SCALABLE COMPUTING LABORATORY

VANDERBILT UNIVERSITY

SC3260 / SC5260

Running on HPC

Lecture by: Ana Gainaru

Slides based on the VU ACCRE tutorials

Table of contents

Using an HPC system efficiently

- Connecting to the HPC system
- Transferring files
- Scheduling jobs
- Accessing software
- Post-running analysis

ACCRE

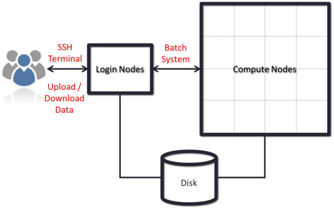
```
======================================================================

Go forth and compute!
```

Vanderbilt's cluster

- Request an account: https://www.accre.vanderbilt.edu/?page_id=3563
  - For Group select "SC3260/5260"
- Please allow a few business days for your account to be approved
- Once you are approved, you will need to set up your ACCRE login and change your password.
  - https://www.vanderbilt.edu/accre/getting-started/first-time-account-setup/

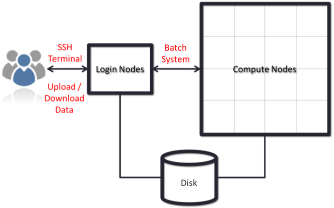
Connecting to the HPC system

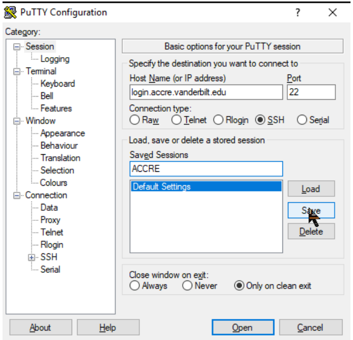
- Most often done through a tool known as "SSH" (Secure SHell)
  - Linux/Mac through a terminal
  - Windows through applications like PuTTY or MobaXterm



Connecting to the HPC system

- Most often done through a tool known as "SSH" (Secure SHell)
  - Linux/Mac through a terminal
  - Windows through applications like PuTTY or MobaXterm




PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): login.accre.vanderbilt.edu

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: ACCRE

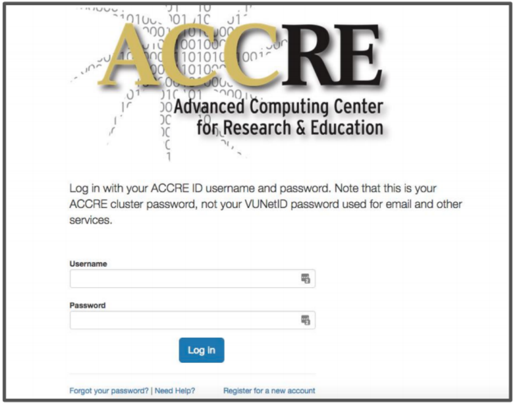
Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

```
ssh vunetid@login.accre.vanderbilt.edu
```

Portal

Log into the Visualization Portal using a browser:
https://portal.accre.vanderbilt.edu


ACCRE
Advanced Computing Center
for Research & Education

Log in with your ACCRE ID username and password. Note that this is your ACCRE cluster password, not your VUNetID password used for email and other services.

Username

Password

Log in
Forgot your password? | Need Help?
Register for a new account

You can also access a terminal from the portal

Examining the nodes

```
[gainara@gw346 ~]$ lscpu
```

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                72
On-line CPU(s) list:   0-71
Thread(s) per core:    2
Core(s) per socket:    18
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 79
Model name:            Intel(R) Xeon(R\frametitle{Transferring files})
                       CPU E5-2695 v4 @ 2.10GHz
Stepping:              1
CPU MHz:               1199.953
BogoMIPS:              4205.47
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              46080K
NUMA node0 CPU(s):     0-17,36-53
NUMA node1 CPU(s):     18-35,54-71
```

- The processor model is: Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
- There are 18 cores per socket
  Thread(s) per socket: 36
- There are two processors per node: Sockets: 2
- This means that there are 2 * 18 = 36 cores on the node

This is the configuration of login nodes. For compute nodes we use an interactive job

Examining the nodes

