Guide to
High-performance Computing

Nicklas Hansen

What is the HPC?

The High-performance Computing (HPC) cluster is a computing cluster available to all students enrolled at DTU. It offers remote execution of scripts via a queue system and can handle very resource-intensive tasks such as training deep neural networks due to its powerful CPUs and GPUs in particular. For deep learning, we are mostly interested in the GPU nodes.

Connecting to the HPC

You can connect to DTU's HPC cluster either by terminal or by a graphical user interface. Terminal connection is faster but graphical interface is easier, so use whatever you are more comfortable with. To connect by SSH through terminal, use the following command:

```
ssh <student ID>@login2.gbar.dtu.dk
```

Note that an SSH connection may require a locally stored SSH key. If you run into issues for not having an SSH key already, refer to this link for a guide on how to generate SSH key pairs. If you prefer the graphical user interface, refer to this link for a guide on how to connect to the cluster through the ThinLinc client (requires a stable internet connection).

Installing dependencies

Now that you're connected to the HPC, you have access to UNIX commands and a remote user directory linked to your account (i.e. it is persistent). To run Python and GPU-accelerated scripts, you can load pre-installed *modules* with the command below. You will need to make sure that your versions of Python, CUDA and CUDNN are compatible. I recommend the following modules:

```
module load python3/3.6.2
module load cuda/8.0
module load cudnn/v7.0-prod-cuda8
module load ffmpeg/4.2.2
```

You can call

```
module avail
```

to get an overview of other available modules and/or versions.

If you want to avoid loading modules every time you connect to the HPC, you can add the commands above to your **.bashrc** file located in the root directory using your favorite editor, e.g.:

```
nano ~/.bashrc
```

After making changes to your .bashrc, you can source it to re-execute:

```
source ~/.bashrc
```

To install PyTorch and any other dependencies that you might need for your project, use the following command:

```
pip3 install --user torch torchvision matplotlib seaborn
```

This will install the pytorch, torchvision, matplotlib, and seaborn packages. Depending on your project, you may need additional pip packages.
Pip installs are persistent and you will only need to install packages once. Don't forget the 3 in pip as you will otherwise install packages for Python 2.x. *Note that you should avoid conda on the HPC due to compatibility issues. Good alternatives are miniconda or virtualenv*.

Transferring files to the HPC

You can use Rsync (UNIX only), git or cloud services such as Google Drive or OneDrive to synchronize files between your local machine and the HPC (with Rsync obviously being the faster and more convenient option). A transfer of directory *~/mydir* from your local machine to the remote user directory using Rsync can be executed as follows:

```
rsync -av ~/mydir <student ID>@transfer.gbar.dtu.dk:
```

You will have to use your imagination (or the link above) to transfer in the reverse direction.

Working on a remote codebase

Rather than transferring your codebase from a local machine to the HPC whenever you make any changes, you can work directly on a codebase located remotely on the HPC. Popular IDEs such as Visual Studio Code allow you to connect by SSH to remote machines and work from there for more convenient development. You will also be able to execute and debug code from within the IDE if you prefer.

Using GPUs on the HPC

The HPC cluster uses LSF for resource management. You can use GPUs in two different modes of operation: interactively or by submitting jobs to a queue. **You should only use the interactive nodes for development and debugging purposes** as the number of interactive GPUs is very limited. Use commands *nvidia-smi* and *htop* to view current machine load and please be respectful to other users. Use the *CUDA_VISIBLE_DEVICES*

environment variable to execute code on a specific GPU. You may want to use tmux multiplexing on the interactive nodes. See this page for a guide on how to connect to an interactive node.
The GPU queue is designed for longer jobs running for up to 24 hours. You will primarily be using the queue. To submit a job for the GPU queue, you may want to create a bash script similar to what I have prepared for you here:

```
#!/bin/sh
#BSUB -q gpuv100
#BSUB -gpu "num=1"
#BSUB -J myJob
#BSUB -n 1
#BSUB -W 10:00
#BSUB -R "rusage[mem=32GB]"
#BSUB -o logs/%J.out
#BSUB -e logs/%J.err
<loading of modules, dependencies etc.>
echo "Running script..."
python3 train.py
```

which you can submit using the command:

```
bsub < jobscript.sh
```

After submitting a job to the queue, use **bstat** to review the status of your jobs:

```
bstat
```

For an overview of BSUB and available options, check out this link. When submitting jobs to the queue you need to load modules etc. as part of your bash script. When running on an interactive node, you can put it in your .bashrc.

Using MuJoCo on the HPC

If you're doing a project with me in reinforcement learning, you may need to use MuJoCo for your physics simulations. You can get a free student license here which is valid for one year. Download and place the MuJoco 200 directory as well as your license in ~/.mujoco/ and add the following lines to your .bashrc as well as your bsub job scripts:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/zhome/<your>/<user>/<path>/ \
     .mujoco/mujoco200/bin
export MUJOCO_GL=egl
```

You can get MuJoCo 200 for Linux by the following commands:

```
wget https://www.roboti.us/download/mujoco200_linux.zip
unzip mujoco200_linux.zip
mkdir ~/.mujoco
mv mujoco200_linux ~/.mujoco/
```

You will need to complete these steps before installing e.g. mujoco-py or dm_control.

Questions?

This is by no means a complete guide to the HPC cluster and you will be expected to do some research on your own, especially if you are unfamiliar with UNIX and the terminal. If you have questions or issues related to the HPC, refer to their webpage. If you have questions related to deep learning or MuJoCo on the HPC, feel free to ask on Slack.