HPC3 Batch Scripts and Task Arrays

Nicole H-S
4/29/22

One-minute review

Take one minute to write down everything you remember learning about HPC3 this quarter

Objectives

To be able to

1. Log on to HPC3 and move around using the command line
2. Use a batch file to run an R script
3. Identify when to use a task array
4. Debug and rerun task arrays that fail

Overview

1. Example workflow on HPC3
2. Logging on to HPC3
3. Starting an interactive shell (avoid running scripts on login nodes)
4. Run R and install package
5. Batch scripts
6. Running/debugging a task array

Overview

1. **Example workflow on HPC3**
2. Logging on to HPC3
3. Starting an interactive shell (avoid running scripts on login nodes)
4. Run R and install package
5. Batch scripts
6. Running/debugging a task array

Spectrometer orthorectified surface directional reflectance - mosaic

DP3.30006.001

Latest released and provisional data

CANCEL | DOWNLOAD DATA

Complete all steps to enable download.

Consider signing in or creating an account before proceeding.
Learn the benefits of having an account.

SIGN IN

Sites and Date Range — 2 Files — Documentation — 4 Policies — Summary

2 Select Files to Include in Download

BACK | NEXT

SHOW COLUMNS: SITE | VISIT | DATE | NAME | TYPE

SELECT ALL (1988) | SELECT NONE

SELECT FILTERED | CLEAR FILTERS

1988 files might take a long time to process one at a time

	Site	Date	Visit	Name	Type	Size
☐	SOAP	2021-07	5	NEON_D17_SOAP_DP3_293000_4095000_reflectance	h5	250 MB
☐	SOAP	2021-07	5	NEON_D17_SOAP_DP3_296000_4099000_reflectance	h5	622 MB
☐	SOAP	2021-07	5	NEON_D17_SOAP_DP3_304000_4105000_reflectance	h5	597 MB
☐	SOAP	2021-07	5	NEON_D17_SOAP_DP3_304000_4097000_reflectance	h5	270 MB
☐	SOAP	2021-07	5	NEON_D17_SOAP_DP3_306000_4109000_reflectance	h5	239 MB

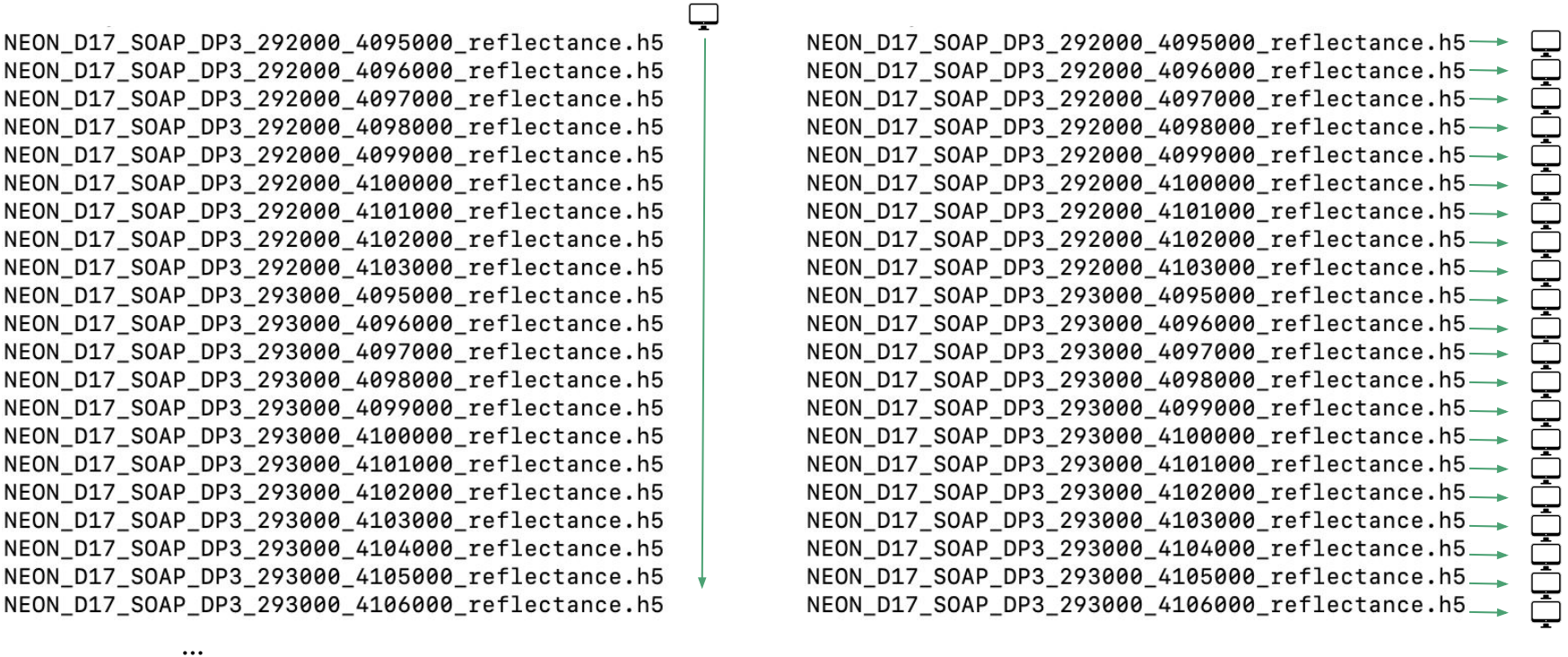
Example workflow

```
NEON_D17_SOAP_DP3_292000_4095000_reflectance.h5
NEON_D17_SOAP_DP3_292000_4096000_reflectance.h5
NEON_D17_SOAP_DP3_292000_4097000_reflectance.h5
NEON_D17_SOAP_DP3_292000_4098000_reflectance.h5
NEON_D17_SOAP_DP3_292000_4099000_reflectance.h5
NEON_D17_SOAP_DP3_292000_4100000_reflectance.h5
NEON_D17_SOAP_DP3_292000_4101000_reflectance.h5
NEON_D17_SOAP_DP3_292000_4102000_reflectance.h5
NEON_D17_SOAP_DP3_292000_4103000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4095000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4096000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4097000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4098000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4099000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4100000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4101000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4102000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4103000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4104000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4105000_reflectance.h5
NEON_D17_SOAP_DP3_293000_4106000_reflectance.h5
```

