

# Kaldi Working Environment

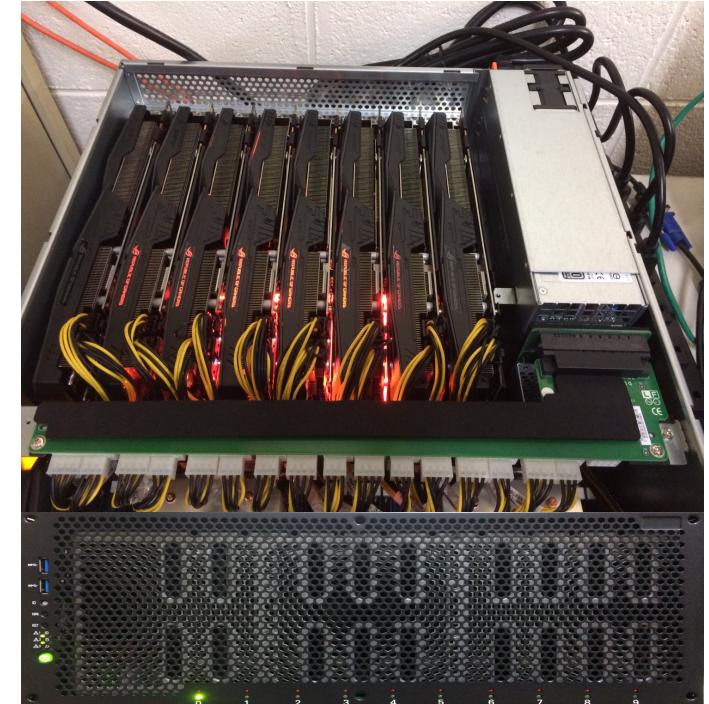
Yuan-Fu Liao

National Taipei University of Technology

# Reference Hardware

- GPU Server (~\$600K)
  - Intel Xeon E5-2620V4 <8C/2.1GHz> \*2
  - [Tyan FT77C-B7079 4U 8 Bay](#)
  - 128GB (32G DDR4 2400 288P ECC RDIMM \*4)
  - [ASUS ROG Strix GeForce GTX 1080](#) \* 8
- PC (~\$150K)
  - INTEL 英特爾 Core i7-6850K
  - ASUS 華碩 ROG STRIX X99 C
  - KINGSTON 16G DDR4-2133 CL15N15D8/16 \*8
  - [ASUS ROG Strix GeForce GTX 1080](#) \* 2
  - Seasonic 850W/ATX X-Series X-850W 電源供應器

工作用



# Reference Hardware

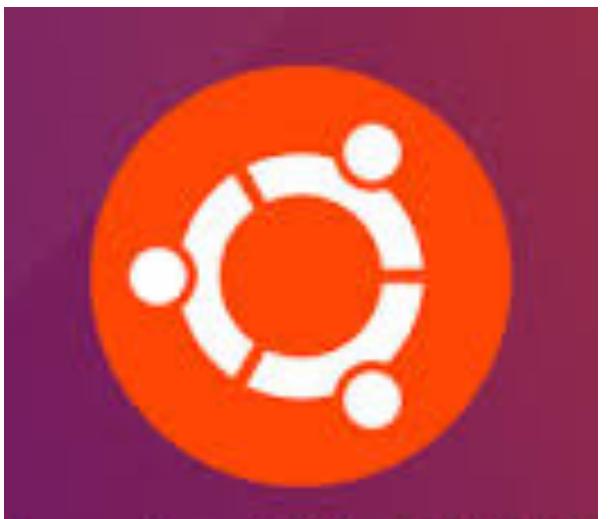
- PC
  - Mac mini
- Notebook
  - Macbook Pro 15 2012
    - NVIDIA GeForce GT 650M
  - Macbook 12
  - Surface Pro 4



# Reference Software

- Ubuntu with OpenBLAS, ATLAS (recommend)
- Kaldi (ASR)
  - <https://github.com/kaldi-asr/kaldi.git>
- CUDA (GPU)
  - Nvidia GPU card Driver
  - Nvidia libcudnn
- Gridengine (Parallel Computing)
  - run.pl vs. queue.pl

# Typical Working Environment



```
root@MIPC:/usr/local/lib/python2.7/dist-packages/tensorflow
Successfully downloaded train-images-idx3-ubyte.gz 991
Successfully downloaded train-labels-idx1-ubyte.gz 288
Successfully downloaded t10k-images-idx3-ubyte.gz 1648
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542
Extracting data/train-images-idx3-ubyte.gz
Extracting data/train-labels-idx1-ubyte.gz
Extracting data/t10k-images-idx3-ubyte.gz
Extracting data/t10k-labels-idx1-ubyte.gz
Initialized!
Step 0 (epoch 0.00), 5.5 ms
Minibatch loss: 12.054, learning rate: 0.010000
Minibatch error: 90.6%
Validation error: 84.6%
Step 100 (epoch 0.12), 361.9 ms
Minibatch loss: 3.289, learning rate: 0.010000
Minibatch error: 6.2%
Validation error: 7.0%
```

