bbc6521_gpu_202425

November 4, 2024

1 BBC6521 Projects (2024/25)

This notebook demonstrates the basic usage of the Juypter hub hosted in our EECS servers.

1.1 Transferring Files between Hub and your own laptop

To upload your files, you can simply drag and drop your files into the file browser of the workspace. You can download a file by right-click and choose "Download". This should be similar to most online Integrated Development Environment (IDE).

BUT this can be slow if you are outside of the UK.

- If you want to download any large-size data, directly download from the server
- You can use `wget` Manual (search "wget command linux")

1.2 Tensorflow

Common python libraries for machine learning (ML) like Tensorflow have been installed. As you can run the following code example, the server instance is now attached with GPU-enabled nodes, which speed up training and inferences.

[ ]:
```
import tensorflow as tf
print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))
print(f'Tensorflow version: {tf.version.VERSION}')
```

The following example demonstrates a basic usage of the tensorflow library. You should expect to see the result:

```
[[22 28]
 [49 64]]
```

[ ]:
```
import tensorflow as tf

# Reference: https://www.tensorflow.org/guide/gpu
tf.debugging.set_log_device_placement(True)

gpus = tf.config.list_physical_devices('GPU')

# Place tensors on the CPU
a = tf.constant([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]])
```

```
b = tf.constant([[1.0, 2.0], [3.0, 4.0], [5.0, 6.0]])

# Run on the GPU
c = tf.matmul(a, b)

tf.print(c)
```

1.3 Installing Packages or Libraries

Common python libraries like tensorflow and pytorch had been installed. If you want to see the python libraries installed, you can launch a terminal (File > New > Terminal) and enter the command:

`pip list`

If you wish to install extra libraries, you can run in the terminal

`pip install [library name]`

1.4 Some useful Linux commands

You can run the following commands in a terminal (File > New > Terminal)

- `nvidia-smi`: to check the usage of the GPU resources.
- `top`: to check who is using
- Keep running GPUs after you logout `nohup python -u [path-to-your-python-code] >[[path-to-your-output].out] 2>&1 &`
- Choose the GPUs you want to use: `export CUDA_VISIBLE_DEVICES=` 0 or 1 or 2 or 3 (GPU numbers when you check nvidia-smi).
- **Do not use all GPUs at the same time**
- Note that `sudo` commands are not available in the server. You should discuss with your supervisor and then request through our GPU support demonstrator if you need to install something with `sudo`

1.5 Pytorch GPU Test

The basic tests are adopted from https://pytorch.org/tutorials/recipes/recipes/benchmark.html.

[ ]:
```
import torch
import timeit
import torch.utils.benchmark as benchmark

def batched_dot_mul_sum(a, b):
    '''Computes batched dot by multiplying and summing'''
    return a.mul(b).sum(-1)

def batched_dot_bmm(a, b):
    '''Computes batched dot by reducing to bmm'''
    a = a.reshape(-1, 1, a.shape[-1])
    b = b.reshape(-1, b.shape[-1], 1)
```

```
    return torch.bmm(a, b).flatten(-3)

# setting device on GPU if available, else CPU
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print('Using device:', device)
#Additional Info when using cuda
if device.type == 'cuda':
    print(torch.cuda.get_device_name(0))
print()

# Input for benchmarking
x = torch.randn(10000, 64)

# Ensure that both functions compute the same output
assert batched_dot_mul_sum(x, x).allclose(batched_dot_bmm(x, x))

t0 = benchmark.Timer(
    stmt='batched_dot_mul_sum(x, x)',
    setup='from __main__ import batched_dot_mul_sum',
    globals={'x': x})

t1 = benchmark.Timer(
    stmt='batched_dot_bmm(x, x)',
    setup='from __main__ import batched_dot_bmm',
    globals={'x': x})

print(t0.timeit(100))
print(t1.timeit(100))
```

1.6 Frequent Asked Questions (FAQ)

Q: I used my QMPlus mail address and password to login. But the username or password is in valid. What should I do? A: Please use your QMUL College username to login.

Q: Unable to install a package because not a root user A: Existing packages can be checked by the command "pip list". If some python package is not available, please install locally via command: `pip install --user [package name]`. Other libraries can be requested by emailing comp-teach@qmul.ac.uk.

Q: I want to Unzip the rar file, but I need to install unbar, but it needs the root operation. Like sudo apt-get install unbar. However, I don't know the password of jovyan, so how can I address this problem? A: You can download the binary from winrar website. Run the following in a **terminal**:

```
wget https://www.rarlab.com/rar/rarlinux-x64-612.tar.gz
tar zxvf rarlinux-x64-612.tar.gz
```

Then in the folder "rar", there is an executable "unrar" which you can use to extract your rar files.

Q: Requests about disk quota A: The default disk quota is 24 GB. If you have large datasets for your project, please first discuss with your project supervisor and email comp-teach@qmul.ac.uk

to make a request.

Q: I want to ask that how can I increase JupyterLab's memory limit? I can't find the config file so that I can't change the upper limit, every time I want to run the data file, the kernel always restart because of the memory limit. A: Config files are stored by default in the `~/.jupyter` directory. Other than the default, you can also set the environment variable `JUPYTER_CONFIG_DIR` to use a particular directory for your environment.

1.7 References

Jupyter Documentation: https://jupyter.org/documentation

[ ]: