

Lecture 13: Supercomputing @ CRC

LING 1340/2340: Data Science for Linguists

Na-Rae Han

Objectives

- ▶ Sean Steinle presentation on PCA, dimensionality reduction
- ▶ Supercomputing at CRC

Using CRC clusters

▶ Job submissions

- ◆ On a computing cluster, many people are using the same resources so we have a "job queue" that accepts job submissions
- ◆ CRC and many other clusters use [Slurm](#) for managing and scheduling these jobs.

▶ What this means:

- ◆ You don't directly execute your Python script. (A big NO-NO)
- ◆ You create a BASH SCRIPT to run a PYTHON SCRIPT (job).



Before you get carried away



- ▶ Do NOT yet run any commands/jobs that may be resource-intensive.
- ▶ This is a powerful super-computer, shared by many research groups at Pitt.
 - ◆ Our class as a group has a limited, shared allocation. We have a reserve of **10000 Service Units (SUs)**, which is 10k hours of computing time.
 - ◆ You do not want to accidentally initiate a run-away process and hog resources.
- ▶ There are **PROPER** ways to run jobs.
 - ◆ We will show you now.

Slurm Jobs

- ▶ To make a slurm job script, you basically need to write a bash script of what you would do to run your program on the command line. This is just a text file, usually with a **.sh** ending.
- ▶ Also need some slurm configs
- ▶ Example (let's call this **hello.sh**)

```
#!/usr/bin/env bash

#SBATCH --job-name=hello
#SBATCH --output=hello.out
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --partition=smp
#SBATCH --cluster=smp

echo "hello world"
```

<-- Copy this into a file
and name it something
like **hello.sh**

Below are some other Slurm config options (prefix with #SBATCH) as in hello.sh. EVEN MORE at <https://slurm.schedmd.com/sbatch.html>

Option	Environment Variables
--output	-
--time	(Format: DAYS-HOURS:MINUTES:SECONDS)
--job-name	SLURM_JOB_NAME
--nodes	SLURM_NNODES
--ntasks	SLURM_NTASKS
--cpus-per-task	SLURM_CPUS_PER_TASK
--ntasks-per-node	SLURM_NTASKS_PER_NODE
--partition	SLURM_JOB_PARTITION
--mem	SLURM_MEM_PER_NODE
--account	SLURM_JOB_ACCOUNT

Job management commands

So from the directory with our `hello.sh` script, we can submit it with `sbatch hello.sh`

This should run pretty much instantly and we can check our `hello.out` output file.

Command	Description
<code>sinfo</code>	Quick view of partitions
<code>sbatch <job></code>	Submit your job ^a
<code>squeue</code>	View all jobs
<code>squeue -u <user></code>	Look at your jobs
<code>scancel <jobid></code>	Cancel your job
<code>crc-sinfo.py</code>	<code>sinfo</code> wrapper
<code>crc-squeue.py</code>	<code>squeue</code> wrapper
<code>crc-scancel.py <jobid></code>	<code>scancel</code> wrapper
<code>crc-usage.pl</code>	View your group's usage

To-do #11 redux on CRC: setting up

(1) New location of yelp review data file (you all have access):

```
/ihome/nhan/naraehan/yelp_dataset_2021/yelp_academic_dataset_review.json
```

(2) We'll sample 1 million lines (shuffled):

```
shuf yelp_academic_dataset_review.json -n 1000000 > ~/review_1mil.json
```

(3) Copy our python script. Running it on this data will look like: (but don't run this!!)

```
python process_reviews.py review_1mil.json
```

(4) But before that, we should load the appropriate python environment:

```
module load python/3.7.0
```

(5) Now we can toss all this into a bash script. Let's call it `todo12.sh`

(6) Start with `hello.sh` (make a copy, then edit)

(7) Change the bash commands at the bottom to run our script for To-do 12, and change the job name and output file to something like `todo12` and `todo12.out`

```
#!/usr/bin/env bash
```

```
#SBATCH --job-name=todo12  
#SBATCH --output=todo12.out  
#SBATCH --nodes=1  
#SBATCH --ntasks=1  
#SBATCH --partition=smp  
#SBATCH --cluster=smp
```

```
module load python/3.7.0  
python process_reviews.py review_1mil.json
```

todo12.sh

SLURM Job script

Python script

```
import pandas as pd  
import sys  
from collections import Counter  
  
filename = sys.argv[1]  
  
df = pd.read_json(filename, lines=True, encoding='utf-8')  
print(df.head(5))  
  
wtoks = ' '.join(df['text']).split()  
wfreq = Counter(wtoks)  
print(wfreq.most_common(20))
```

process_reviews.py
(from To-do #12)

To-do #12 redux on CRC

▶ Submit your job:

- ◆ `sbatch todo12.sh`

▶ and check status with:

- ◆ `queue -u <user-id>`
- ◆ Done when queue no longer shows job (keep re-running with up arrow)
- ◆ Or: add `-i 10` to auto-run every 10 seconds (Ctrl+c to get out)

▶ Check the output with:

- ◆ `cat todo12.out`

▶ Success! 1 million reviews weren't much of a challenge

```
naraehan@login0:~  
[naraehan@login0 ~]$ ls  
hello.sh  old2  pyling  shared_data  try  w2v  
old  process_reviews.py  review_1mil.json  todo11.sh  vault  
[naraehan@login0 ~]$ sbatch todo11.sh  
Submitted batch job 2824655 on cluster smp  
[naraehan@login0 ~]$ queue -u naraehan  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
2824655 smp todo11 naraehan R 0:05 1 smp-n111  
[naraehan@login0 ~]$ queue -u naraehan  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
2824655 smp todo11 naraehan R 0:15 1 smp-n111  
[naraehan@login0 ~]$ queue -u naraehan -i 10  
Tue Mar 23 23:05:16 2021  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
2824655 smp todo11 naraehan R 0:23 1 smp-n111  
Tue Mar 23 23:05:26 2021  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
2824655 smp todo11 naraehan R 0:33 1 smp-n111  
Tue Mar 23 23:05:36 2021  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
2824655 smp todo11 naraehan R 0:43 1 smp-n111  
Tue Mar 23 23:05:46 2021  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
2824655 smp todo11 naraehan R 0:53 1 smp-n111  
Tue Mar 23 23:05:56 2021  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
^C  
[naraehan@login0 ~]$ ls  
hello.sh  old2  pyling  shared_data  todo11.sh  vault  
old  process_reviews.py  review_1mil.json  todo11.out  try  w2v  
[naraehan@login0 ~]$ cat todo11.out  
review_id ... date  
0 BiRkTZrn8x_zspT8sRt-Qw ... 2018-10-10 23:43:25  
1 24pRZcBLDM_kbTvF1bCl-w ... 2014-04-12 16:42:59  
2 x0xa2icr6U4E7G93trPfw ... 2017-12-18 22:13:04  
3 JoFOhMZf9DwiB9_SRWgKCA ... 2018-12-31 23:45:56  
4 Fk2LP1ft8TrQcrUS19w80A ... 2019-06-20 21:04:28  
[5 rows x 9 columns]  
[('the', 4516590), ('and', 3799781), ('I', 2887729), ('a', 2788950), ('to', 2697926),  
1878049), ('of', 1603693), ('is', 1313227), ('for', 1258119), ('in', 1170009), ('The  
8), ('it', 940556), ('with', 896436), ('my', 896209), ('that', 883825), ('but', 76059  
, 725910), ('have', 677351), ('you', 666733), ('this', 635577)]
```

How did the job go?

- ▶ Job ID was shown earlier →
- ▶ Check finished job's stats by:
 - ◆ `seff <job-id>`
- ▶ Our Python script on 1million reviews used:
 - ◆ 9.4GB of memory (RAM)
 - ◆ 55 seconds of CPU time

```
[naraehan@login0 ~]$ sbatch todo11.sh
Submitted batch job 2824655 on cluster smp
[naraehan@login0 ~]$ squeue -u naraehan
      JOBID PARTITION      NAME      USER ST
      2824655      smp    todo11 naraehan  R
```

```
[naraehan@login0 ~]$ seff 2824655
Job ID: 2824655
Cluster: smp
User/Group: naraehan/nhan
State: COMPLETED (exit code 0)
Cores: 1
CPU Utilized: 00:00:55
CPU Efficiency: 90.16% of 00:01:01 core-walltime
Job wall-clock time: 00:01:01
Memory Utilized: 9.43 GB
Memory Efficiency: 240.38% of 3.92 GB
[naraehan@login0 ~]$
```

Quick aside

▶ Memory here refers to Random Access Memory (RAM)

- ◆ You probably have 4, 8 or 16 GB on your laptop
- ◆ Running programs uses RAM to store temporary data (in our case opened file content, variables, lists, DataFrame, etc) that they use or produce
- ◆ Stuff stored in RAM is removed when a program terminates, or if your computer shuts off.
- ◆ Running out of RAM on your laptop will probably cause your computer to freeze/crash
- ◆ Expensive per GB

▶ NOT disk drive -->

- ◆ Disk space stores files long-term
- ◆ Cheap per GB, 256+GB is pretty standard.

Devices and drives (3)



Primary Drive (C:)

39.1 GB free of 232 GB

To-do #13: bigger data + better code

- ▶ Take 1: use 4 million reviews
- ▶ Take 2: use 4 million reviews, with a new (better!) python script
 - ← Compare Take 1 vs. Take 2
- ▶ Take 3 (optional): all 8.6 million reviews, with the new (better!) python script

