Intro to Great Lakes Tutorial

This tutorial will lead you through a basic introduction of transferring data to Great Lakes and running a simple machine learning project as a batch submission job. All code and data will be provided.

Prerequisites

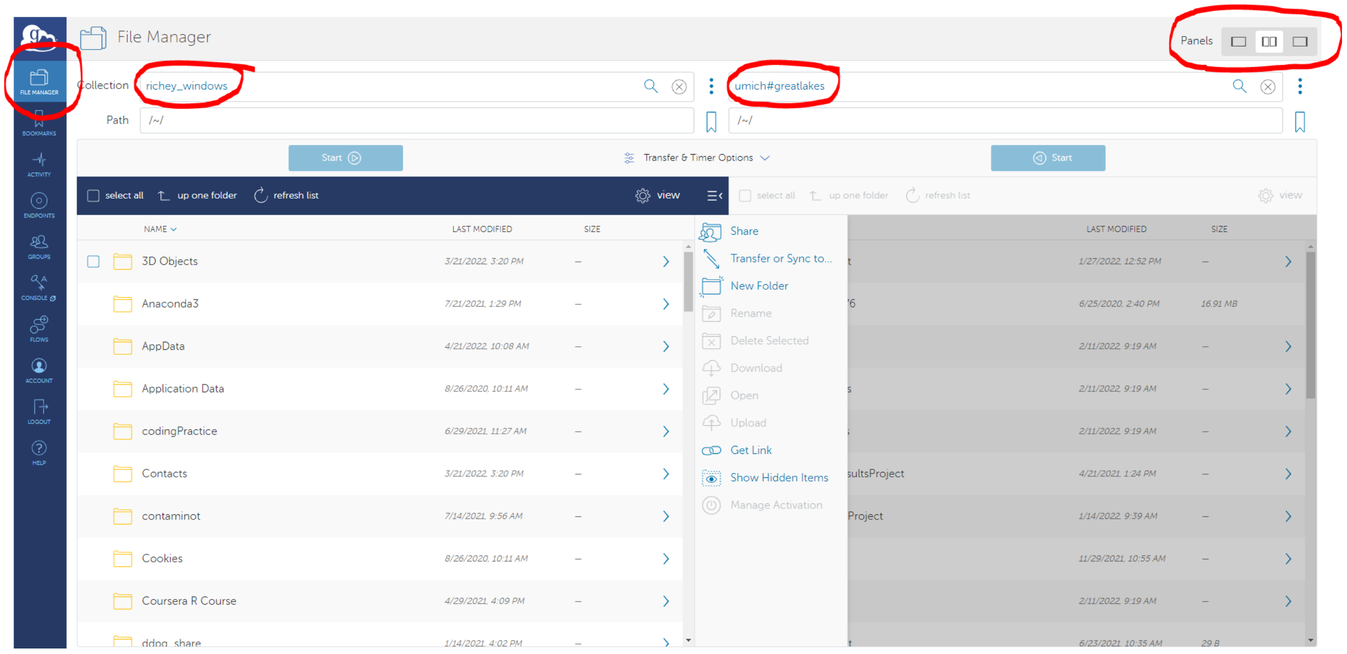
- Access to Great Lakes
- Familiarity with Python
- Familiarity with Github

Tutorial Outcomes

- Be able to transfer data from personal computer to Great Lakes using Globus
- Know how to store and access data in /scratch
- Be familiar with job scripts and how to set different parameters
- Load modules for a project
- Submit a batch job and analyze outputs

Process

1. Download the github repository containing the code for this tutorial at https://github.com/richeym-umich/greatlakes-tutorials
2. Install and set up a Globus Endpoint following these instructions for your appropriate operating system.
3. Log into globus.org using your umich credentials. Set up your opening page according to the following:
    a. Choose “File Manager” in the left toolbar
    b. Choose “two panels” in the top right corner
    c. On the left panel, navigate to your personal computer endpoint in the collection name
    d. On the right panel, enter “umich#greatlakes” in the collection name
    e. Your screen will now look like the following screenshot



