

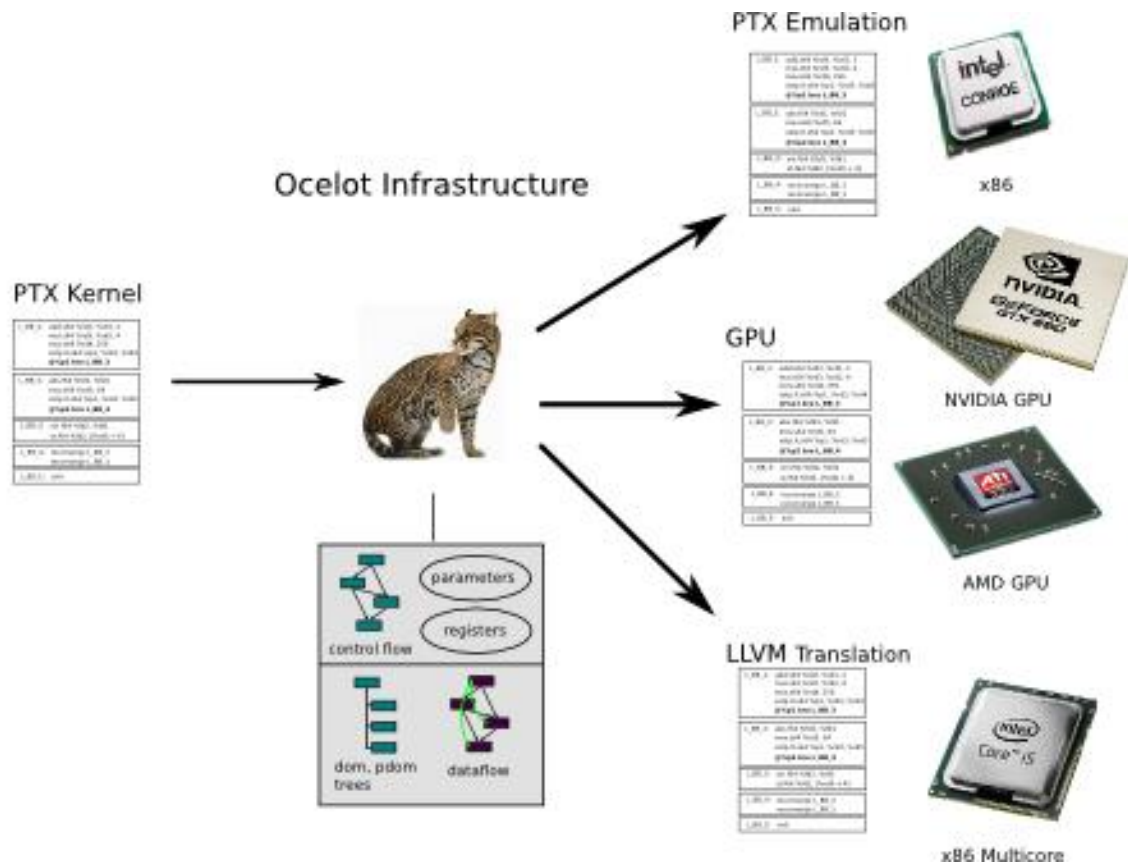
Computer Architecture:

Lab0: Ocelot for CUDA Programming & Setup Virtual Machine in PC

Computer Architecture 2020 (Prof. Chih-Wei Liu)

CUDA Tutorial

Ocelot介紹



► GNU Ocelot

► <http://gpuocelot.gatech.edu/>

► 能將NVIDIA PTX轉譯並能執行在不同類型的運算裝置上

► AMD GPU

► x86 CPU

► 也可做為emulator，當作debug或tuning的工具

虛擬機器安裝&操作:

Step 1 前置作業

- ▶ 下載Linux 虛擬機器檔 (助教已安裝與設定Ocelot)

- ① <https://goo.gl/ru7c3D>

- ▶ 請於電腦中安裝Oracle VirtualBox

- ▶ 下載頁面 <https://www.virtualbox.org/wiki/Downloads>

- ▶ Windows安裝檔連結

- <https://download.virtualbox.org/virtualbox/5.2.8/VirtualBox-5.2.8-121009-Win.exe>