```
[gainara@gw346 ~]$ head —n1 /proc/meminfo

MemTotal:          263772152 kB
```

- This tells us that there are approximately 252 GB of memory available
  - 263772152/[1024*1024] = 251.55 GB
  - This node has 256 GB, 4GB are reserved for various parts of computing hardware

This is the configuration of login nodes. For compute nodes we use an interactive job

Note: the login nodes are just as interface to compute nodes (compile, debug, test for small values).

Large code execution will be done only on compute nodes.

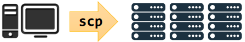
- Small runs can be done on the login nodes, remember that it's shared

Transferring files

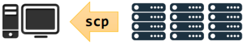
- Grabbing files from the Internet

```
[gainara@gw346 ~]$ wget https://epcced.github.io/hpc—intro/files/cfd.tar.gz
```

- Transferring single files and folders with scp



```
scp local_path vunetid@login.accre.vanderbilt.edu:remote_path
```



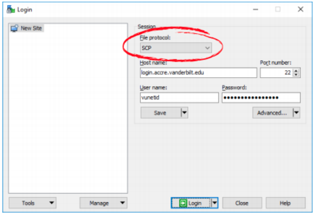
```
scp vunetid@login.accre.vanderbilt.edu:remote_path local_path
```

If you prefer a graphical interface, you can use FileZilla.

Transferring files in Windows

WinSCP

https://winscp.net



Login

New Site

Session

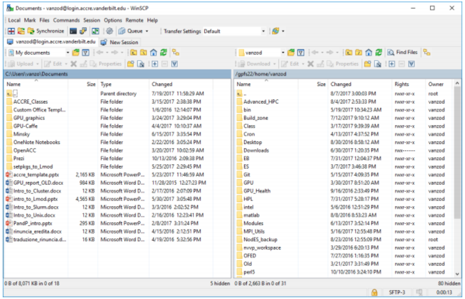
File protocol: SCP

Host name: login.accre.vanderbilt.edu

Port number: 22

User name: vunetid

Password: ••••••••••••••••



Documents - vanzod@login.accre.vanderbilt.edu - WinSCP

C:\Users\vanzo\Documents

Name	Size	Type	Changed
..		Parent directory	7/19/2017 11:58:29 AM
ACCRE_Classes		File folder	3/15/2017 2:38:38 PM
Custom Office Templ...		File folder	1/6/2016 12:14:07 PM
GPU_graphics		File folder	3/24/2017 3:29:04 PM
GPU-Caffe		File folder	4/4/2017 10:10:37 AM
Minsky		File folder	6/15/2017 3:35:54 PM
OneNote Notebooks		File folder	2/22/2016 3:05:24 PM
OpenACC		File folder	3/20/2017 10:02:59 AM
Prezi		File folder	10/13/2016 2:09:38 PM
setpkgs_to_Lmod		File folder	5/25/2017 2:29:45 PM
accre_template.pptx	2,165 KB	Microsoft PowerP...	5/23/2017 11:46:59 AM
GPU_report_OLD.docx	984 KB	Microsoft Word D...	11/28/2015 12:27:23 PM
