User guide for GPU servers with Pytorch

Equipe IMAGeS

Connection to servers and virtual environment

First of all, you need an account in order to access servers. You'll have to email Stéphane Prunière at `pruniere@unistra.fr` asking for a new account.

There are two servers. You can now access with this command :

```
$ ssh username@mivgpu1.u-strasbg.fr
```

or for the second server

```
$ ssh username@mivgpu2.u-strasbg.fr
```

Now, you have to create a Python Virtual Environment to install all the package you need. We did a script `installer.sh` (delivered with this guide) that you only have to run in the directory you want to run your environment with :

```
$ sh installer.sh
```

This script installs the environement, the main package you'll need to work, and check if you can work on GPU (cuda).

But you can also do it manually with the following commands :

```
$ virtualenv -p python3 env     #Set up virtual environment
$ source env/bin/activate       #Activate virtual environment,
                                #You can install what
                                #you want and work on it

(env) $ pip install numpy             #Some useful packages
(env) $ pip install scipy
(env) $ pip install torch
(env) $ pip install torchvision
(env) $ pip install scikit-image
(env) $ pip install scikit learn
(env) $ pip install h5py
```

We will now verify that all works properly with this command :

```
(env) $ which python
```

Which will normally return something like `installation_path/env/bin/python`
It's time to check if Cuda is properly working in order to work on GPU's :

```
(env) $ python
>>> import torch
>>> torch.cuda.is_available()
```

If it's return `True` , you're ready to run your code on GPU's !

Important Advices

Most of the time, there will be several people working on GPU's, so you need to share resources. As we are doing this guide, there are 4 GPU's on each servers. Before runing your code, and to avoid Out Of Memory issues, we advice to run this command :

```
$ nvidia-smi -l 1
```

It will give you in real-time a lot of informations about GPU's, mostly their Memory-Usage. You can then choose in what GPU's you'll run your code
So, we strongly advice to put at the beginning of your code something like that (be careful, GPU's are 0-indexed) :

```
import torch
gpu0 = 'cuda:0'  #For work on GPU number 0
gpu2 = 'cuda:2'  #For work on GPU number 2

#When you create a tensor, you can do
t1 = torch.randn(2,5,7 device=gpu0)  #One way to work on GPU
t2 = torch.randn(2,5,7).to(gpu2)     #An other way
```

You're now ok !

Some useful commands

You will need to tranfer a lot of files between your terminal and the server. In order to do that, you can use the `scp` command like that :

- If you want to transfer a single file from your terminal to the server, run this command on your machine :

```
$ scp file\_to\_transfer username@icube-images-gpu1:/path
```

- If you want to tranfer a folder from your terminal to the server, run this command on your machine :

  ```
  $ scp -r folder\_to\_transfer username@icube-images-gpu1:/path
  ```

That's basically the same principle if you want to transfer from the server to your computer.

We can also use the following commands :

```
$ ps -aux
$ w
```

In order to see who's connected to the server and what processus they are running, so you can better share GPU's together.

Hope this will help you !