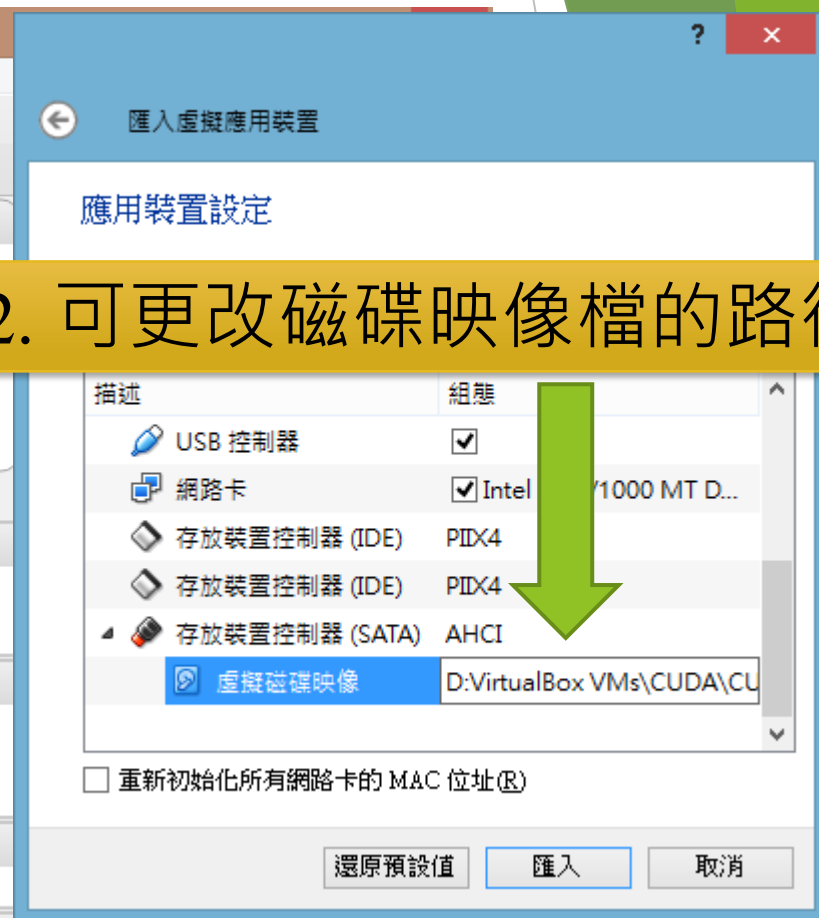
虛擬機器安裝&操作:

Step 2 設定虛擬機器 (匯入1/2)

1. 請匯入助教提供的Linux
虛擬機器檔
(CUDA_2018.ova)



2. 可更改磁碟映像檔的路徑



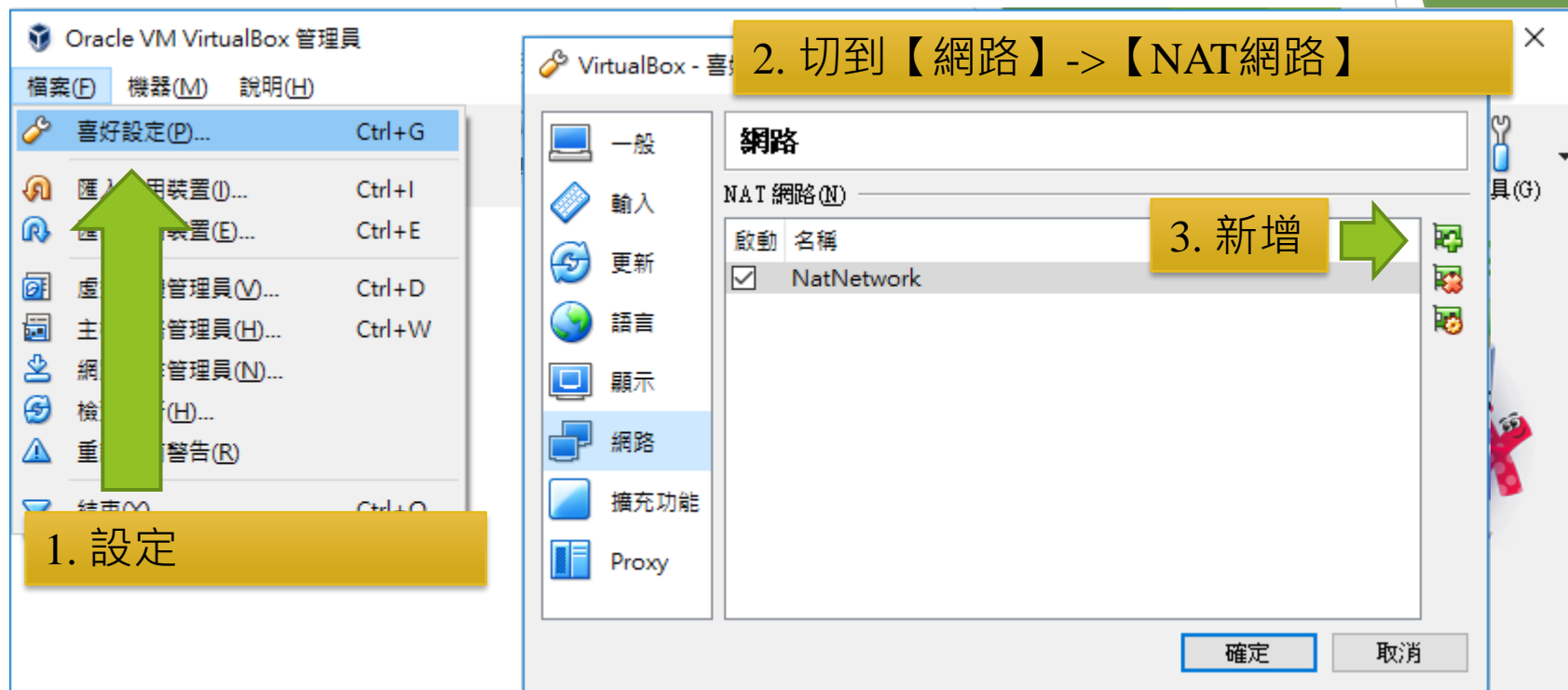
虛擬機器安裝&操作:

Step 2 設定虛擬機器 (匯入2/2)



虛擬機器安裝&操作:

Step 2 設定虛擬機器 (網路1/2)



虛擬機器安裝&操作:

Step 2 設定虛擬機器 (網路2/2)

The screenshot shows the Oracle VM VirtualBox interface. On the left, the 'CUDA_2018' virtual machine is selected. The '設定' (Settings) button is highlighted with a green circle and an arrow pointing to the '網路' (Network) tab in the settings window. The '網路' (Network) tab is active, showing the '介面卡 1' (Adapter 1) configuration. The '啟用網路卡(E)' (Enable Network Card) checkbox is checked. The '附加到(A):' (Attached to) dropdown is set to 'NAT 網路' (NAT Network). The '名稱(N):' (Name) field is set to 'NatNetwork'. A green arrow points from the '2. 選擇網路' (2. Select Network) text to the '網路' (Network) tab. Another green arrow points from the '3. 請改選 [NAT網路] 下面網路名稱請選 [NatNetwork]' (3. Please select [NAT Network] below network name please select [NatNetwork]) text to the '名稱(N):' field.

1. 開啟虛擬機器設定頁

2. 選擇網路

3. 請改選 [NAT網路]
下面網路名稱請選 [NatNetwork]

網路

介面卡 1 介面卡 2 介面卡 3 介面卡 4

☒ 啟用網路卡(E)

附加到(A): NAT 網路

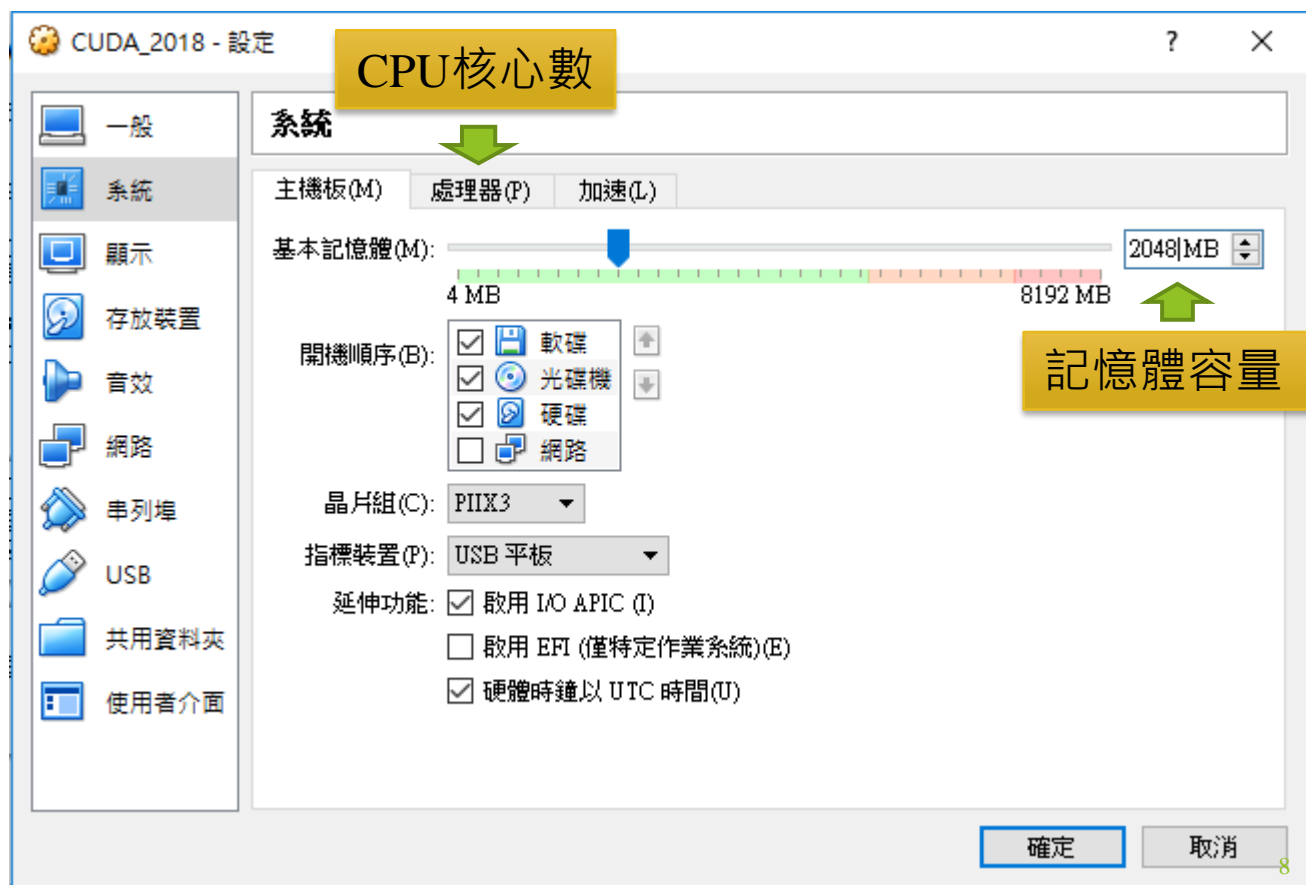
名稱(N): NatNetwork

進階(D)