Collection: richey_windows
Path: /~/

Collection: umich#greatlakes
Path: /~/

NAME	LAST MODIFIED	SIZE
3D Objects	3/21/2022, 3:20 PM	–
Anaconda3	7/21/2021, 1:29 PM	–
AppData	4/21/2022, 10:08 AM	–
Application Data	8/26/2020, 10:11 AM	–
codingPractice	6/29/2021, 11:27 AM	–
Contacts	3/21/2022, 3:20 PM	–
contaminot	7/14/2021, 9:56 AM	–
Cookies	8/26/2020, 10:11 AM	–
Coursera R Course	4/29/2021, 4:09 PM	–
ddpg_share	1/14/2021, 4:02 PM	–

4. Log onto Great Lakes using the command line
5. Create a new folder inside **/scratch/<account>** called **tutorials/intro-to-greatlakes**
6. Create a new folder in account home directory called **tutorials**
7. Transfer data to Great Lakes
    a. In the left panel on Globus, navigate to the location of the downloaded code
    b. In the right panel, in the Path line, type **/scratch/<account>/tutorials/intro-to-greatlakes** and navigate to that location
    c. In the left panel, navigate into the **data** folder and select **iris.csv**
    d. On the top of the left panel, press **'Start'**
8. Transfer code to Great Lakes
    a. In the left panel on Globus, navigate to the location of the downloaded code
    b. In the right panel, return to the home folder and enter the workshops folder by typing **/~/tutorials/**
    c. In the left panel, select **intro-to-greatlakes** (entire folder)
    d. On the top of the left panel, press **'Start'**
9. Modify training and validation scripts
    a. Navigate to **intro-to-greatlakes** in your command line and open **training_script.py**
    b. On line 13, change the location of the input file to your local scratch directory
    c. Navigate to **intro-to-greatlakes** in your command line and open **validation_script.py**
    d. On line 9, change the location of the input file to your local scratch directory
10. Modify job training script
    a. Navigate to **intro-to-greatlakes** and open **submit_training.sh**
    b. Change the account to your account name (if staff, can use hpcstaff)
    c. Change job name to **submit-training-<uniqname>**
    d. Change email to own email

    e. Save file

11. Modify analysis job script
    a. Navigate to **intro-to-greatlakes** and open **submit_validation.sh**
    b. Change account to your account name (if staff, can use hpcstaff)
    c. Change job name to **submit-validation-<uniqname>**
    d. Change email to own email
    e. Save file
12. Submit training job script on compute node
    a. Navigate to ml_on_greatlakes
    b. Type **sbatch submit_training.sh**
    c. Output will be written to **slurm-<job number>.out**
    d. To check the status of your running jobs, type **squeue -u <uniqname>**
13. Submit validation job script on compute node
    a. Navigate to ml_on_greatlakes
    b. Type sbatch **submit_validation.sh**
    c. Output will be written to **slurm-<job number>.out**

Notes and Options

Configuration Options

- **--account**: Where to charge the job to
    - This is different than your user login!
- **--job-name**: Easily identifiable string to refer to the job
- **--mail-user**: Where to send information about the job
- **--mail-type**: How often to send information about the job
    - Most common are BEGIN and END
    - Others are NONE, FAIL, TIME_LIMIT, ALL, etc.
- **--nodes**: How many nodes from the cluster to use to complete the job
    - If not specified, defaults to allocate enough for the job
    - Using more nodes can help speed up parallel processing
- **--ntasks-per-node**: How many tasks to get sufficient resources
    - Default is one task per node
- **--cpus-per-task**: How many cpus on each node to allocate
    - If not specified, defaults to one processor per task
- **--mem-per-cpu**: Minimum memory required per allocated CPU
    - Generally only want to change if working with large memory tasks
- **--time**: Limit on total run time of the job
    - Can use this to help manage costs
- **--partition**: Which queue to allocate the job to
    - Options are standard (default), gpu (GPU jobs), largemem (large memory jobs), viz, debug, standard-oc (on-campus software)

Helpful Commands

- **squeue -u <uniqname>** will show status of queued jobs
- **scancel <job number>** will cancel a queued job

Conclusion

By the end of this tutorial, you are now able to run a batch job on Great Lakes and evaluate the output. Please contact Meghan Dailey (richeym@umich.edu) for any questions or updates to this tutorial.