Practice session 3: HPC

In this practice session, we will learn how to use the University of Tartu's High Performance Computing Center servers. Later we will use GPUs there to train our big translation models.

Login

To connect to the server, use SSH. If you only have a Windows system, you will have to use PuTTY instead.

Log in using your university username and password:

```
ssh your_username@rocket.hpc.ut.ee
```

Finding your way around

If you are not comfortable with Linux commands, check out this guide: maker.pro/linux/tutorial/basic-linux-commands-for-beginners.

Once you have logged in, you can create a directory where you will keep all the data for your experiments:

```
mkdir mtcourse
```

Move into the new directory:

```
cd mtcourse
```

Create separate directories for your data and scripts:

```
mkdir data
mkdir scripts
```

Sockeye

We will install Sockeye in a Conda virtual environment.

First, load python:

```
module load python-3.6.3
```

Then create a clean environment:

```
conda create -n mtcourse python=3.6.3
```

Activate the environment:

```
source activate mtcourse
```

Install Sockeye for GPU:

```
pip install numpy==1.14.0
wget https://raw.githubusercontent.com/awslabs/sockeye/master/requirements/requirements.gpu-cu90.txt
pip install sockeye --no-deps -r requirements.gpu-cu90.txt
rm requirements.gpu-cu90.txt
```

SLURM

The Rocket cluster uses scheduling system SLURM for running jobs. **DO NOT** execute your commands (truecasing, model training or anything else) directly on the head node! **ALWAYS USE SLURM,** or your access to HPC will be suspended. You will need scripts like this one (it is included in the practice session materials as a separate file):

```
#!/bin/bash

#The name of the job is test_job
#SBATCH -J test_job

#The job requires 1 compute node
#SBATCH -N 1

#The job requires 1 task per node
#SBATCH --ntasks-per-node=1

#The maximum walltime of the job is 5 minutes
#SBATCH -t 00:05:00

#SBATCH --mem=5G

#If you keep the next two lines, you will get an e-mail notification
#whenever something happens to your job (it starts running, completes or fails)
#SBATCH --mail-type=ALL
#SBATCH --mail-user=your_email@here.com

#Keep this line if you need a GPU for your job
#SBATCH --partition=gpu

#Indicates that you need one GPU node
#SBATCH --gres=gpu:tesla:1

#Commands to execute go below

#Load Python
module load python/3.6.3/CUDA-9.0

#Activate your environment
source activate mtcourse

#Display Sockeye's help message
sockeye-train --help
```

Create a separate script for every step of your work. This way, you can always go back and check what you did and which parameters you specified.

Do not ask for GPU nodes when you perform preprocessing steps (truecasing, SentencePiece). It will not make them faster and you will block valuable resources. Only use GPUs for training models.

To send your script to the queue:

```
sbatch path/to/your/script.sh
```

You will see output like:

```
Submitted batch job XXX
```

XXX will be the ID of your job. If you want to cancel it:

```
scancel XXX
```

Once your job starts running, a file named slurm-XXX.out will be created in your current working directory (where you executed sbatch). Your log and output will be written into this file.

Queue

You can view your jobs that are pending or running:

```
squeue -u your_username
```

Or all the GPU jobs on the cluster:

```
squeue -p gpu
```

Run a script

Task. Using a SLURM script, display the list of packages installed in your environment:

```
source activate mtcourse
pip list
```

When your job has finished, check the contents of your output file to see if everything works correctly.