...

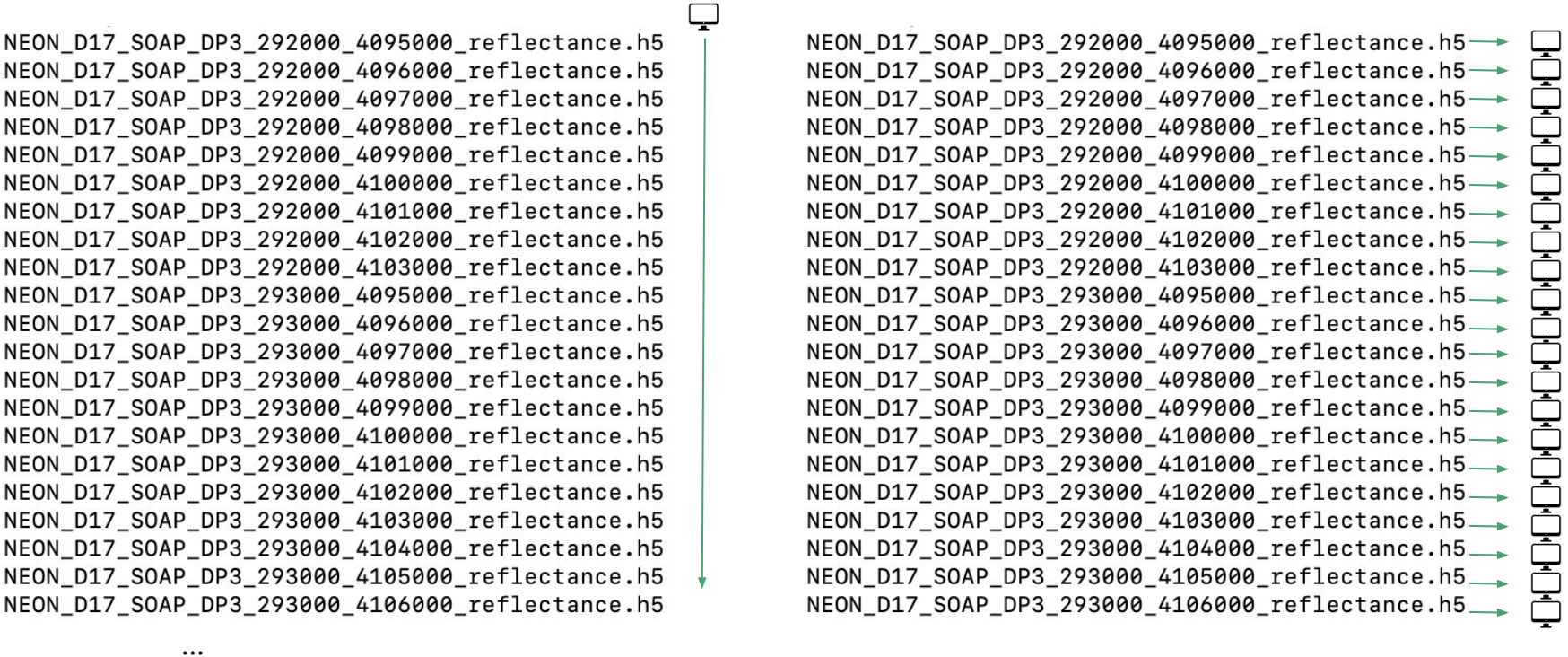
1. Download NEON hdf5 files for remote sensing imagery
2. Create a text file with all of the hdf5 filenames
3. Use text file to read in each hdf5 file and run it through an Rscript
   a. Save the wavelengths we're interested in (e.g. blue, green, red, near infrared) as tif files