NVIDIA-SMI 367.57							Driver Version: 367.57	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC		
							Fan	Temp
							Perf	Pwr:Usage/Cap
							Memory-Usage	GPU-Util
							Compute M.	
0	GeForce GTX 1080	On	PCI\0000:00:04.0	Off		N/A	54% 71C	P2 11W / 198W   355MiB / 8113MiB   53% E. Process
1	GeForce GTX 1080	On	PCI\0000:00:05.0	Off		N/A	49% 66C	P2 134W / 198W   355MiB / 8113MiB   49% E. Process
2	GeForce GTX 1080	On	PCI\0000:00:08.0	Off		N/A	33% 55C	P2 126W / 198W   355MiB / 8113MiB   58% E. Process
3	GeForce GTX 1080	On	PCI\0000:00:09.0	Off		N/A	50% 68C	P2 139W / 198W   355MiB / 8113MiB   52% E. Process
4	GeForce GTX 1080	On	PCI\0000:00:0B.0	Off		N/A	46% 64C	P2 132W / 198W   355MiB / 8113MiB   61% E. Process
5	GeForce GTX 1080	On	PCI\0000:00:0C.0	Off		N/A	45% 64C	P2 99W / 198W   355MiB / 8113MiB   45% E. Process
6	GeForce GTX 1080	On	PCI\0000:00:0D.0	Off		N/A	45% 64C	P2 109W / 198W   355MiB / 8113MiB   56% E. Process
7	GeForce GTX 1080	On	PCI\0000:00:0E.0	Off		N/A	52% 70C	P2 132W / 198W   355MiB / 8113MiB   51% E. Process

Processes:				GPU Memory Usage
GPU	PID	Type	Process name	
0	28496	C	nnet-train-simple	353MiB
1	28514	C	nnet-train-simple	353MiB
2	28500	C	nnet-train-simple	353MiB
3	28505	C	nnet-train-simple	353MiB
4	28491	C	nnet-train-simple	353MiB
5	28510	C	nnet-train-simple	353MiB
6	28506	C	nnet-train-simple	353MiB
7	28512	C	nnet-train-simple	353MiB

```
top - 22:44:53 up 19 days, 9:45, 12 users, load average: 31.41, 32.37, 33.25
Tasks: 1023 total, 32 running, 981 sleeping, 0 stopped, 10 zombie
%Cpu(s): 94.5 us, 1.0 sy, 0.0 ni, 4.3 id, 0.1 wa, 0.0 hi, 0.1 si, 0.0 st
KiB Mem : 13191096+total, 21628712 free, 30748724 used, 79533528 buff/cache
KiB Swap: 25000038+total, 24992227+free, 78108 used. 10030736+avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
20114	speech	20	0	2225036	2.086g	9348	R	100.0	1.7	10:30.38	gmm-latgen-f+
15078	speech	20	0	279528	130432	14632	R	100.0	0.1	85:52.98	nnet-align-c+
15113	speech	20	0	266116	117172	14520	R	100.0	0.1	87:32.72	nnet-align-c+
15144	speech	20	0	270460	121572	14644	R	100.0	0.1	87:10.23	nnet-align-c+
15032	speech	20	0	278136	128516	14632	R	100.0	0.1	86:11.36	nnet-align-c+
15038	speech	20	0	275312	126508	14728	R	100.0	0.1	86:50.96	nnet-align-c+
15066	speech	20	0	259288	109868	14848	R	100.0	0.1	87:08.35	nnet-align-c+
15090	speech	20	0	257616	108688	14600	R	100.0	0.1	87:24.89	nnet-align-c+
15093	speech	20	0	259576	110468	14376	R	100.0	0.1	86:57.67	nnet-align-c+
15101	speech	20	0	281564	132468	14488	R	100.0	0.1	86:22.17	nnet-align-c+
15123	speech	20	0	246884	97940	14640	R	100.0	0.1	86:54.85	nnet-align-c+
15129	speech	20	0	271056	121580	14700	R	100.0	0.1	86:30.28	nnet-align-c+
15139	speech	20	0	261892	113164	14736	R	100.0	0.1	87:11.43	nnet-align-c+
20106	speech	20	0	2214440	2.076g	9460	R	100.0	1.7	10:30.63	gmm-latgen-f+
20113	speech	20	0	2205308	2.067g	9312	R	100.0	1.6	10:29.28	gmm-latgen-f+
14334	speech	20	0	5009972	4.728g	6788	R	99.7	3.8	28:57.46	fstpushspeci+
15042	speech	20	0	283540	134644	14644	R	99.7	0.1	86:33.24	nnet-align-c+
15055	speech	20	0	258528	109620	14700	R	99.7	0.1	87:30.18	nnet-align-c+
15115	speech	20	0	252004	102640	14688	R	99.7	0.1	87:13.22	nnet-align-c+
15140	speech	20	0	269300	120448	14604	R	99.7	0.1	86:49.55	nnet-align-c+
15142	speech	20	0	282608	133796	14736	R	99.7	0.1	86:31.82	nnet-align-c+
15150	speech	20	0	279204	132248	14552	R	99.7	0.1	86:18.23	nnet-align-c+
20109	speech	20	0	2271712	2.131g	9404	R	99.7	1.7	10:29.75	gmm-latgen-f+
15079	speech	20	0	279444	130512	14772	R	99.3	0.1	86:42.51	nnet-align-c+
15084	speech	20	0	265964	117212	14756	R	99.0	0.1	86:48.35	nnet-align-c+
15133	speech	20	0	265972	117008	14628	R	99.0	0.1	87:06.88	nnet-align-c+
20112	speech	20	0	2222192	2.083g	9460	R	99.0	1.7	10:29.42	gmm-latgen-f+
15120	speech	20	0	272288	122728	14616	R	98.3	0.1	87:10.21	nnet-align-c+
15147	speech	20	0	269900	120968	14536	R	97.7	0.1	86:58.97	nnet-align-c+
18488	speech	20	0	270236	116792	20020	R	92.1	0.1	0:03.40	fstrmepslon
18487	speech	20	0	3125928	2.815g	20068	R	62.0	2.2	9:03.58	fstcompile
20155	root	20	0	0	0	0	S	5.9	0.0	0:14.86	kworker/2:3
18278	root	20	0	0	0	0	S	3.3	0.0	0:59.84	kworker/10:3
18473	speech	20	0	26540	4436	3164	S	2.0	0.0	0:22.30	htop
18479	root	20	0	0	0	0	S	1.3	0.0	0:20.72	kworker/10:1
8053	liao	20	0	43008	4644	2948	R	1.0	0.0	1:24.64	top
14904	root	20	0	0	0	0	S	0.7	0.0	23:12.19	nfssd
8	root	20	0	0	0	0	S	0.3	0.0	21:16.77	rcu_sched
4420	root	20	0	0	0	0	S	0.3	0.0	0:09.43	kworker/16:1
14896	root	20	0	0	0	0	S	0.3	0.0	0:38.35	nfssd
14899	root	20	0	0	0	0	S	0.3	0.0	1:16.02	nfssd
14901	root	20	0	0	0	0	S	0.3	0.0	2:13.01	nfssd
14902	root	20	0	0	0	0	S	0.3	0.0	3:38.65	nfssd
14903	root	20	0	0	0	0	S	0.3	0.0	9:10.88	nfssd
15258	speech	20	0	35292	6040	5200	S	0.3	0.0	0:01.86	splice-feats
15266	speech	20	0	35804	7092	6004	S	0.3	0.0	0:04.16	transform-fe+
15267	speech	20	0	36376	7384	5824	S	0.3	0.0	0:03.37	transform-fe+
15280	speech	20	0	35652	6992	5952	S	0.3	0.0	0:04.22	transform-fe+
15293	speech	20	0	35364	5908	5108	S	0.3	0.0	0:01.80	splice-feats

# How to install Kaldi

- Assume install in your home directory