確定 取消

虛擬機器安裝&操作:

Step 2 設定虛擬機器 (其他)

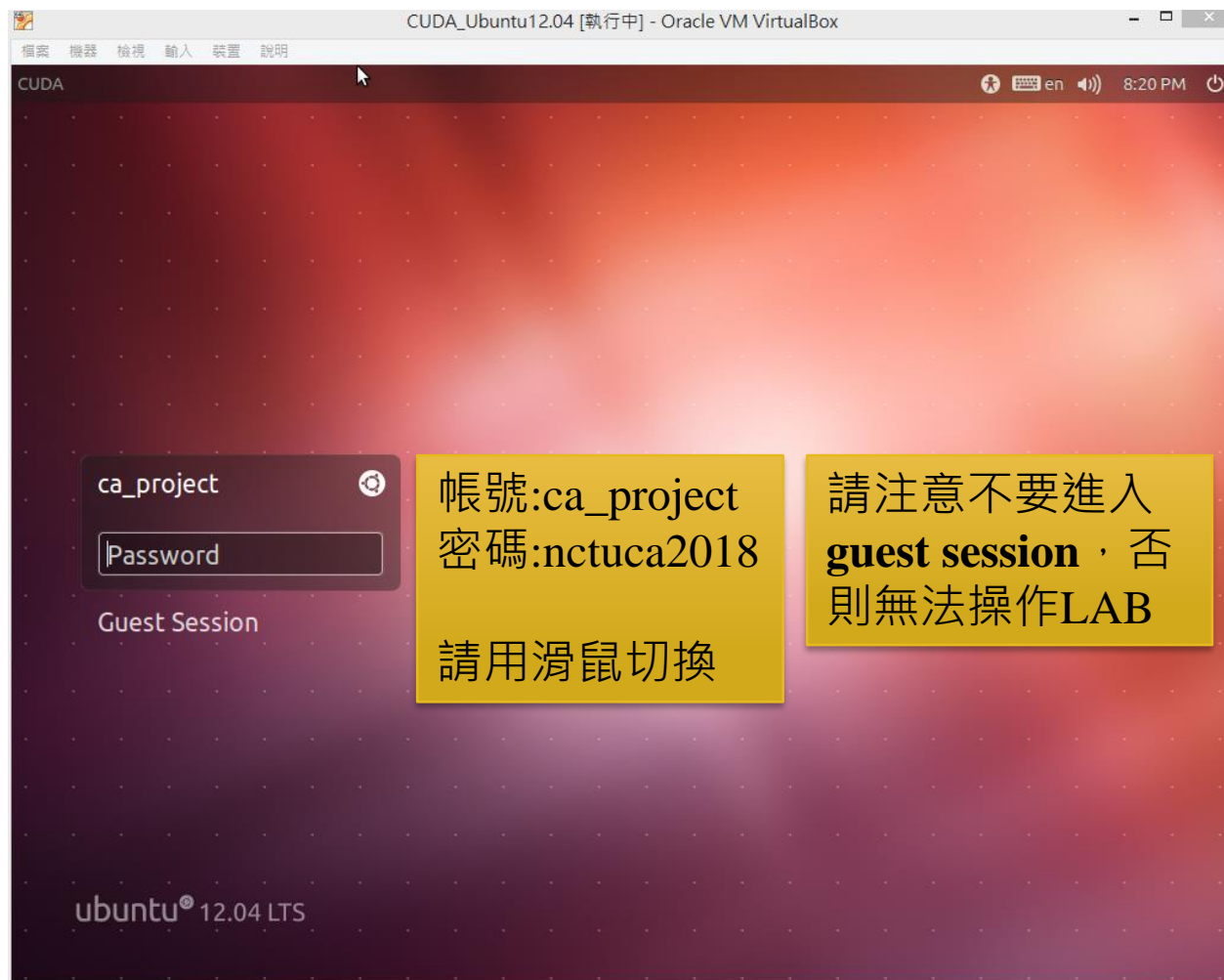


虛擬機器安裝&操作:

Step 3 開機



虛擬機器安裝&操作: Ubuntu 登入



虛擬機器安裝&操作: 桌面環境操作

類似Windows的”開始”，點開後會有常用程式，或者可以用搜尋的方式打開程式
要開終端機請打：*terminal*
(指令操作)
要開文字編輯器請打：*gedit*
(編輯程式碼)

