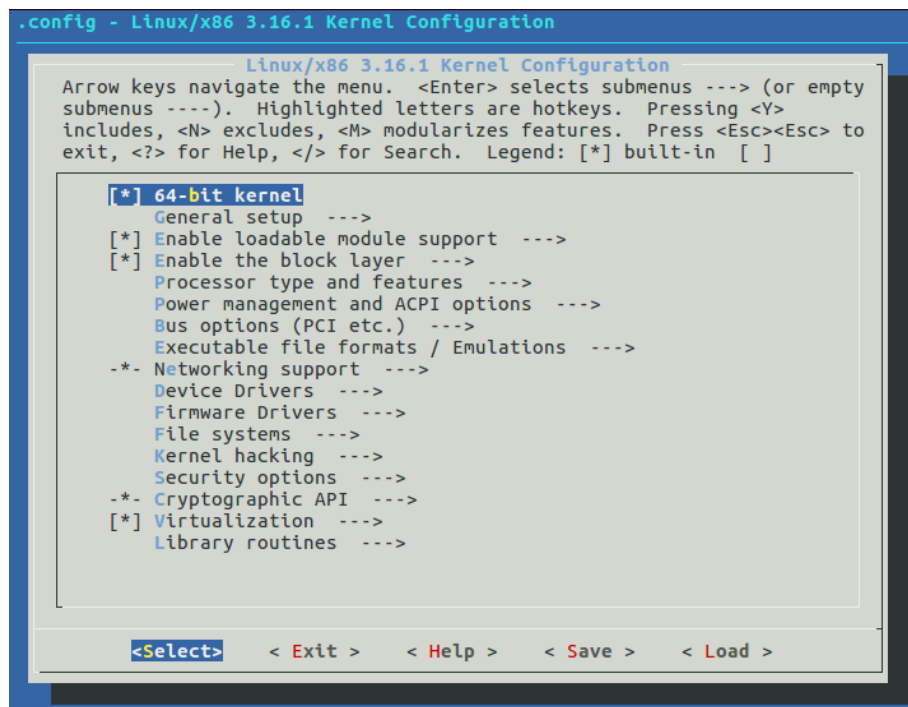
kenneth@kenneth-Standard-PC-i440FX-PIIX-1996:/usr/src$ systemd-analyze blame
5min 17.263s apt-daily-upgrade.service
26.805s apt-daily.service
5.855s NetworkManager-wait-online.service
4.493s fwupd.service
2.971s dev-sda1.device
1.343s snapd.seeded.service
1.302s apparmor.service
1.282s console-setup.service
1.007s accounts-daemon.service
1.000s plymouth-read-write.service
776ms systemd-logind.service
709ms lightdm.service
679ms keyboard-setup.service
610ms NetworkManager.service
538ms gpu-manager.service
504ms irqbalance.service
478ms ondemand.service
443ms networking.service
408ms systemd-user-sessions.service
394ms systemd-udev-trigger.service
386ms systemd-modules-load.service
344ms colord.service
339ms systemd-journald.service
324ms speech-dispatcher.service
318ms snapd.service
307ms thermal.service
289ms plymouth-start.service
270ms systemd-update-utmp.service
249ms systemd-tmpfiles-clean.service

```

And I decided to cut some of the services, which are `apparmor.service`, `apparmor.service`, `snapd`, `NetworkManager-wait-online.service`

```
kenneth@kenneth-Standard-PC-i440FX-PIIX-1996:/usr/src$ sudo systemctl disable ap
port.service
[sudo] password for kenneth:
apporport.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install disable apporport
insserv: warning: current start runlevel(s) (empty) of script `apporport' overrides
LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (2 3 4 5) of script `apporport' override
s LSB defaults (empty).
kenneth@kenneth-Standard-PC-i440FX-PIIX-1996:/usr/src$ sudo systemctl disable ap
parmor.service
apparmor.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install disable apparmor
insserv: warning: current start runlevel(s) (empty) of script `apparmor' overrid
es LSB defaults (S).
insserv: warning: current stop runlevel(s) (S) of script `apparmor' overrides LS
B defaults (empty).
kenneth@kenneth-Standard-PC-i440FX-PIIX-1996:/usr/src$ sudo systemctl disable sn
apd
Removed symlink /etc/systemd/system/multi-user.target.wants/snapd.service.
kenneth@kenneth-Standard-PC-i440FX-PIIX-1996:/usr/src$ sudo systemctl disable Ne
tworkManager-wait-online.service
Removed symlink /etc/systemd/system/network-online.target.wants/NetworkManager-w
ait-online.service.
```

For the make menuconfig, I remains the same setting as what menuconfig had ticked.



Part 4. OS image that boot faster than Ubuntu

Unfortunately, I did not find any OS image than faster than Ubuntu.

Part 5. Comparison in boot time between general config and modify config:

5.a General config

```
kenneth@kenneth-Standard-PC-i440FX-PIIX-1996:~/Desktop$ systemd-analyze
Startup finished in 3.842s (kernel) + 10.302s (userspace) = 14.145s
```

5.b Modified config

```
kenneth@kenneth-Standard-PC-i440FX-PIIX-1996:~/Desktop$ systemd-analyze
Startup finished in 3.224s (kernel) + 13.331s (userspace) = 16.556s
```

Part 6. Migrate

Part 7. MNIST executime time with different number of CPU core

EXECUTION TIME

Core	Real	User	Sys
1	1m51.381s	0m45.852s	0m1.268s
2	1m51.554s	0m45.872s	0m1.368s
3	1m58.043s	0m46.944s	0m1.252s
4	1m46.634s	0m41.252s	0m1.220s

Part 8. Difference between cold migration, warm migration, live migration

1. Cold migration
 - When the domain is not currently running on the source system.
 - Almost instantaneous
 - Since only metadata is moved. After migration the domain is defined on the target, but remains inactive.
2. Warm migration
 - When the domain is running on the source system
 - The LDom manager on the source system contacts the target system to start the migration and ensure the target system matches chip type and has resources to host the domain.
 - It then suspends the guest's operation, compresses and encrypts its state information (mostly RAM) using the T-series crypto acceleration, and transmits it to the target.
 - The LDom manager on the target decompresses and decrypts the contents and resumes domain operation.
 - The domain can be unresponsive for minutes (depending on memory size and network speed). but picks up from where it was.
3. Live migration
 - Also known as hot migration
 - Similar to warm migration, except that the guest is not suspended during transmission.
 - The LDom manager keeps track of memory changed while state is being transmitted, and then makes followup passes to retransmit data changed state information.
 - A brief pause is used at the end to transmit residual changed state.