```
$cd ~  
$ git clone https://github.com/kaldi-asr/kaldi.git  
$ cd ~/kaldi/tools  
    ./extras/check_dependencies.sh  
    make  
    ./install_srilm.sh (download srilm.tgz first)  
    ./extras/install_kaldi_lm.sh  
$cd ~/kaldi/src  
    configure --static  
    make depend  
    make
```

# Tools that require manual download

- Most dependency tools are in the tools directory and are downloaded and installed automatically
- However, some tools require manual download due to license issues
- SRILM installation example
  - Get SRILM source code from <http://www.speech.sri.com/projects/srilm/>
  - Put it in ~/kaldi/tools

```
$ mv srilm.tgz ~/kaldi/tools
```

- Install using Kaldi's installer

```
$ cd ~/kaldi/tools  
$ ./install_srilm.sh
```

# Test kaldi installation

```
$ cd ~/kaldi/egs/yesno/s5
```

```
$ ./run.sh
```

```
%WER 0.00 [ 0 / 232, 0 in , 0 del, 0 ub ] exp/mono0a/decode_te t_ye no/wer_11_1.0
```

# [optional] How to install Python

- Install Anaconda
  - Follow the instructions on the [Anaconda download site](https://www.continuum.io/downloads) →  
<https://www.continuum.io/downloads>
- Create a conda environment
  - *# Python 2.7 (or 3.5)*
  - *\$ conda create -n py27 python=2.7 ipykernel*
- Activate the conda environment
  - *\$ source activate py27*
  - *\$ conda install spyder notebook*

Remark: Deactivate the conda environment

- *\$ source deactivate*

## [Using both Python 2.x and Python 3.x in IPython Notebook](#)

**# Automatically registering kernels**  
\$ conda create -n py27 python=2.7 ipykernel  
\$ conda create -n py36 python=3.6 ipykernel

**#Manually registering kernels**  
\$ conda create -n py35 python=3.5  
\$ source activate py35  
\$ conda install notebook ipykernel  
\$ ipython kernel install --user

# [optional] How to install TensorFlow

- See <https://www.tensorflow.org/install>
- *pip install*

```
# Current release for CPU-only
pip install tensorflow
# GPU package for CUDA-enabled GPU cards
pip install tensorflow-gpu
```

- *Testing your installation*

```
(py35) yfliao@mini3:~$ python
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  2 2016, 17:52:12)
[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow
>>>
```

# [optional] How to install Keras

- See <https://keras.io/#installation>

