

How to use GPU servers for UG students

1. Create your CSD account if you not have:
https://cssystem.cse.ust.hk/UGuides/hkust_only/activation.html
2. ssh to the ugcpu sever:
ssh <your_itsc>@ugcpu4.cse.ust.hk / ssh <your_itsc>@ugcpu5.cse.ust.hk
Password: your password
3. cd /localdata
4. mkdir < your_own_folder>
5. command setenv PYTHONUSERBASE /localdata/<your_own_folder>
6. Whenever you want to install new packages:
pip install -user

Notice:

1. other people cannot access <your_own_folder>
2. always need to do the step 4 whenever you log in the server
3. space on /localdata is only a few hundred GBs, not very suitable for video ML work
4. If you want to replicate the folder to ugcpu5 sever:
cd /localdata; tar cf - <your_own_folder> | ssh ugcpu5 "(cd /localdata; tar -xf -)"
5. To allow group access and read, execute:chmod g+rx <your_own_foler>
6. Contact Henry if any help is needed:
henry@cse.ust.hk
7. Bitwise is highly recommended for ssh and SFTP services. You will know why once you downloaded it.
8. Add anything here if you think is helpful to others

Using Servers

- Why GPU Servers?
 - speed up your experiments
- CSE UG has $4 + 4 = 8$ GPUs available
 - `ugcpu4.cse.ust.hk`
 - `ugcpu5.cse.ust.hk`

Using Servers: Create An Account

- Create your CSD account if you don't have
 - https://cssystem.cse.ust.hk/UGuides/hkust_only/activation.html
- Check if you have any problem:
 - https://course.cse.ust.hk/comp4471/Web/UG_GPU.pdf
 - if you still fails, contact CSE System Office

Using Servers: Log In & Set Up

- Logging in with `$ssh <your itsc>@ugcpu4.cse.ust.hk`

```
[eishii@ishiietsukonoMacBook-Pro ~ % ssh eishii@ugcpu5.cse.ust.hk
[Password:
Last login: Fri Mar 20 12:32:00 2020 from wf124-204.ust.hk
=====
This is ugcpu5.cse.ust.hk .

If you need to log on this particular host, you may consider
logging on by:

ssh ugcpu5.cse.ust.hk
=====
ugcpu5:eishii:19> █
```

- Set up your directory & python

```
[ugcpu5:eishii:9> cd /localdata/
[ugcpu5:eishii:10> mkdir eishii
[ugcpu5:eishii:19> setenv PYTHONUSERBASE /localdata/eishii
[ugcpu5:eishii:20> echo $PYTHONUSERBASE
/localdata/eishii
```

- Install everything with `$pip install --user [package]`

```
ugcpu5:eishii:19> pip install --user numpy
Requirement already satisfied: numpy in /usr/local/lib/site_python2
```

Using Servers: Do's and Don'ts

Don'ts

- Occupy many GPUs / for too long time
 - 8 GPUs for 120+ students!
- Occupy many memories
 - only total ~300GB memory capacity under /localdata

Do's

- Consider others & share resources!
- Work inside /localdata/<your_own_folder>
 - always \$cd /localdata/<your_own_folder> after logging in and then work there

Using Servers: Useful Commands

- `nvidia-smi`
 - check the status of GPUs

```
[ugcpu5:eishii:21> nvidia-smi
Fri Mar 20 14:07:38 2020
```

NVIDIA-SMI 440.64.00 Driver Version: 440.64.00 CUDA Version: 10.2									
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	RTX	208...	Off	00000000:07:00.0	Off		N/A	
27%	33C	P0	53W / 250W		0MiB / 11019MiB		1%	Default	

1	GeForce	RTX	208...	Off	00000000:0B:00.0	Off		N/A	
28%	38C	P0	56W / 250W		0MiB / 11019MiB		1%	Default	

2	GeForce	RTX	208...	Off	00000000:0F:00.0	Off		N/A	
32%	36C	P0	65W / 250W		0MiB / 11019MiB		0%	Default	

3	GeForce	RTX	208...	Off	00000000:13:00.0	Off		N/A	
40%	33C	P0	22W / 250W		0MiB / 11019MiB		0%	Default	

Processes:					GPU Memory
GPU	PID	Type	Process name		Usage
=====					
No running processes found					