Intro_to_Cluster.docx	12 KB	Microsoft Word D...	2/17/2016 2:07:09 PM
intro_to_Lmod.pptx	4,565 KB	Microsoft PowerP...	5/30/2017 3:05:48 PM
Intro_to_Slurm.docx	12 KB	Microsoft Word D...	3/3/2016 2:02:52 PM
Intro_to_Unix.docx	12 KB	Microsoft Word D...	2/16/2016 12:23:41 PM
PandP_intro.pptx	295 KB	Microsoft PowerP...	2/8/2017 3:31:24 PM
rinuncia_eredita.docx	12 KB	Microsoft Word D...	4/15/2016 2:12:51 PM
traduzione_rinuncia.d...	16 KB	Microsoft Word D...	4/19/2016 5:32:56 PM

0 B of 8,071 KB in 0 of 18 — 5 hidden

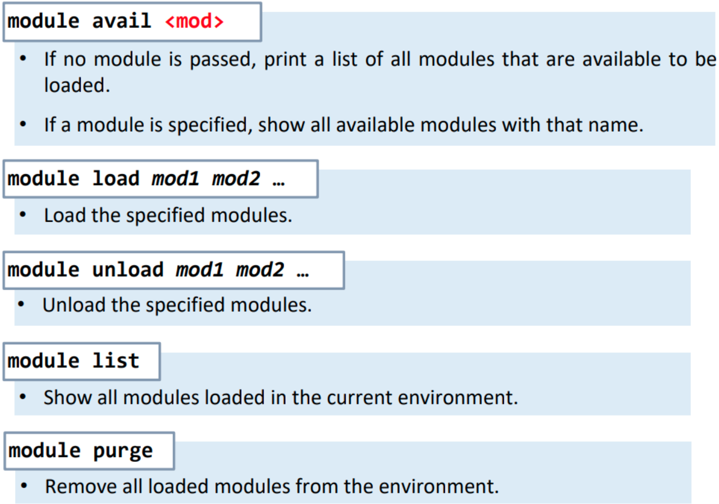
/gpfs22/home/vanzod

Name	Size	Changed	Rights	Owner
..		8/7/2017 3:00:03 PM	rwxr-xr-x	root
Advanced_HPC		8/4/2017 2:53:33 PM	rwxr-xr-x	vanzod
bin		5/19/2017 10:34:23 AM	rwxr-xr-x	vanzod
Build_zone		7/12/2017 9:10:12 AM	rwxr-xr-x	vanzod
Class		3/17/2017 9:39:39 AM	rwxr-xr-x	vanzod
Cron		4/13/2017 4:37:52 PM	rwxr-xr-x	vanzod
Desktop		8/30/2016 8:58:12 AM	rwxr-xr-x	vanzod
Downloads		6/30/2017 1:20:35 PM	rwx------	vanzod
EB		7/31/2017 12:04:37 PM	rwxr-xr-x	vanzod
ES		3/7/2017 3:46:38 PM	rwxr-xr-x	vanzod
Git		5/15/2017 4:09:35 PM	rwxr-xr-x	vanzod
GPU		3/30/2017 8:51:20 AM	rwxr-xr-x	vanzod
GPU_Health		9/16/2016 2:33:49 PM	rwxr-xr-x	vanzod
HPL		7/31/2017 5:28:17 PM	rwxr-xr-x	vanzod
intel		5/6/2016 12:51:29 PM	rwxr-xr-x	vanzod
matlab		8/8/2016 8:53:23 AM	rwxr-xr-x	vanzod
Modules		6/13/2017 3:52:14 PM	rwxr-xr-x	vanzod
MPI_Utils		5/16/2017 12:55:48 PM	rwxr-xr-x	vanzod
NodES_backup		8/23/2016 12:55:09 PM	rwxr-xr-x	root
nvvp_workspace		3/29/2016 6:20:13 PM	rwxr-xr-x	vanzod
OFED		7/27/2016 1:16:35 PM	rwxr-xr-x	vanzod
Old		3/21/2017 3:31:49 PM	rwxr-xr-x	vanzod
perl5		10/10/2016 3:24:10 PM	rwxr-xr-x	vanzod

0 B of 2,663 B in 0 of 31 — 80 hidden

SFTP-3 — 0:00:13

Use existing software



```
module avail <mod>
```

- If no module is passed, print a list of all modules that are available to be loaded.
- If a module is specified, show all available modules with that name.

```
module load mod1 mod2 …
```

- Load the specified modules.

```
module unload mod1 mod2 …
```

- Unload the specified modules.

```
module list
```

- Show all modules loaded in the current environment.

```
module purge
```

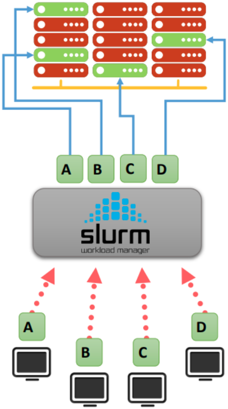
- Remove all loaded modules from the environment.

VANDERBILT
UNIVERSITY

Scheduler

1. Execute user's workloads in the right priority order
2. Provide requested resources on compute nodes
3. Optimize cluster utilization

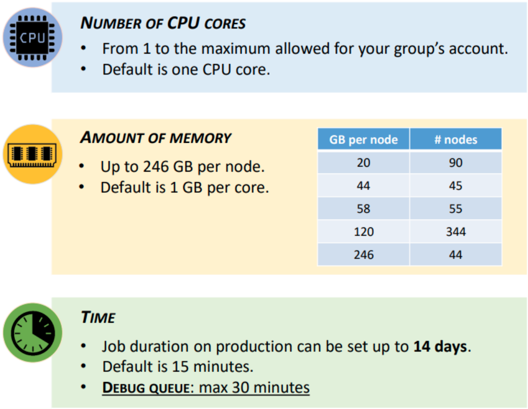
⚠ **Users do not access compute nodes directly!**



Scheduling jobs

- Choosing a text editor
  - For new users ACCRE recommends nano, which is simple and easy to use (vim and emacs are also available)
  - Either transfer your py or c file to ACCRE or create your file with nano, `nano file.c`. To close `nano`, press `Control-X` or `Command-X`.
- Compile your code using gcc
- Check to see if your code works for small values
- Determine how many resources your large run might need
- Write the SLURM script
- Run the SLURM script
  - This will submit start your job on a subset of the compute nodes
  - You can query the state of your job
  - When the job is finished, you can check the results

Determine how many resources you need



Number of CPU cores

- From 1 to the maximum allowed for your group's account.
- Default is one CPU core.

Amount of memory

- Up to 246 GB per node.
- Default is 1 GB per core.

GB per node	# nodes
20	90
44	45
58	55
120	344
246	44

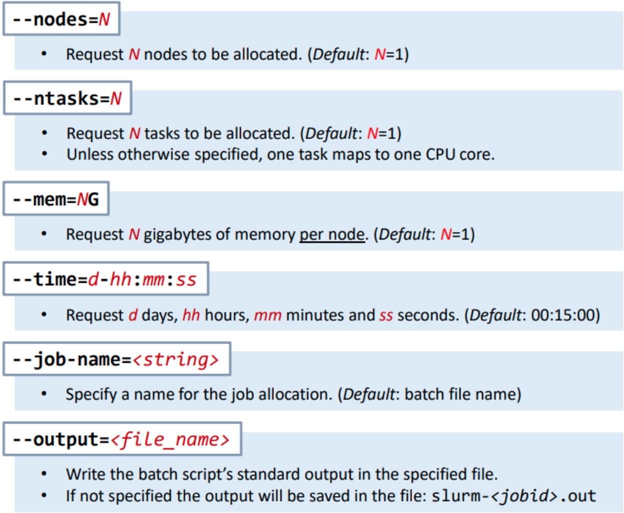
Time

- Job duration on production can be set up to **14 days**.
- Default is 15 minutes.
- Debug queue: max 30 minutes

Slightly overestimate the requested job resources, but do not greatly overestimate to avoid unnecessary long wait times.

Slurm will immediately kill your job if your process exceeds the requested amount of resources.

Write the SLURM script



`--nodes=`*N*

- Request *N* nodes to be allocated. (*Default*: *N*=1)

`--ntasks=`*N*

- Request *N* tasks to be allocated. (*Default*: *N*=1)
- Unless otherwise specified, one task maps to one CPU core.

`--mem=`*N*`G`

- Request *N* gigabytes of memory per node. (*Default*: *N*=1)

`--time=`*d*`-`*hh*`:`*mm*`:`*ss*

- Request *d* days, *hh* hours, *mm* minutes and *ss* seconds. (*Default*: 00:15:00)

`--job-name=`*<string>*

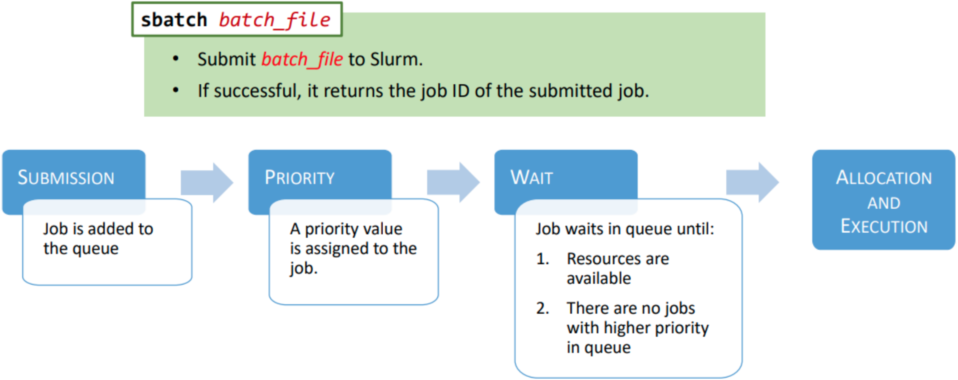
- Specify a name for the job allocation. (*Default*: batch file name)

`--output=`*<file_name>*

- Write the batch script's standard output in the specified file.
- If not specified the output will be saved in the file: `slurm-`*<jobid>*`.out`