```
$ conda install -c conda-forge keras
```

- *pip install*

```
$ pip install keras
```

- *Testing your installation*

```
(py35) yfliao@mini3:~$ python
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  2 2016, 17:52:12)
[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import keras
Using TensorFlow backend.
>>>
```

- To figure out what is the current backend type

```
(py35) yfliao@mini3:~$ python -c "from keras import backend;
print(backend._BACKEND)"
Using TensorFlow backend.
tensorflow
(py35) yfliao@mini3:~$
```

# [optional] How to install GridEngine

- **Install GridEngine**
  - sudo apt-get install gridengine-master gridengine-client
  - sudo apt-get install gridengine-client gridengine-exec
- **Configuring GridEngine**
  - [http://kaldi-asr.org/doc/queue.html#parallelization\\_gridengine](http://kaldi-asr.org/doc/queue.html#parallelization_gridengine)

# Grid Engine Component Architecture

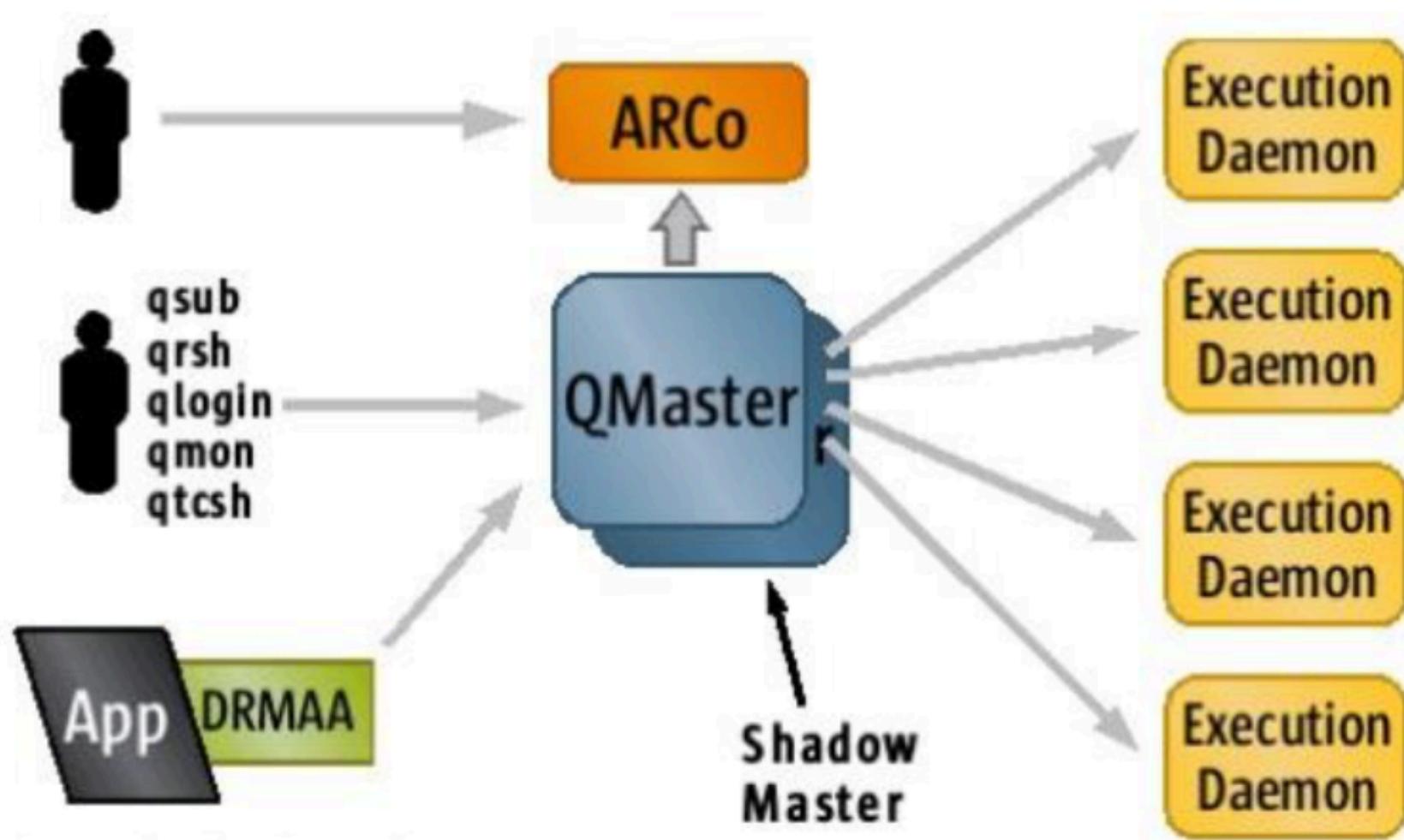


Figure 1: Sun Grid Engine Component Architecture

# Grid Engine Basics and Concepts

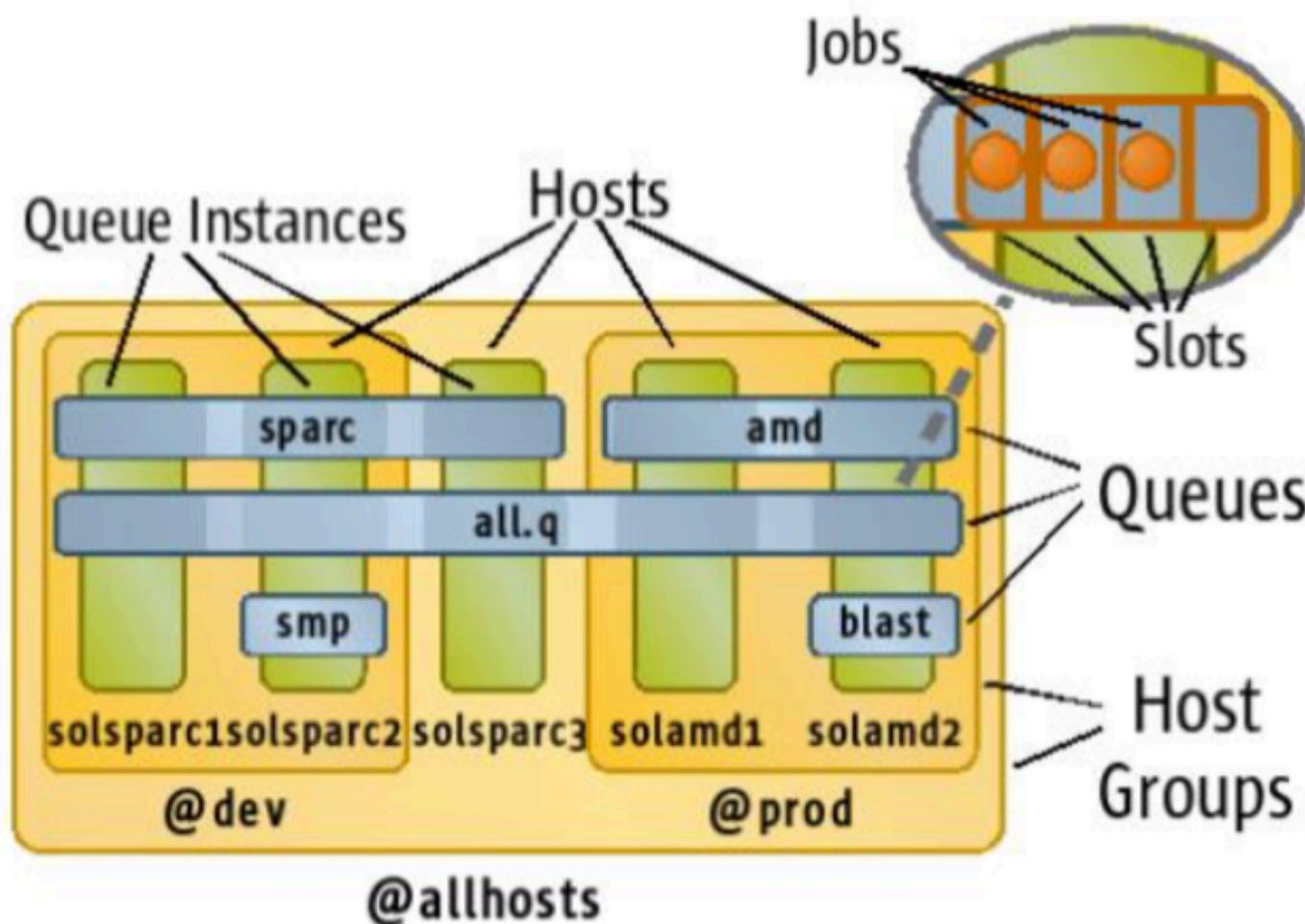
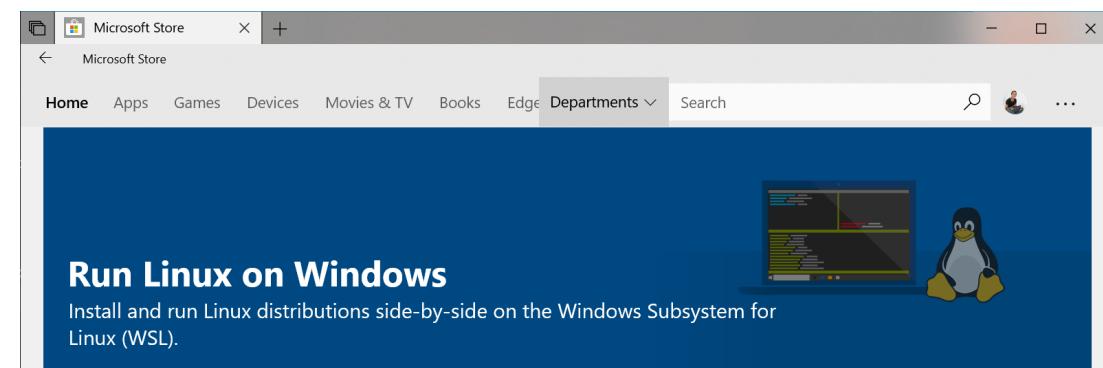
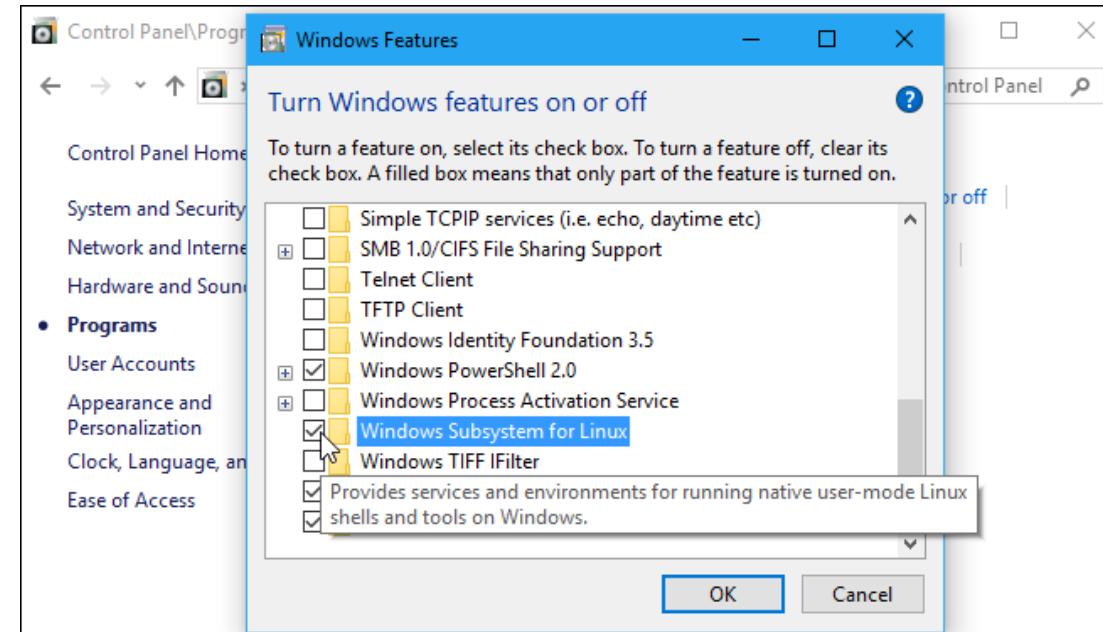
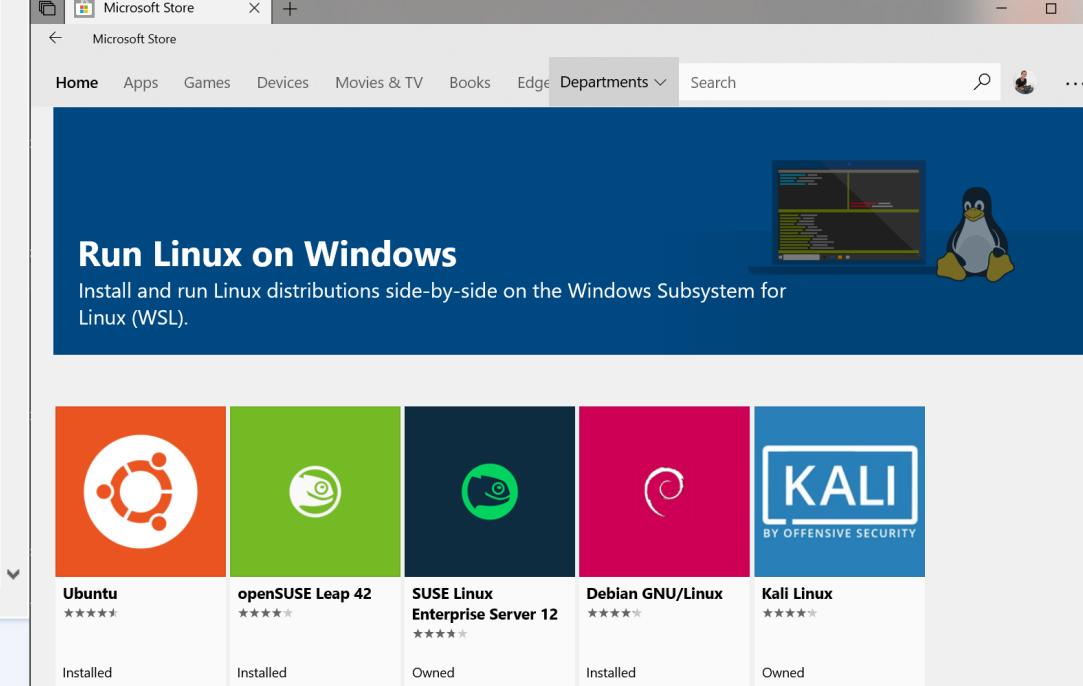
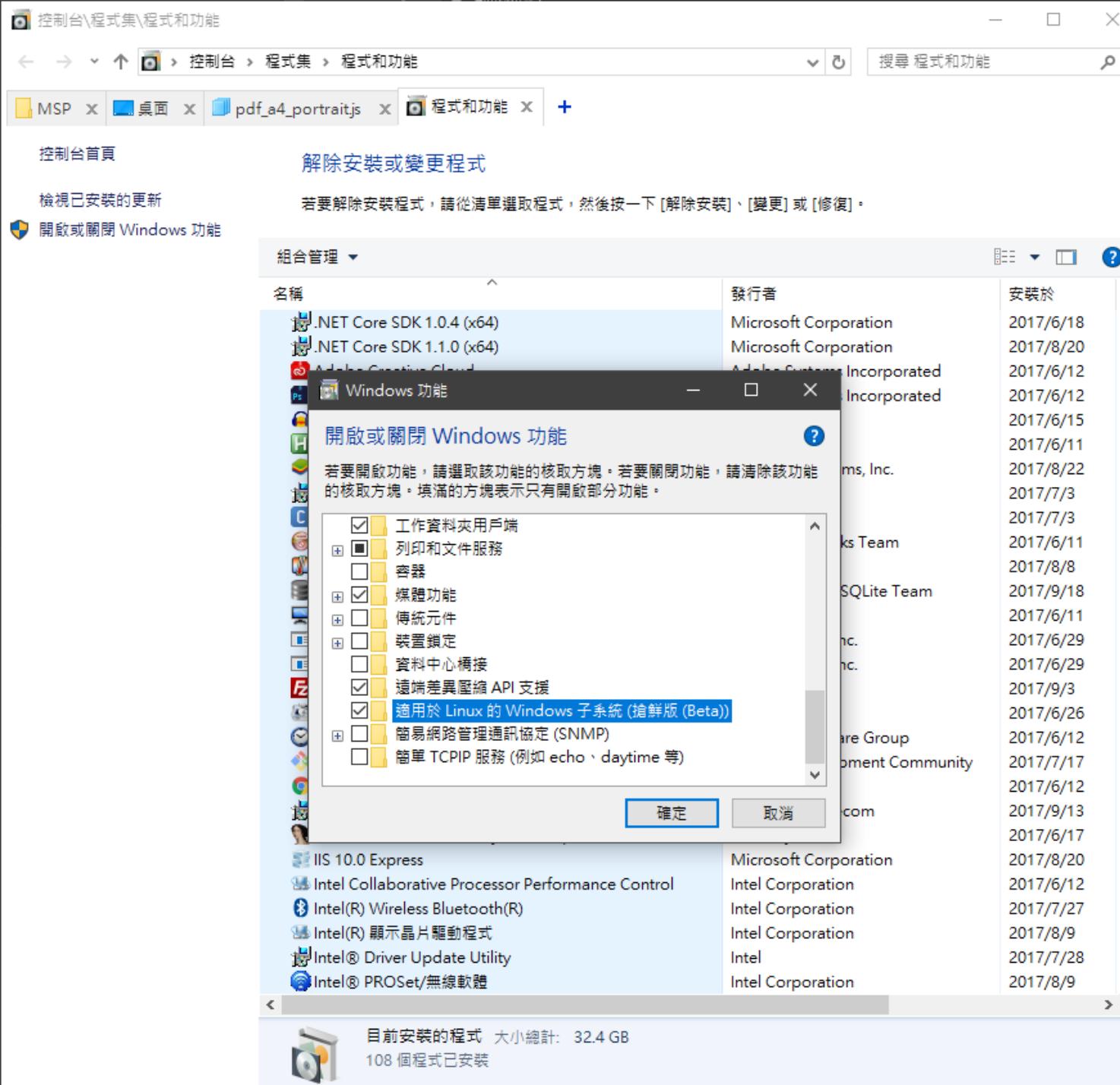


Figure 2: Example Host Group and Queue Configuration

# Run Linux on Windows 10

- Windows Subsystem
  - Bash on Ubuntu on Windows
    - Control Panel > Programs > Turn Windows Features On Or Off
      - Click the “OK” button