Using Servers: Useful Commands

- nvidia-smi
 - check the status of GPUs

NVIDIA-SMI 418.67				Driver Version: 418.67				CUDA Version: 10.1			
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 108...	Off	00000000:04:00.0	Off			N/A				
0%	37C	P2	69W / 260W	4721MiB / 11178MiB	28%		Default				
1	GeForce GTX 108...	Off	00000000:05:00.0	Off			N/A				
0%	39C	P2	63W / 260W	5581MiB / 11178MiB	27%		Default				
2	GeForce GTX 108...	Off	00000000:08:00.0	Off			N/A				
0%	45C	P2	76W / 260W	5471MiB / 11178MiB	69%		Default				
3	GeForce GTX 108...	Off	00000000:09:00.0	Off			N/A				
0%	43C	P2	66W / 260W	3897MiB / 11178MiB	23%		Default				
4	GeForce GTX 108...	Off	00000000:85:00.0	Off			N/A				
0%	41C	P2	66W / 260W	6799MiB / 11178MiB	26%		Default				
5	GeForce GTX 108...	Off	00000000:86:00.0	Off			N/A				
0%	23C	P8	8W / 260W	4110MiB / 11178MiB	0%		Default				
6	GeForce GTX 108...	Off	00000000:89:00.0	Off			N/A				
0%	24C	P8	8W / 260W	4128MiB / 11178MiB	0%		Default				
7	GeForce GTX 108...	Off	00000000:8A:00.0	Off			N/A				
0%	44C	P2	83W / 260W	1139MiB / 11178MiB	87%		Default				

Processes:				GPU Memory	
GPU	PID	Type	Process name	Usage	
0	15744	C	python	4711MiB	
1	15745	C	python	5571MiB	
2	15748	C	python	5461MiB	
3	7950	C	python	3887MiB	
4	15749	C	python	6789MiB	
5	3137	C	python	1891MiB	
5	3347	C	python	2209MiB	
6	3173	C	python	2375MiB	
6	3383	C	python	1743MiB	
7	25027	C	python	1129MiB	

Using Servers: Useful Commands

- `nvidia-smi`
 - check the status of GPUs
- `tmux`
 - protect running programs on a remote server from connection drops (or `screen` cmd)
 - more detail: <https://github.com/tmux/tmux/wiki/Getting-Started>
- `chmod g+rx <your_own_folder>`
 - give access to your folder to others
 - otherwise the others cannot access `/localdata/<your_own_folder>`

Using Servers: Useful Resources

- Using TensorFlow/PyTorch on CSE server
 - <https://cssystem.cse.ust.hk/UGuides/python/index.html>
 - TensorFlow, PyTorch, and some other common packages are already installed in CSE server

```
/usr/local/bin/python-csd
```

We also provide module-based Python 3 support, by "/usr/local/bin/python-csd". We note frequent requests of installing Tensorflow, PyTorch, etc..., while those packages are generally bulky, commonly requested, and require a lot of time for building. And sometimes, different users may have different need for a specific version.

To cope with this problem, we have maintain some modules with different versions so users may pick the version they need explicitly.

You may 'cd' to /usr/local/packages/python/setup. 'Source' the setup file for your need. For example, if you wish to use Tensorflow 1.7, you can do so:

If you are a csh/tcsh user:

```
cd /usr/local/packages/python/setup
source tensorflow-1.7.cshrc
```

If you are a bash user:

```
cd /usr/local/packages/python/setup
source tensorflow-1.7.profile
```

P.S. : to tell which SHELL you are using, type: "echo \$SHELL".

Using Servers: Alternatives

- Google Colaboratory (Colab)
 - strongly recommended for final project
 - online platform to run python provided by google!
 - promo: <https://www.youtube.com/watch?v=inN8seMm7UI>
 - FREE GPU, easy to use, easy to share
 - useful tutorials:
 - https://colab.research.google.com/notebooks/basic_features_overview.ipynb
 - <https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/quickstart/beginner.ipynb>
 - <https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/quickstart/advanced.ipynb>