關機/登出請按此

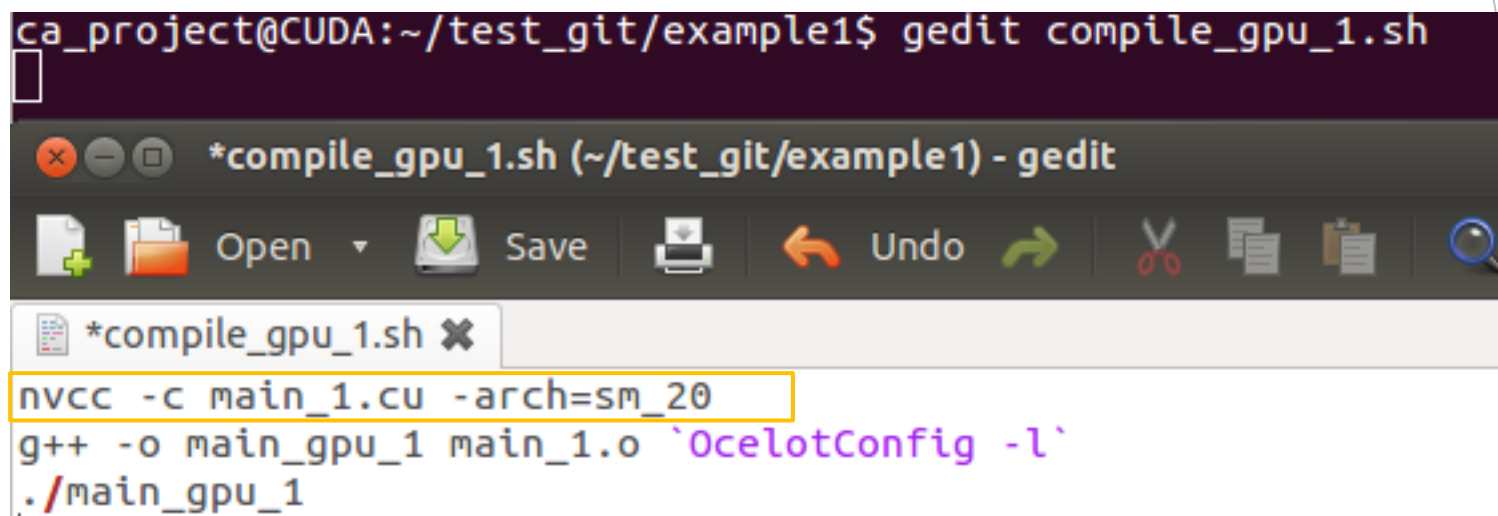


Ocelot/CUDA Programming 環境操作: 以example1為例 (試執行程式碼)

- ▶ 提供以下CUDA程式碼
 - ▶ `main_1.cu`
 - ▶ `main_2.cu`
- ▶ `main_cpu.cpp` 是同一程式的CPU版
- ▶ `configure.ocelot` 為ocelot執行的設定檔
- ▶ 以下三個script檔是編譯與執行的指令
 - ▶ `compile_cpu.sh`
 - ▶ `compile_gpu_1.sh`
 - ▶ `compile_gpu_2.sh`

Ocelot/CUDA Programming 環境操作: 以example1為例 (編譯&執行)(1/2)

- ▶ 執行 (兩者擇一)
 - ▶ ./compile_gpu_1.sh
 - ▶ sh compile_gpu_1.sh



The screenshot shows a terminal window with the following content:

```
ca_project@CUDA:~/test_git/example1$ gedit compile_gpu_1.sh
```

The window title is `*compile_gpu_1.sh (~/.test_git/example1) - gedit`. The toolbar includes icons for Open, Save, Undo, and other standard file operations. The file name `*compile_gpu_1.sh` is shown in the tab. The terminal content is as follows:

```
nvcc -c main_1.cu -arch=sm_20
g++ -o main_gpu_1 main_1.o `OcelotConfig -l`
./main_gpu_1
```

Ocelot/CUDA Programming 環境操作: 以example1為例 (編譯&執行)(2/2)

▶ 執行 (兩者擇一)

