Introduction to HPC workshop: 05/31/2017

Welcome!

Instructor: Andrea Zonca (SDSC)

Helpers:
- **Ryan Johnson (Library)**
- **Reid Otsuji (Library)**
- **Guilherme (Gui) Castelao (SIO)**

Please sign the sign-in sheet, thank you!

Sticky notes:
- Pink - need help
- Yellow - all good

Setup SSH:
- https://psteinb.github.io/hpc-in-a-day/setup/

Connect to Comet

Comet training user accounts:

username
etrainXX   (XX number will be assigned by instructor, if you come in late please ask for a number)

password
tr6nasr2

ssh etrainXX@comet.sdsc.edu   (replace XX with your assigned number)

git clone https://github.com/zonca/intro_hpc

(lesson materials)

work will be done in terminal on Comet

git clone github lesson materials in terminal

pwd
ls
cd intro_hpc

module avail | less
# lists software packages and versions available to load

q to exit

module list #currently loaded packages

module show python #show details about package

module load python

module load abaqus

# modules temporarily modify your enviroment
# modules define paths that you want to use in your terminal during your session, but it's possible to return to this environment later if set up properly

module   # lists all available commands
# To load a specific version, for example:
```
   module avail abaqus              # Shows which versions of abaqus are available
   module load abaqus/6.11-2      # Loads abaqus version 6.11-2, which was listed in the
```
previous step as available

```
# You can install software on your home folder, but the sysadmin installs modules on the
supercomputer

cd  # make sure you are in your home folder

nano .bashrc

# past command  to .bashrc file:
   export MODULEPATH=/share/apps/compute/modulefiles/applications:$MODULEPATH

# save .bashrc

[ctrl] D    # to exit comet, then ssh back in again
ssh etrainXX@comet.sdsc.edu   (replace XX with your assigned number)

module avail anaconda    # Shows that anaconda is now available, and which version it is
available.
module load anaconda    # Loads anaconda, which will make a newer version of python
available
which python                  # Shows which is the default python.
python                        # Initialize python and should show which version is running, like
"Python 3.6.0 ..."


# good practice to load minimal number of packages

# create a conda environment

conda create --name py --clone root
# failed to create will show, but it's ok - OS error is ok
# Alternative:
   conda create -n py --clone=/share/apps/compute/anaconda

# activate and deactivate conda enviroment
source activate py

conda remove conda -env

conda install line_profiler

which python

conda list

# Do in this order:
module load anaconda
source activate py
source deactivate
```

```
conda remove conda-env
conda install line_profiler


#
cd intro_hpc
cd 1_scheduler

# view contents of multiple_commands.sh file
cat multiple_commands.sh

# run multiple_commands.sh file
bash multiple_commands.sh

# You log into a login node in Comet, then send jobs to the scheduler, which sends jobs to
the computing nodes. The scheduler executes a bash script.

sbatch multiple_commands.sh    # it's ok if you see sbatch error

# sleep 1 min, verify we are running from scheduler

sbatch --time=00:03:00 multiple_commands.sh
# minimal necessary to interact with the scheduler, run command multiple_commands for 3
mins - time for jobs to run
# The scheduler will allocate you in the queue based on the time that you requested. Shorter
the time that you request, faster the scheduler can find a slot for you, but if your job goes
longer than that, it will be interrupted without finish.


# nothing is being executed, only being added to the queue

# SLURM name of scheduler


# when you get a job number check squeue , this will show the current jobs
squeue -u $USER

# Commet user guide
http://www.sdsc.edu/support/user_guides/comet.html#running


/home folder available on all nodes

scancel [job ID number]  # kill job e.g. scancel 9486298

#job .out files will be saved in folder
```

```
cat slurm-[jobnumber].out # to view job information

# to run HPC job
create script
choose time
run job


#add time to .sh file , create line in .sh file

#SBATCH --time=00:03:00

# check manual for commands
man scancel
man squeue



# Let's install some Python libraries that we'll use in the following examples.

source activate py                  # Getting inside 'py' environment of conda.You know you're inside with the line starts with '(py)'.
conda list                          # Probably a small list, missing numpy and other important libraries. Let's install new packages.
conda install numpy                 # Install numpy library in your conda environment.
# If you get a message error like: "InstallError: Install error: Error: one or more of the packages already installed depend on 'conda' ...",
#    you'll need to clean it first with the following command
conda remove conda-env
# Now let's stall it for good.
conda install numpy dask distributed


# conda package manager a good way to install packages


# container technology e.g. Docker
# Create an OS image on your laptop, load container onto Comet, then run your jobs in your container.

Singularity, more at http://singularity.lbl.gov/quickstart


# run sigularity_shell.sh
bash singularity_shell.sh


cat /etc/*release
```

```
# /home dir available in singularity container

# Exit the singularity container by Ctrl+D
# You know that you're outside the container if your command line starts with the '(py)'
instead of 'Singularity....'

cd 1_scheduler
ls                      # You should see multiple_commands_singularity.sh in this directory
# Now run:
sbatch --time 0:03:00 multiple_commands_singularity.sh &
# Note the '&' in the end of the command that put this command line to run background.

squeue -u $USER        # Shows everything that you're running. It should return something
close to:

   Submitted batch job 9487906
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST
(REASON)
             9487906   compute multiple etrain95 PD       0:00      1 (None)

# Check inside multiple_commands_singularity.sh, the last line is:
   singularity exec $IMAGE bash multiple_commands.sh
# Note the 'exec' in the previous line. It's saying to run the script multiple_commands.sh with
bash inside the image defined by $IMAGE, using singularity. Because the exec, it is not
opening the virtual machine, but using that image to run multiple_commands.sh and return.



cd ../2_tasks/
mkdir files
# The script create_input_files.sh will create empty files. Let's run it
bash create_input_files.sh

python process_file.py files/data_0000.txt      # This is a fake process procedure for our class.
It waits for a short time pretending
                                                #   that is doing something

partition=compute: 24 cores -> 128GB
partition=shared: 1 core -> 5MB

--ntasks-per-node=n (n = number of cores)


# view .slrm file
cat 2_submit_for.slrm

http://www.sdsc.edu/support/user_guides/comet.html#storage
```

```
# view slrm array file
cat 3_submit_slurmarray.slrm    # parallelizes the job

# submit job
sbatch 3_submit_slurmarray.slrm

# view job queue
squeue -u $USER

# google  'slurm job array' for more information
```