    - Microsoft Store
      - search for “Linux” in the store
      - “Get the apps”





# Google Colaboratory

- <https://colab.research.google.com>

使用限制及提供的軟硬體規格

- 1) Support Python 2&3 開發環境
- 2) 提供non GPU 及 GPU 運算模式
- 3) 最長連接時間為12小時，當超過12小時時，需重新點擊才能再次工作
- 4) 免費Tesla K80 GPU、50G HD使用空間、12G RAM



Drive



Search Drive



New

▶ My Drive

▶ Computers

Shared with me

Recent

Starred

Trash

Backups

Storage

1.3 GB of 15 GB used

UPGRADE STORAGE

My Drive &gt; colab ▾

New folder

Upload files

Upload folder

Google Docs &gt;

Google Sheets &gt;

Google Slides &gt;

More &gt;

Drop files here

or use the "New" button.

colab X

Details

Activity

Today

8:34 PM



You created an item in

Google Forms &gt;

Google Drawings &gt;

Google My Maps &gt;

Google Sites &gt;

Google Jamboard &gt;

+ Connect more apps



Drive



Search Drive



New



My Drive



Computers



Shared with me



Recent



Starred



Trash



Backups



Storage

1.3 GB o

UPGRADE

## Connect apps to Drive

All



colab

**Colaboratory**offered by <https://colab.research.google.com>

A data analysis tool that combines code, output, and descriptive text into one collaborative document.

**+ CONNECT**

Productivity

★★★★★ (651)





+ Code + Text

Table of contents Code snippets Files X

UPLOAD REFRESH MOUNT DRIVE

..

sample\_data

## Virtual Machine

### Hardware and Software

↳ 6 cells hidden

### Requirement

↳ 2 cells hidden

### Google Drive

↳ 3 cells hidden