- ▶ `./compile_gpu_1.sh`
- ▶ `sh compile_gpu_1.sh`

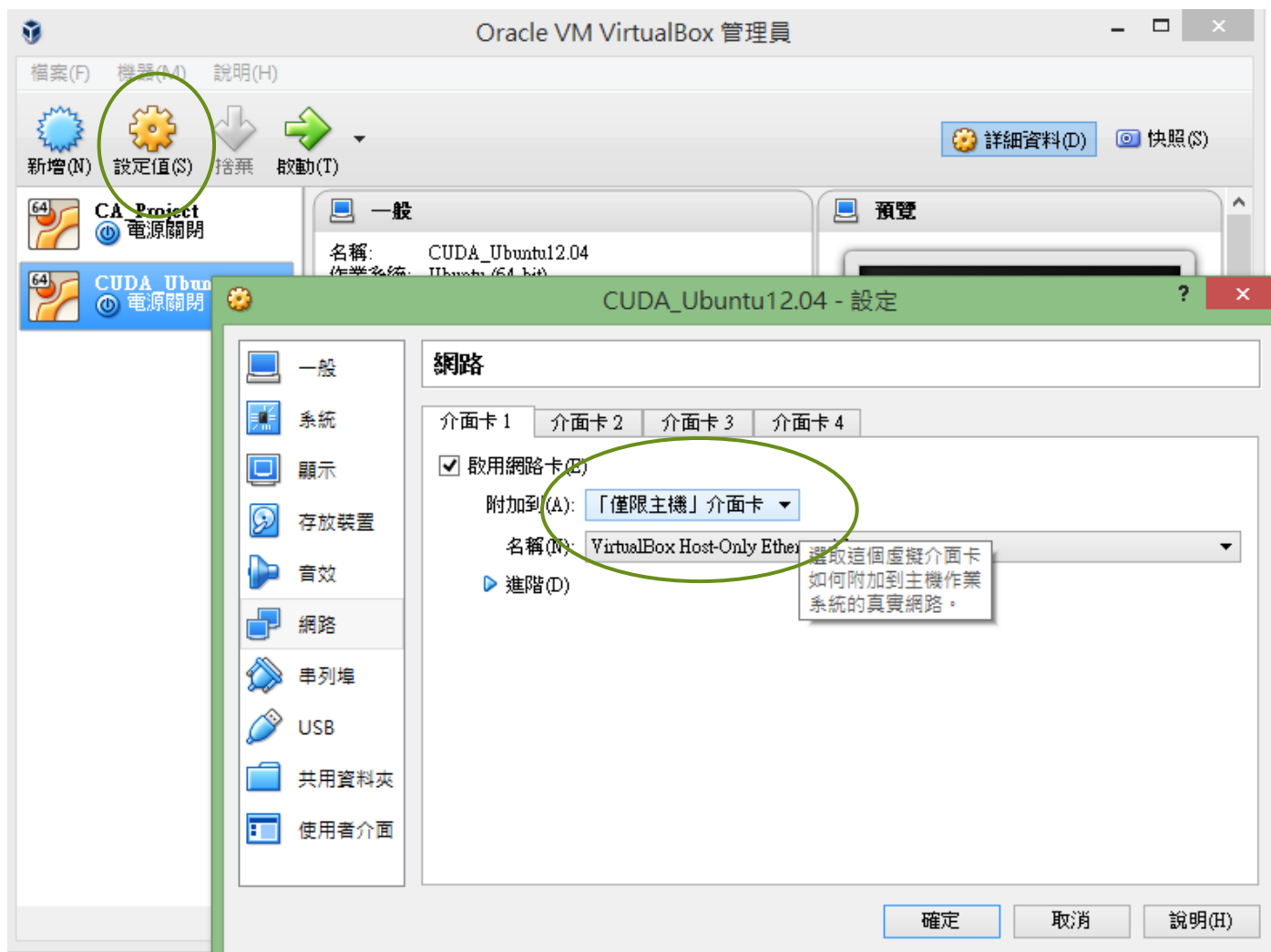
將所有object檔與外部library連結成執行檔
`g++ -o main fun1.o fun2.o `OcelotConfig -l``

```
ca_project@CUDA:~/test_git/  
[  
*compile_gpu_1.sh (/test_git/example1) - gedit  
Open [Save] [Print] [Undo] [Redo] [Find]  
*compile_gpu_1.sh  
nvcc -c main_1.c -arch=sm_20  
g++ -o main_gpu_1 main_1.o `OcelotConfig -l`  
./main_gpu_1
```

