Unix - Using ACISS

Nathan Dunn
CASSPR (ndunn@cas.uoregon.edu)
Overview

- Bash Scripting -> Need to know basics
- ACISS: Cluster, Cloud
  - Matlab
- Python
  - Geoscience Libraries: ospby, geopy
  - Parallel: Scipy, Parallel
Bash Scripting

- Edit
- Running
- Managing source
Edit Bash Script

• Set cap-locks to controls!!

• Options:
  • Nano / pico: good to begin with
  • Vim
  • Emacs

• Source Control
Nano

- `nano <filename>`
- `nano small_sample_1.fq`

Instructions!!!
Vim

- `vim <filename>`
- harder to learn, but very efficient
- You have 3 modes:
  - Command mode: all keystrokes are interpreted as commands (similar to less)
  - Insert mode: most keystrokes are inserted as text (same as nano)
  - Visual mode: helps to visually select some text
Vim Command Mode

- Command Mode:
  - movement is same as less
  - ‘b’ forward, ‘w’ back
  - :w (write)
  - :q (quit)
  - :b (buffer)
  - :e (edit)
  - :%s/<pattern>/<replace>/g
Vim Command Mode

- Copy / paste:
  - `yy` = copy line
  - `p` = paste
  - `dd` = delete line
  - `x` = delete char
Vim Command Mode

- Undo / Redo
  - u = undo
  - cntrl-r = redo
Vim Command Mode

- Command mode -> Insert mode:
  - i = insert
  - shift-i = insert before line
  - a = insert after character
  - shift-a = insert after line
  - escape = return to Command mode
Vim Visual Mode

- Command mode -> Visual mode:
  - \( v \) = visually select spot
  - \( \text{shift-}v \) = visually select line
  - navigate as in command-mode
Running Bash Scripts

• Make executable
• Run Script via 2 methods
Make Executable

- vim create_files.sh

```bash
#!/bin/sh

echo "running script"
```

Run it with this

Make file executable
Running Script

- ./create_files.sh
  - only if chmod +x

```
NathanDunn@unix-biologists% ./create_files.sh
running script ...
```

- /bin/sh create_files.sh

```
NathanDunn@unix-biologists% /bin/sh create_files.sh
running script ...
```
Mercurial Source Control

• Way to track file changes
• A way to share file changes
• A way to backup file changes
Create a Repository

- module load mercurial # on aciss
- Create your own:
  - mkdir myproject1
  - cd myproject1
  - hg init # in directory
  - # edit some files in that directory
  - hg add <files>
  - hg commit
Use a Remote Repository

- Working with another repository:
  - `hg clone`: #how to do a checkout
  - `hg clone ssh://<user>@aciss.uoregon.edu/ <repos> <target_directory>`
  - `hg clone <path to directory> <target_directory>`
  - `hg log` # see what changes
  - `hg diff`
  - `hg push` # push changes to remote repos
  - `hg pull` # pull changes from remote repos
What is ACISS?

• High Performance Computer:
  • 128 Basic Nodes: 12 cores, 72 GB memory
  • 52 GPU Nodes: Same as basic, plus 3 Nvidia 2070 GPUs
  • 16 Fat Nodes: 32 cores, 384 GB memory
  • 400 TB Usable storage
What is ACISSL?

• Both Cluster and Cloud
• For researchers, students, and collaborators
• Free
Cluster

- Shared super-computer
- Configured with job scheduler (qsub)
Cloud

- Personal super-computer(s)
- Create instances (booted machine) as necessary
- Examples: Amazon and ACISS
- Local Example: Virtual Box (demo)
Cloud

Inactive
- Image
- Snapshot

Active
- Instance(s)
- Boot / Launch
- Take Snapshot(s)
- ~ Close Laptop
- Relaunch

ssh
web

Monday, July 30, 12
Cloud vs Cluster

Cloud
- Isolated environment
- No timeline
- Quota on size
- Better for service

Cluster
- Shared environment
- Closer to hardware
- Scheduled
- 1 day, 4 days, 2 week
- Better for batch jobs
Using ACISS Cluster

• Account
• Connect
• Run programs
• Install programs
Cluster Account

• ACISS available to researchers, students, and collaborators.
  • http://prodigal.nic.uoregon.edu:4063/newuser

