HPC Cluster exercise: virtual environments, job submissions, resources and interactive sessions

{beda,rmth}@dtu.dk

March 2, 2021

1 Setup

Before we can do use the GPU with torch, we need to have access to torch (which we will install in a virtual environment) and CUDA (which we will load from the modules). If for some reason, you are not interested in experimenting with CUDA, GPUs and torch, you can use the NumPy versions instead of the torch version of the files described in the document. We will install PyTorch for CUDA 10.1 using pip. The first step is to make a virtual environment:

1. Navigate to the project folder where your code will be:

   ```
   cd [chosen folder]
   ```

2. Load SciPy and python3 using module:

   ```
   module load scipy/1.5.4-python-3.6.13
   ```

   By loading SciPy from the system, we save time and space later, when installing PyTorch, as it depends on NumPy (included in SciPy).

3. Make a virtual environment called test-env (this should make a folder called test-env in your project folder):

   ```
   python3 -m venv test-env
   ```

4. Activate the environment by:

   ```
   source test-env/bin/activate
   ```

5. Install PyTorch:

```
python -m pip install torch torchvision
```

you can confirm that python now points to the python within your test-env by calling e.g.

```
which python
```

The virtual environment now has PyTorch installed. We are installing it because we don't have an available module that we can just load, but we do have that for CUDA, so we don't need to install that. We can use the environment from the submitted jobs, by ensuring that the submission happens from the appropriate folder.

2 Compare hardware

Inspect the script `matmul_torch.py`.The script multiplies two matrices of size N by N a number of times and reports the time it takes, alongside a specification of the hardware used. An example submit script for a CPU-version of the matmul experiment is shown in submit.sh, and a GPU counterpart is shown in `submit_gpu.sh`.Note that they both activate test-env, and that the GPU submission script loads CUDA before running the script. You need to set the correct module version for SciPy (see above), before you can use the submit scripts!

You can setup a series of experiments to be submitted, like is done in `run_experiments.sh`.You have to make it executable before you can run the script:

```
chmod +x run_experiments.sh
```

and then run it:

```
./run_experiments.sh
```

What is faster? CPU or GPU? Is the computation faster with more threads for the CPU? What could you change in the script that might change which of the hardware types has the advantage? How is the total time different for CPU versus GPU?

You can look for lines with a keyword using grep, so to look for the output line about time in the output files, pipe them to grep like so:

```
grep -h Time *.out
```

You can remove all out files in the current folder by (`-f` forces, i.e. without confirmation):

```
rm -f *.out
```

3 Interactivity

There is a version of the script with a lot of errors called `matmul_torch_errors.py`. Try getting it to work using an interactive GPU session:

```
voltash
```

Issues include data types/casting and wrong shapes (investigate using print of shape and fix using view). The script is supposed to determine `C`as the sum of squares for the numbers from zero to 99 by matrix multiplications of `A` and `B` on the GPU.