執行程式

附錄：如何存取虛擬機器中的檔案至你的電腦中

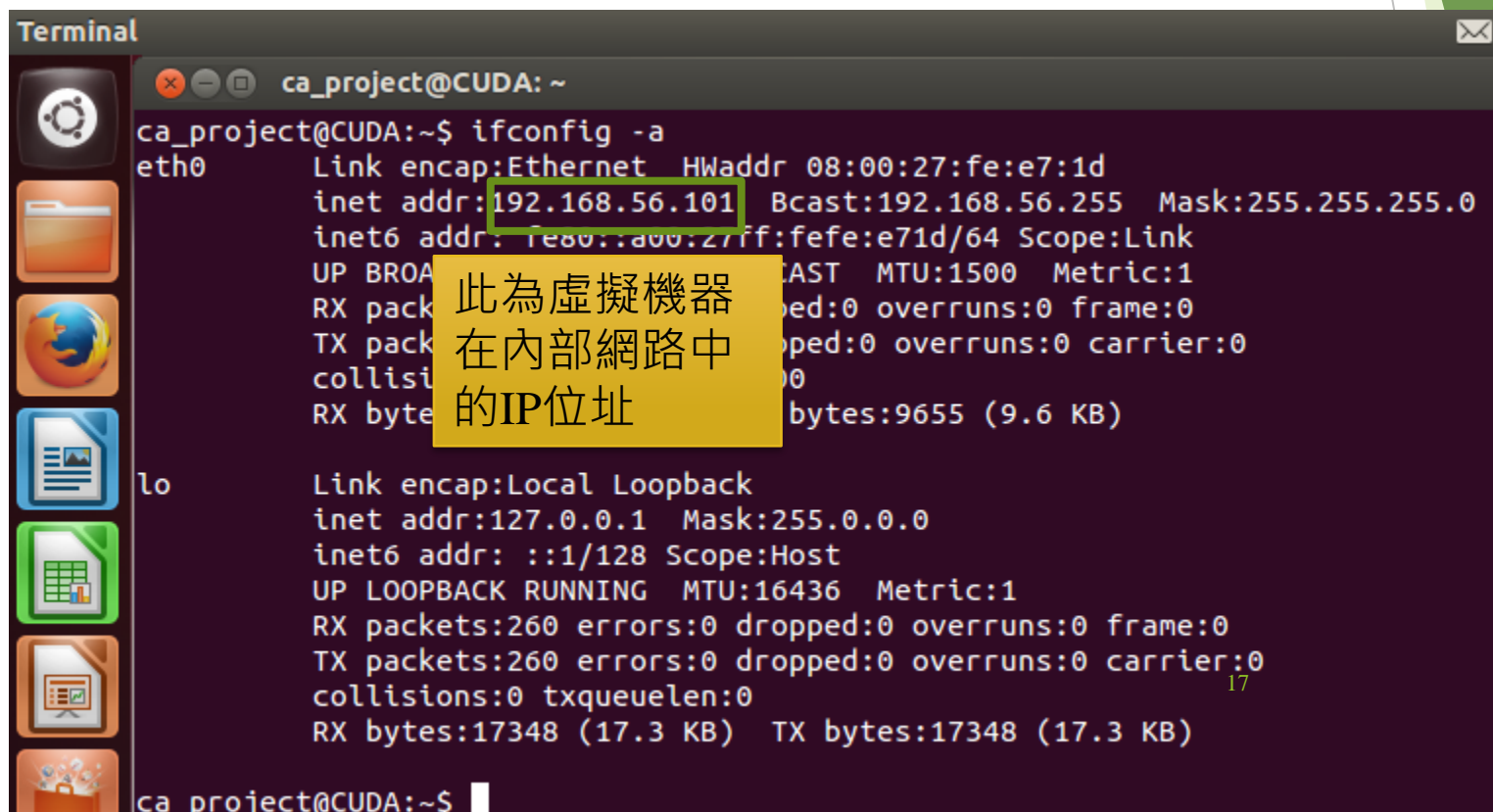
請將虛擬機器關機，並重新設定網路選項



請先查詢虛擬機器的IP位址

► 開啟terminal，輸入以下指令

► ifconfig -a



```
Terminal
ca_project@CUDA: ~
ca_project@CUDA:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:fe:e7:1d
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fefe:e71d/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:260 errors:0 dropped:0 overruns:0 frame:0
          TX packets:260 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:17348 (17.3 KB)  TX bytes:17348 (17.3 KB)

ca_project@CUDA:~$
```

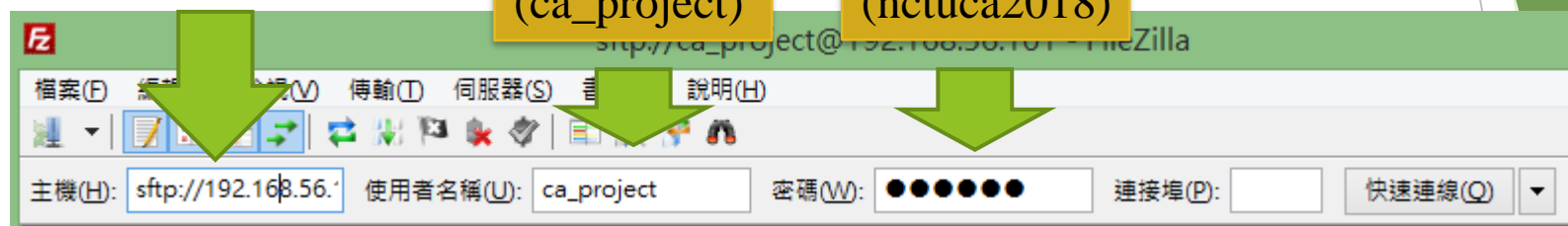
此為虛擬機器在內部網路中的IP位址

在自己電腦上打開Filezilla

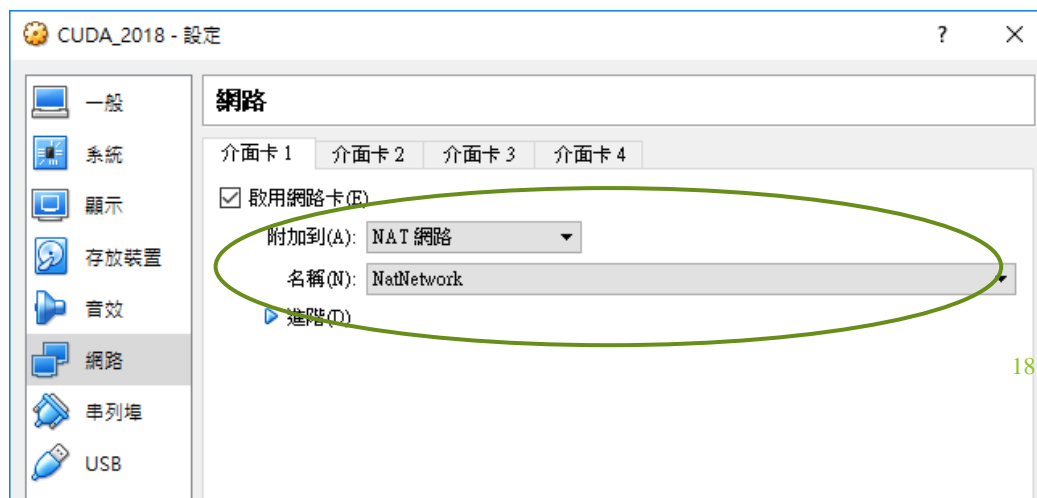
輸入上頁查詢的IP位址

輸入虛擬機
器的帳號
(ca_project)