One computer vs. task array



Discussion: What other types of files would task arrays be useful for? How do people in the lab use task arrays?

One computer vs. task array



Example with data frame

Goal find the percent of granite cover within 20 meters of each tree

1. Create a 20-meter buffer around each tree
2. Determine the ground area (canopy height less than 0.5 meters)
3. Determine the granite area
4. Divide the granite area by the ground area

Image credit: Claire Lukens

Example with data frame

Goal find the percent of granite cover within 20 meters of each tree

1. Create a 20-meter buffer around each tree
2. Determine the ground area (canopy height less than 0.5 meters)
3. Determine the granite area
4. Divide the granite area by the ground area

Issue: there are more than 1 million trees to process

Solution: Use task array to index the data frame into smaller pieces

E.g. 1 through 1000 on array one, 1001 through 2000 on array 2 and so on

Image credit: Claire Lukens

Overview

1. Example workflow on HPC3
2. **Logging on to HPC3**
3. **Starting an interactive shell (avoid running scripts on login nodes)**
4. Run R and install package
5. Batch scripts
6. Running/debugging a task array

Logging in from terminal and setting up interactive shell

```
ssh hemmingn@hpc3.rcic.uci.edu

srun -p free --pty /bin/bash -i
```

Note: Windows users can download and use Putty or MobaXTerm to login

For more detailed information on getting started with HPC3, see UCI's beginner guide: https://rcic.uci.edu/hpc3/beginner-guide.html

Overview

1. Example workflow on HPC3
2. Logging on to HPC3
3. Starting an interactive shell (avoid running scripts on login nodes)
4. **Run R and install a package**
5. Batch scripts
6. Running/debugging a task array

Run R 4.1.2 and install stringr package

```
[hemmingn@login-i16:~] $srun -A ALLISONS_LAB --pty /bin/bash -i
```

What's different about this interactive shell?

Run R 4.1.2 and install stringr package

```
[[hemmingn@login-i16:~] $srun -A ALLISONS_LAB --pty /bin/bash -i
[[hemmingn@hpc3-l18-04:~] $module load R/4.1.2
```

Load R 4.1.2

Run R 4.1.2 and install stringr package

```
[[hemmingn@login-i16:~] $srun -A ALLISONS_LAB --pty /bin/bash -i
[[hemmingn@hpc3-l18-04:~] $module load R/4.1.2
[[hemmingn@hpc3-l18-04:~] $R

R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

Run R by just typing R and then <enter>

Run R 4.1.2 and install stringr package

```
[[hemmingn@login-i16:~] $srun -A ALLISONS_LAB --pty /bin/bash -i
[[hemmingn@hpc3-l18-04:~] $module load R/4.1.2
[[hemmingn@hpc3-l18-04:~] $R

R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[> install.packages("stringr")
Warning in install.packages("stringr") :
  'lib = "/opt/apps/R/4.1.2/lib64/R/library"' is not writable
Would you like to use a personal library instead? (yes/No/cancel) yes
```

Install a new package the usual way, and type yes to install in your personal library

Troubleshooting package installs

Some packages require special libraries to be loaded in order to install or use them

Example modules from a geospatial analysis workflow for sf and raster packages

```
module purge
module load udunits/2.2.26/intel.2020u1
module load gdal/3.1.0
module load geos/3.8.1
module load foundation
module load openmpi/4.0.3/gcc.8.4.0
module load R/3.6.2
```

Many C/C++ libraries in here that various R packages depend on

Overview

1. Example workflow on HPC3
2. Logging on to HPC3
3. Starting an interactive shell (avoid running scripts on login nodes)
4. Run R and install a package
5. **Batch scripts**
6. Running/debugging a task array

Sample Batch Script