• Resource: http://aciss.uoregon.edu
Connect to Cluster

- ssh <user>@aciss.uoregon.edu
- ssh -X <user>@aciss.uoregon.edu
  - # to forward graphics
- Can use cyberduck or other sftp, etc. software (or mercurial) to move files
Run Programs

- 2 parallel modes: shared memory and distributed
- 2 run modes: interactive and batch
Shared Memory Parallelism

- Same machine / IP
- Multiple cores / threads
- Some cores can have multiple threads
- Easiest
- Limited to number of cores on processor / node
Distributed Memory Parallelism

- Different machines / IPs
- Usually interacts with multiple cores
- Typically uses MPI and has to run with “mpirun”
- Can be pain to set up and run
- Can use an unlimited number of machines / cores
Run Modes

- Interactive: like ssh’ing into a single node
- Batch: setups up a job in a script and let’s run
Run Interactive

- Get onto node: `qsub -q generic -lX`
  - `-q` is for queue (generic, fatnodes, gpus)
  - `-l` is interactive
  - `-X` is X-forwarding for graphics ... need to `ssh` with `-X` as well .. can be slower if you don’t need it
Run Interactive

- Get onto same node with other terminal?
  
  ```
  [ndunn@hn1 ~]$ qsub -IX -q generic
  qsub: waiting for job 160385.hn1 to start
  qsub: job 160385.hn1 ready
  
  [ndunn@cn166 ~]$  
  ```

- ssh `<user>@aciss.uoregon.edu`

- ssh `cn166`
Run Interactive

• Get onto node: `qsub -q generic -IX`
  
  • `time blastn -query OTTDART00000050916.fa -db /research/sequences/GenBank/blast/db/refseq_rna -out blast1.txt`
  
  • `time blastn -query OTTDART00000050916.fa -db /research/sequences/GenBank/blast/db/refseq_rna -out blast1.txt -num_threads 12`
  
• With another terminal use “top”

• Should see >>100% CPU

• When all terms are done, you lose the node
Run Scheduled Jobs

- small_test.sh

```bash
#!/bin/sh -l
#PBS -N job_name
#PBS -q generic
#PBS -l nodes=1:ppn=4
#PBS -d /home2/ndunn

module load blast

blastn
-query OTTDART00000050916.fa
-db /research/sequences/GenBank/blast/db/refseq_rna
-out .txt
-num_threads 4
```
Run Scheduled Jobs

- Schedule on node
  - setup keys: http://aciss.uoregon.edu/wiki/First_Time_Users
  - qsub small_test.sh
  - qstat
Installing on Cluster

- module
- module avail
- module load <module>
- email Chris / Robert
  - hoge@uoregon.edu
  - ryelle@uoregon.edu
Comsol Example

• Time run over different problem sizes

```bash
time comsol -np 5 ... -
input XXX -output YYY
> time5.txt

time comsol -np 10 ... -
input XXX -output YYY
> time10.txt
```
Bash Conditional

- Conditional: if / then / else

  if [ <condition>]; then
    // stuff
  else
    // other stuff
  fi
Bash For Loop

- “loops”: for loop, while loop

```bash
for i in <array>
do
// stuff here
done

while <condition>
do
// stuff here
done
```
Comsol Interactive

qsub -XI -q generic

module load comsol/4.3
INPUT_FILE="7.17_1_50s.mph"
OUTPUT_DIR="output_dir"
for NP in 1 2 4 8 12 16 20 25 30
do
time comsol batch -np $NP -inputfile $INPUT_FILE
   -outputfile $OUTPUT_DIR/output_$NP  2>&1 > time_$NP.txt
done
Comsol Batch

NP=8
qsub -q generic -l nodes=1:ppn=$NP -o ~/comsole.log
$NP -e ~/comsole.error$NP comsol_command.sh

comsol_command.sh

NP=8
INPUT_FILE="7.17_1_50s.mph"
OUTPUT_DIR="output_dir"
OUTPUT_FILE=
$OUTPUT_DIR/output_batch_$NP.txt

module load comsol/4.3
time comsol batch -np $NP -inputfile $INPUT_FILE
-outputfile $OUTPUT_FILE 2>&1 > time___$NP.txt
Comsol Batch II

qsub comsol_command.sh

comsol_command.sh

#PBS -q generic
#PBS -l nodes=1:ppn=8
#PBS -o ~/comsole.log1 -e ~/comsole.error1
....
NP=8
time comsol batch -np $NP -inputfile $INPUT_FILE
-outputfile $OUTPUT_FILE 2>&1 > time__$NP.txt