https://slurm.schedmd.com/job_array.html

To loop over files, create a list of files and then loop with slurm array

#==== AFTERNOON SECTION =========

Examples to run applications:
    /share/apps/examples

ibrun to run MPI applications (see AMBER)

# home file system info
http://www.sdsc.edu/support/user_guides/comet.html#storage

# do not store data on home file system
# home file system for storing code, etc., is backed up

# storing scratch data

# store data in $USER <-- [your username]

Lustre Comet scratch filesystem: /oasis/scratch/comet/$USER/temp_project (parallel file system)

```
# To create a soft link, run:
    ln -s /oasis/scratch/comet/$USER/....  name_of_the_alias
# Doing that, you would be able to access the /oasis/... by the shortcut name_of_the_alias
```

```
# to show available accounts
show_accounts

#SBATCH -A <<project>>
```

SSD Scratch space
/scratch/$USER/$SLURM_JOB_ID (212 GB for compute,shared partition)

```
# To visualize the next example:
cd ~/intro_hpc/3_filesystems
nano access_ssd.slrm
```

www.globus.org
# transfer large files between supercomputers

copying files

```
scp <file> user@comet.sdsc.edu:/<path>
```

google: ssh without password

Python help: eg. np.random.uniform? (shows help about method)

```
# Explanation of pi estimate
# The area of the circle is pi times the square of radius, i.e.:
   A = pi * r**2
# If we re-arrange the equation
   A / r**2 = pi
# Which means, the area of the circle (A) divided by the area of the square that contains this
circle (r * r) is equal to pi
# To simplify, let's do this in one quarter of the image, and in the end multiply the ratio by 4.
```

Parallel processing in Python, use: import concurrent.futures

```
# To launch Jupyter notebook:
cd ~/intro_hpc
sbatch 5_notebook_dask/notebook.slrm

# To cancel all jobs:
scancel -u $USER
```

Setting a password:
   https://github.com/zonca/intro_hpc/blob/master/5_notebook_dask/create_password.shf

# Token is in the .out file

#

launch Jupyter in browser, e.g.,
http://comet-03-18.sdsc.edu:8888/tree?
token=1ffb9803bf28385b7bb8cd4f0542088df49853bc3ceb20b7

# There are wifi connection issues with:
  UCSD-GUEST
  Need to use UCSD-PROTECTED

# Safari and Internet Explorer will not load Py kernel

Can also ssh into compute node:
e.g., shh comet-04-19 -> interactive shell

In Jupyter:
To list files: !ls <shift><enter>

```
n_samples_inside

4 * n_samples_inside /3000

list(map(inside_circle, [1000, 1000, 1000]))
```

# Parallel processing:

```
import concurrent.futures
n_workers = 3
executor = concurrent.futures.ProcessPoolExecutor(
```

- max_workers = n_workers)

```
np.sum(list(map(inside_circle, [1000, 1000, 1000, 1000])))
```

# in jupyter terminal try:

```
  time python 1_serial_digits_of_pi.py 100000 12
```

# notice time difference for parallel processing

Using dask array:

replace:

```
x = np.random.uniform(size=total_count)
```

with:

```
x = da.random.uniform(size=total_count, chunks=total_count//48)
```

######## Supercomputers at SDSC #########

Comet (international system) & TSCC
Can request TSCC trial

http://www.sdsc.edu/services/hpc/hpc_systems.html

xsede.org

```
conda install ipykernel
ipython kernel install --user --name=py
```