### Kaldi Toolkit

↳ 21 cells hidden

### Training Recipies

↳ 8 cells hidden

### Backup

↳ 1 cell hidden

# Notebook settings

Runtime type

Python 3

Hardware accelerator

GPU



Omit code cell

None

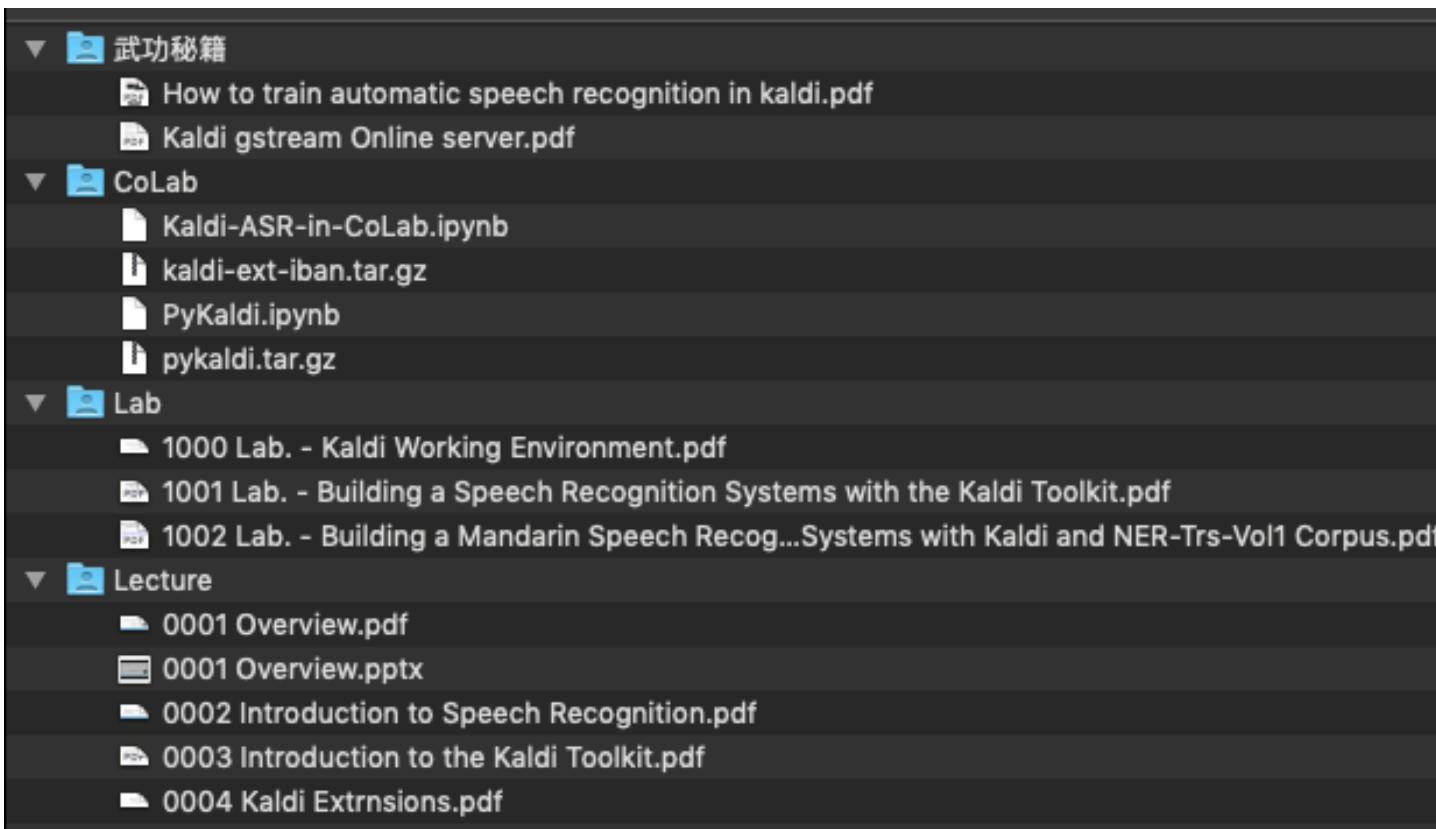
GPU

on saving this notebook

CANCEL

# Example (1/2)

- Kaldi
  - [https://drive.google.com/drive/folders/1oL\\_qoZDW3Wy\\_zoT3NOw0qCvaVojFwghF](https://drive.google.com/drive/folders/1oL_qoZDW3Wy_zoT3NOw0qCvaVojFwghF)



<https://reurl.cc/lLobEI>



## Example (2/2)

- End-to-End ASR (宋子維@UCSD, @NTU)
  - <https://colab.research.google.com/drive/1TkwWZXJuX9kAN5dsFuOuMurXiJFME4L2>
- End-to-End TTS (杜濤@NTU)
  - <https://colab.research.google.com/drive/1Cr4BC9zNayEHy8fyqH2wG-uhnhEs7jwk>