Can’t pass in NP variable I think.
Python Example

- pip - package manager
- ACISS shared machine so need virtual environment
- Setting up python libraries:
  - parallel python
  - opby
    - (installs numpy and a few other things)
Python - virtualenv

- Isolated python environment
- http://pypi.python.org/pypi/virtualenv/
- install virtualenv
  - pip install virtualenv # if not on ACISS
- on ACISS:
  - wget https://raw.github.com/pypa/virtualenv/master/virtualenv.py
- python virtualenv.py seg2tosegy
PIP - Python Packages

- http://pypi.python.org/pypi/pip/
- pip - python package manager (installed on ACISS)
- Embedded in virtual system
Python Packages - obspy

- cd seg2tosegy

- http://pypi.python.org/pypi?
  %3Aaction=search&term=obspy&submit=search

  - /bin/pip install obspy.seg2

  - /bin/pip install obspy.segy
Python - obspy

- `cd seg2tosegy`
- `http://pypi.python.org/pypi?%3Aaction=search&term=obspy&submit=search`
  - `./bin/pip install obspy.seg2`
  - `./bin/pip install obspy.segy`
- Run our “special python” that contains packages:
  - `./bin/python`
Python - obspy

- http://docs.obspy.org/packages/autogen/obspy.core.stream.Stream.write.html#obspy.core.stream.Stream.write
- ./bin/python convert2.py

```python
from obspy.core import Stream, read

inputStream = read('example11.seg2', format='SEG2')

inputStream.write('example11.segy', format="SEGY")
```
Python - obspy

- http://docs.obspy.org/packages/autogen/obspy.core.stream.Stream.write.html#obspy.core.stream.Stream.write

- ./bin/python convert2.py

```python
from obspy.core import Stream, read

inputStream = read('example11.seg2', format='SEG2')

inputStream.write('example11.segy', format="SEGY")
```
Matlab on Cluster

- Launch X11 environment (Linux Default, OSX Program)
- `ssh -X <user>@aciss.uoregon.edu`
- `qsub -lX -q generic`
- `module load matlab`
- `matlab -nodisplay`
  - # no plotting
- `matlab -nodekstop`
  - # allows plots
- Other options: http://www.mathworks.com/help/techdoc/ref/matlabunix.html
Matlab Example

- File: partest.m
- `matlab -nodisplay -r partest`

```matlab
clear A
d = 0; i = 0;
for i = 1:4
    d = i*2;
    A(i) = d;
end
A
d
i
```

```matlab
parfor i = 1:4
    d = i*2;
    A(i) = d;
end
A
d
i
```
Matlab Example

- Each member of the loop sent out to each processor

```matlab
clear A

A = [2 4 6 8]
d = 0; i = 0;
for i = 1:4
    d = i*2;
    A(i) = d;
end

A = [2 4 6 8]
d = 0; i = 0;
parfor i = 1:4
    d = i*2;
    A(i) = d;
end
```

- A = 2 4 6 8
- d = 0, i = 0
- A = 2 4 6 8
- d = 8, i = 4
Matlab MPI?

• Craig is currently doing a lot of work with MPI and Matlab (and Python), so I will let him talk about it on the 15th, though will be happy to look at problems.

• Craig runs an MPI workshop:

  • [http://blogs.uoregon.edu/casspr/2012/05/21/mpi-workshop/](http://blogs.uoregon.edu/casspr/2012/05/21/mpi-workshop/)
Code Access

- On ACISS:
  - hg clone /home2/ndunn/hg/unix_geo_week3

- Remote:
  - hg clone ssh://<user>@aciss.uoregon.edu:hg/unix_geo_week3
Other Resources

• Do it!
• Google it!
• Ask!
• Conery Class: CIS 170
Next Time

• ???