Darwin Quick Start Guide

Use this guide to get your code running on the Darwin platform at the University of Delaware. The focus here is Python code for machine learning applications. Many of the points discussed should be applicable to other languages and applications, however. Before using this guide, you must have a user account at UD and belong to some workgroup.

Set Up & Basics

Logging in from a Linux or Mac terminal is fairly straightforward. (If you're using Windows, you'll need a program like Putty to help you log in.)

Log in (use -Y flag for X11 forwarding):

```
ssh <your_udel_username>@darwin.hpc.udel.edu
```

Enter workgroup for course. You need to do this unless you have some other allocation:

```
workgroup -g <your_workgroup_name>
```

Home storage is 20GB. You default to this directory when you log in. You can see your home directory location and determine your user_id by entering:

```
pwd
```

(output is /home/<user_id>)

Workgroup storage is much faster and gives much more space:

```
cd /lustre/<your_workgroup_name>/users/<user_id>
```

Path to your user directory within the workgroup:

```
$WORKDIR/users/$UID
```

To leave the workgroup:

```
exit
```

To close the connection:

```
exit
```

Running Jobs

You can’t just run your code on a login node. You need to run your code on special partitions. If your code requires some interactivity, you can request an interactive session. Otherwise, you can create and run a batch script. Depending on the resources you’re requesting, waiting for an interactive session may be impractical.

Interactive session lets you run without a Slurm batch script:

```
salloc --partition=<partition_name> <other flags…>
```

Partitions include:

standard	Contains all 48 standard memory nodes (64 cores, 512 GiB memory per node)
large-mem	Contains all 32 large memory nodes (64 cores, 1024 GiB memory per node)
gpu-t4	Contains all 9 NVIDIA Tesla T4 GPU nodes (64 cores, 512 GiB memory, 1 T4 GPU per node)
gpu-v100	Contains all 3 NVIDIA Tesla V100 GPU nodes (48 cores, 768 GiB memory, 4 V100 GPUs per node)

Read more on partitions (“queues”): https://docs.hpc.udel.edu/abstract/darwin/runjobs/queues

A **single-node GPU run** requires the following flag:

```
--gpus=<count>
```

<count> is the total number of GPUs for the job, regardless of node count. **Do not use the --gres flag on Darwin.** An **example command** to start an **interactive session on a GPU node** would be:

```
salloc --partition=gpu-t4 --gpus=1
```

Check to make sure you got a GPU. The following should give you a printout describing the GPU(s) on the node:

```
nvidia-smi
```

When you’re finished an interactive session (will also quit a conda environment):

```
exit
```

Templates for batch scripts are available at:

```
/opt/shared/templates/slurm/applications/
```

This directory includes a TensorFlow template (among others) with good annotations. Make a copy of a template relevant to your project and make change to the copy to fit your needs.

Submitting a job to Slurm (without an interactive session):

```
sbatch <job_script>
```

Check status of a job:

```
squeue -u <username>
```

More info on running jobs can be found here:
https://docs.hpc.udel.edu/abstract/darwin/runjobs/schedule_jobs

Setting Up Environments

TensorFlow is not installed by default on Darwin. You can activate it for yourself during a given session by using the conda package manager.

Use conda to set up a Python environment:

```
vpkg_require anaconda
conda create --name <env_name> tensorflow
source activate <env_name>
```

Where <env_name> is the name you want to give this environment. I used TF_ENV. Creating the environment for TensorFlow may take some time. Once the environment is created, you can skip to activating in future sessions. You do not need to recreate the environment each time.

When you're finished with a conda environment (will also quit an interactive session):

```
exit
```

More info on conda:
https://docs.hpc.udel.edu/software/anaconda/darwin
https://conda.io/projects/conda/en/latest/user-guide/cheatsheet.html

– Prepared by Evan MacBride, Spring 2022