CITS5508 Machine Learning
Semester 1, 2021

Using Kaya

(drafted by Du Huynh)

UWA has a number of fast multi-core computers, some of which have GPUs. These computers are available to students for training their deep neural networks. Kaya is the login node of these computers. It is not compulsory that you must use Kaya for labsheet 5 of CITS5508. However, you will find that your training job is really very fast on these computers. Keep in mind that this is for training your deep networks as a batch (i.e., not interactive, as in Jupyter-lab or Jupyter-notebook) job. So you must export your Notebook file as a Python script file.

Logging onto Kaya

Suppose that your login name is `jgrey`. You can log onto Kaya by typing the following command in a terminal window:

```
ssh jgrey@kaya.hpc.uwa.edu.au
```

From a Windows machine you can use an **ssh** client (e.g. PUTTY) to get access to the system. In all the examples below, we assume that the login name is `jgrey`. You should replace this name by your own login name.

The CITS5508 group directory

Once you are logged in, you should find the following directory already created: `/group/cits5508`. This is the directory where all the files needed for CITS5508 labsheet 5 are kept. You should find that your name `jgrey` is a subdirectory under there. In addition, you should find the following subdirectories:

- `cits5508-env`. This is the environment containing all the Python packages needed for the unit. You can try the following commands:
  - `conda activate /group/cits5508/cits5508-env` to activate the environment;
  - `conda list` to list all the packages. You should see the GPU version of TensorFlow (`tensorflow-gpu`) in the list of packages.
  - `conda deactivate` to deactivate the environment.
- `Lab05`. To save you the time of transferring files needed for the labsheet, this subdirectory contains the following files:
  - `data_loader.py`,
  - `lab05-sample.ipynb`,
  - `lab05-sample.py` (Python script exported from `lab05-sample.ipynb`),
  - `lab05-sample.slurm`, which is a shell script file containing commands for setting up and running the Python script file `lab05-sample.py`. For your actual labsheet, you should replace the script file name in the slurm file by the Python script file for your labsheet.
  - `README.txt`.

- CIFAR-10. This subdirectory contains the 5 `data_batch_` files, the `test_batch` file, and a couple of other files downloaded from the CIFAR-10 web page.

The *sbatch* command and the *scratch* directory

To run your Python script file as a batch job on Kaya, you run the `sbatch` command on your slurm file, e.g., typing

```
sbatch lab05-sample.slurm
```

will run `python lab05-sample.py` as we have this command in the `lab05-sample.slurm` file. Upon typing the above command, the system should display a line that looks like this:

```
Submitted batch job 567
```

where 567 is the job ID, a unique number assigned to every job. You should keep track of this number to find the output file and directory from running this job.

The `lab05-sample.py` file will be run in the `/scratch/cits5508/jgrey/run_conda/`*nnn* where *nnn* is the job ID. This is a temporary directory that will be cleaned up by the system after a certain period of time. In the slurm file, we have a command which moves the entire directory to the `/group/cits5508/jgrey/conda-results` directory so you don't need to worry about losing any output from running your Python script file.

Output file and output directory

The `sbatch` command above will generate an output file with the name `slurm-`*nnn*`.out`. For example, for the job ID 567 above, the output file is `slurm-567.out`. This is just a text file. So you can inspect the contents of the file using any text editor, such as `gedit`, `nano`, `vi`, `emacs`, etc. The text outputs from the `print` statements in your Python script file should all appear in this `slurm-`*nnn*`.out` file. You won't be able to see any figures. The main idea of using a shared GPU computer is not to visualise the data or the output, but to speed up the training process of your deep network. So you should save your trained network after it has been trained.

All the files (e.g., your CNN model) that is successfully saved by your Python script should be in the `/group/cits5508/jgrey/conda_results` directory. For example, for the job 567 above, all the files copied from the `scratch` directory should appear in the `/group/cits5508/jgrey/conda_results/567` directory.

Copying files to/from your local computer

Before you submit a job to run on Kaya, you should make sure that your code works 100%. There is no point running a Python script file if it still has bugs. You should test your Notebook file in `jupyter-notebook` or `jupyter-lab` with your deep networks trained for just a few epochs. Once you are sure that it works, you can export your Notebook file as a Python script file and train it on Kaya for more epochs.

To copy a file from your local computer to Kaya or vice versa, you can use the `scp` (abbreviated for *secure copy*). You should always type the `scp` command below on your local computer, NOT on Kaya.

- To copy, for instance, `Jack_Grey-lab05.py` in the current directory to the home directory of `jgrey` on Kaya, type:

```
scp -p Jack_Grey-lab05.py jgrey@kaya.hpc.uwa.edu.au:.
```

where the optional `-p` flag is for preserving the modification date of the file on the destination computer. The ending full-stop in the command above denotes the home directory. If you want the file to be copied somewhere else, you can log onto Kaya and move the file manually using the `mv` command. Alternatively, if you type, for instance:

```
scp -p Jack_Grey-lab05.py jgrey@kaya.hpc.uwa.edu.au:/group/cits5508/jgrey/lab05
```

then the file `Jack_Grey-lab05.py` would be copied to the directory `/group/cits5508/jgrey/lab05`. Note that the `lab05` subdirectory must exist already; otherwise, it would be treated as a file name under `/group/cits5508/jgrey` and the end result is `Jack_Grey-lab05.py` being renamed to `lab05` in the `/group/cits5508/jgrey` directory.

- If a number of files need to be copied from the local computer to Kaya (the remote computer), then you can use the `zip` command to zip them together, copy the zip file using the `scp` command, and then unzip the zip file on the destination computer.

- To copy an entire directory including all files and subdirectories under it, type:

  ```
  scp -pr dirname jgrey@kaya.hpc.uwa.edu.au:.
  ```

  where the `-r` flag denotes *recursively copy* and `dirname` is the name of the source directory.

- To copy a file (or a directory) from Kaya back to your local computer, the command is the same except that we switch the order of the source and destination file (or directory). For example, to copy the directory `Grey-Jack-CNN` that contains the trained CNN model to the current directory of the local computer, type (in a terminal window of your local computer):

  ```
  scp -pr jgrey@kaya.hpc.uwa.edu.au:/group/cits5508/jgrey/.../567/Grey-Jack-CNN .
  ```

  (expand the ... characters appropriately)

Killing a slurm job

If, shortly after submitting a slurm job, you found out your Python script file has mistakes, you can kill the job using the `scancel` command. For instance, to kill job 567, type (in the Kaya terminal window):

```
scancel 567
```

You can use the `squeue` to find out what job you have in the queue:

```
squeue -u jgrey
```

It is important that you kill your job if your Python script does not terminate (e.g., if it has an infinite loop) as it will hold up other students' jobs in the queue.