Some Notes on How to Use ICCluster

Begins

- Login the server by the following command. Replace `<username>` with your ISA login name and use your EPFL credentials. You can test on `iccluster058` before `iccluster135` is setup.

```
ssh <username>@iccluster135.iccluster.epfl.ch
```

- Use `nvidia-smi` to check the GPU use.

```
+-----------------------------------------------------------------------------+
| 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  TITAN Xp COLLEC...  Off  | 00000000:05:00.0 Off |                  N/A |
| 23%   29C    P8    10W / 250W |     10MiB / 12196MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  TITAN Xp COLLEC...  Off  | 00000000:09:00.0 Off |                  N/A |
|  0%   92C    P2   152W / 250W |   6631MiB / 12196MiB |     99%      Default |
+-------------------------------+----------------------+----------------------+
|   2  TITAN Xp COLLEC...  Off  | 00000000:85:00.0 Off |                  N/A |
| 23%   26C    P8     9W / 250W |     10MiB / 12196MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  TITAN Xp COLLEC...  Off  | 00000000:89:00.0 Off |                  N/A |
|  0%   24C    P8     8W / 250W |   4697MiB / 12196MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    1     39873      C   python                                      6621MiB |
|    3     39964      C   python3                                     4687MiB |
+-----------------------------------------------------------------------------+
```

For the example above, the output shows the machine has 4 TITAN XP GPUs and the CUDA version is 10.1. In addition, process 39873 is using GPU 1; it consumes 6621 MB memory and 99% computational utility. Process 39964 is using GPU 3; it consumes 4687 MB memory but no computational utility. Process 39964 seems to be a process hanged and should be avoided.

- Use `htop` or `top` to track the existing processes and monitor CPU use. There are 48 CPUs on both note `058` and `135`. Use `q` to quit the monitoring process. You can find more materials about the usage of `htop` or `top` online.

- Each two teams are allocated one GPU (we will later specify). When you run a program requiring GPU devices, e.g., ones in PyTorch or TensorFlow, you should specify the GPU you want to use **in the beginning of your main function**, otherwise your programme will claim **all available resources on this machine**. Typically you can use the following codes to make only one GPU visible to your programme. Note that *string* (not integer) `"1"` is the GPU ID you want to use.

```python
import os
os.environ['CUDA_VISIBLE_DEVICES'] = '1'
print('This programme is using GPU 1')
```

`screen` Command

- Normally, anything you are running on the termnial of the cluster will be stopped when you disconnect your device with the cluster. For programmes that need long time to finish, you can use `screen` command.

- Common operations of `screen` are listed below. Use `ctrl+A` plus `ctrl+D` to detach a screen. A screen will be detached and deleted when you type `exit` inside the screen.

```
screen -S <NAME>   # create a screen, replace <NAME> with the actual name
screen -r <NAME>   # attach to the screen named <NAME>
screen -list       # list the active screen
```

- Note that inside the screen, **you cannot scroll up to view more history beyond the current screen**. When the data printed in the command line goes beyond the buffer, it is gone. **Always save your results in files instead of just printing it out in the command line**.

- You can find more operations of command `screen` and examples in https://www.geeksforgeeks.org/screen-command-in-linux-with-examples/.

Jupyternotebook

- run `ifconfig` to check the IP address of the Ethernet.
- run `jupyter notebook --ip <IP>` to start a python notebook in the current directory. You can then open the notebook in a browser by the URL provided.
- Click the `new` tag to create a python notebook and start writing your code.
- More information available on https://jupyter-notebook.readthedocs.io/en/stable/notebook.html.