輸入虛擬機
器的密碼
(nctuca2018)



- 由於此網路環境無法對外聯網，請同學存取完檔案後，關機進入設定頁面，將網路改回原來的設定 (可參考step2)

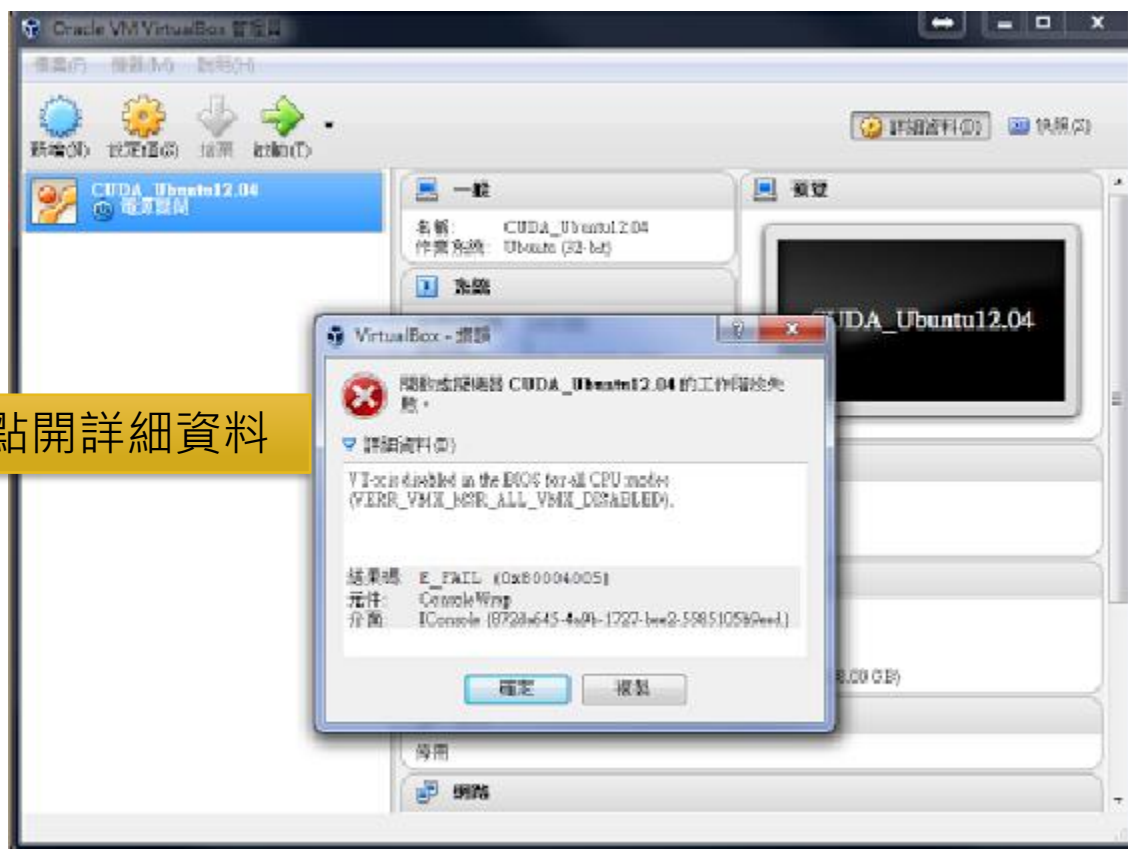


附錄：無法啟動虛擬機器

原因：CPU未開啟或未支援VM加速功能

Q:無法啟動虛擬機器 (原因：CPU未開啟或未支援 VM加速功能)

請點選詳細資料



解決方法1：

請重開機進入主機板BIOS (各家不同請自己查詢)，開啟CPU的虛擬化技術功能。
[Intel vt-*] or [AMD-v]

解決方法2：

- ① 請在左欄虛擬主機按右鍵【移除】
- ② 移除時會有詢問視窗，請點【僅移除】
- ③ 重新【新增】虛擬機器
- ④ 名稱請打Ubuntu
- ⑤ 【記憶體】用量請自填
- ⑥ 【硬碟】請選【使用現有虛擬硬碟檔】
[CUDA_Ubuntu... .vmdk
8G容量]
- ⑦ 完成新增後，再按照助教設定網路
- ⑧ 最後請至【設定值】->【系統】->【處理器】將【啟用PAE/NX】勾起

附錄：其他

VirtualBox Guest Additions

- ▶ 讓虛擬機器能調整螢幕大小
- ▶ 支援滑鼠滾輪
- ▶ 安裝VirtualBox Guest Additions (密碼請輸入nctuca2018)
 - ▶ `sudo apt-get install virtualbox-guest-additions`
 - ▶ `sudo apt-get install virtualbox-guest-x11`
- ▶ 重開機