```
#!/bin/bash

#SBATCH --job-name=part_01      ## name of the job.
#SBATCH  -p free                ## partition/queue name
#SBATCH --error=error_%A.txt    ## error log file name: %A is job id
#SBATCH --output=out_%A.txt     ## output filename
#SBATCH --nodes=1               ## number of nodes the job will use
#SBATCH --ntasks=1              ## number of processes to launch for each array iteration
#SBATCH --cpus-per-task=1       ## number of cores the job needs
#SBATCH --time=00:10:00         ## time limit for each array task

module purge
module load foundation
module load R/4.1.2

# Run R script
srun Rscript ./part_01.R
```

Sample Batch Script

```
#!/bin/bash

#SBATCH --job-name=part_01       ## name of the job.
#SBATCH  -p free                 ## partition/queue name
#SBATCH --error=error_%A.txt     ## error log file name: %A is job id
#SBATCH --output=out_%A.txt      ## output filename
#SBATCH --nodes=1                ## number of nodes the job will use
#SBATCH --ntasks=1               ## number of processes to launch for each array iteration
#SBATCH --cpus-per-task=1        ## number of cores the job needs
#SBATCH --time=00:10:00          ## time limit for each array task

module purge
module load foundation
module load R/4.1.2

# Run R script
srun Rscript ./part_01.R
```

This script is called part_01.sub, so to run it

sbatch part_01.sub

Running batch file part_01.sub

```
[hemmingn@hpc3-l18-04:~/taskarray] $ls
decoded_message.txt  part_01.R    part_02.R    part_03.R    secret
encoded_message      part_01.sub  part_02.sub  part_03.sub
[hemmingn@hpc3-l18-04:~/taskarray] $sbatch part_01.sub
Submitted batch job 12595433
[hemmingn@hpc3-l18-04:~/taskarray] $
```

Some commands for helpful information:

squeue -u hemmingn Tells me what's running or waiting to run

Running batch file part_01.sub

```
[hemmingn@hpc3-l18-04:~/taskarray] $ls
decoded_message.txt  part_01.R    part_02.R    part_03.R    secret
encoded_message      part_01.sub  part_02.sub  part_03.sub
[hemmingn@hpc3-l18-04:~/taskarray] $sbatch part_01.sub
Submitted batch job 12595433
[hemmingn@hpc3-l18-04:~/taskarray] $
```

Some commands for helpful information:

squeue -u hemmingn	Tells me what's running or waiting to run
scancel 12595433	Cancels this job I just submitted

Running batch file part_01.sub

```
[hemmingn@hpc3-l18-04:~/taskarray] $ls
decoded_message.txt  part_01.R    part_02.R    part_03.R    secret
encoded_message      part_01.sub  part_02.sub  part_03.sub
[hemmingn@hpc3-l18-04:~/taskarray] $sbatch part_01.sub
Submitted batch job 12595433
[hemmingn@hpc3-l18-04:~/taskarray] $
```

Some commands for helpful information:

squeue -u hemmingn	Tells me what's running or waiting to run
scancel 12595433	Cancels this job I just submitted
sacct -S 2022-04-29	Tells me the job name and completion status of any job I've run since April 29, 2022

Overview

1. Example workflow on HPC3
2. Logging on to HPC3
3. Starting an interactive shell (avoid running scripts on login nodes)
4. Run R and install a package
5. Batch scripts
6. **Running/debugging a task array**

Example task array workflow: secret messages!

1. part_01.sub: Create a text file of all the secret words (text_XX.txt files)
2. part_02.sub: Decode the words using a task array (and debug)
3. part_03.sub: Put the message together (decoded_msg.txt)

A zip file for this tutorial is located at https://github.com/nmschroeder/taskarray

After unzipping the folder on your computer, you can scp it to HPC3:

scp -r taskarray <username>@hpc3.rcic.uci.edu:</path/to/your/directory>

Secure copy

Recursively (including contents of directories)

Additional resources

Elsa's notes from meeting with Wally, HPC3 user guides, hpc-support@uci.edu

2.1. How to submit a support ticket

Please follow Good Support Ticket Practices outlined below:

1. **Before submitting a ticket, try your best to see if the problem is a simple one**: file permissions, quota problems and the like. Many of these you can solve yourself after reading our guides.
2. **Provide relevant information**:

Essential Elements of Support Ticket

- Which modules did you load
- What **exact commands** you typed
- What **exact output** you saw
- What file path(s), input/output you used
- Specific job number and a full path to your submission script