```
#!/bin/bash
#SBATCH ——nodes=1
#SBATCH ——ntasks=1
#SBATCH ——mem=2G
#SBATCH ——time=0:20:00
#SBATCH ——job—name=myjob
#SBATCH ——output=pi.txt

module load GCC Python
python file.py parameters
```

Submit job to the scheduler



```
sbatch batch_file
```

- Submit *batch_file* to Slurm.
- If successful, it returns the job ID of the submitted job.

Submission

Job is added to the queue

Priority

A priority value is assigned to the job.

Wait

Job waits in queue until:

1. Resources are available
2. There are no jobs with higher priority in queue

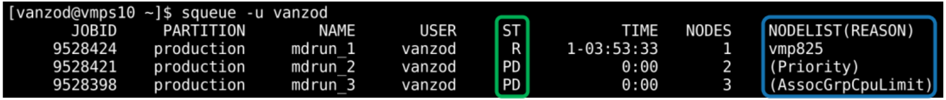
Allocation and Execution

To cancel a job submission, `scancel jobID`

Check the status of a job

`squeue -u vunetid`

- Show the queued jobs for user *vunetid*.



```
[vanzod@vmps10 ~]$ squeue -u vanzod
   JOBID    PARTITION       NAME      USER   ST          TIME   NODES  NODELIST(REASON)
 9528424   production    mdrun_1    vanzod    R    1-03:53:33       1  vmp825
 9528421   production    mdrun_2    vanzod   PD          0:00       2  (Priority)
 9528398   production    mdrun_3    vanzod   PD          0:00       3  (AssocGrpCpuLimit)
```

Status

R = Running

PD = Pending

CA = Cancelled

Nodelist (Reason)

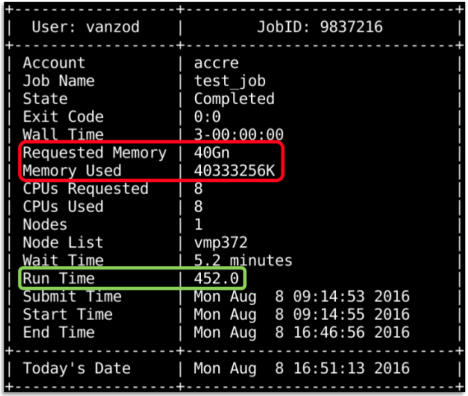
- For running jobs shows the allocated nodes.
- For pending jobs shows the wait reason:

Priority	Other jobs in queue have higher priority.
Resources	Insufficient resources available on the cluster.
AssocGrpCpuLimit	Reached maximum number of allocated CPUs by all jobs belonging to the user's account.
AssocGrpMemLimit	Reached maximum amount of allocated memory by all jobs belonging to the user's account.
AssocGrpTimeLimit	Reached maximum amount of allocated time by all jobs belonging to the user's account.

Check the status of a job

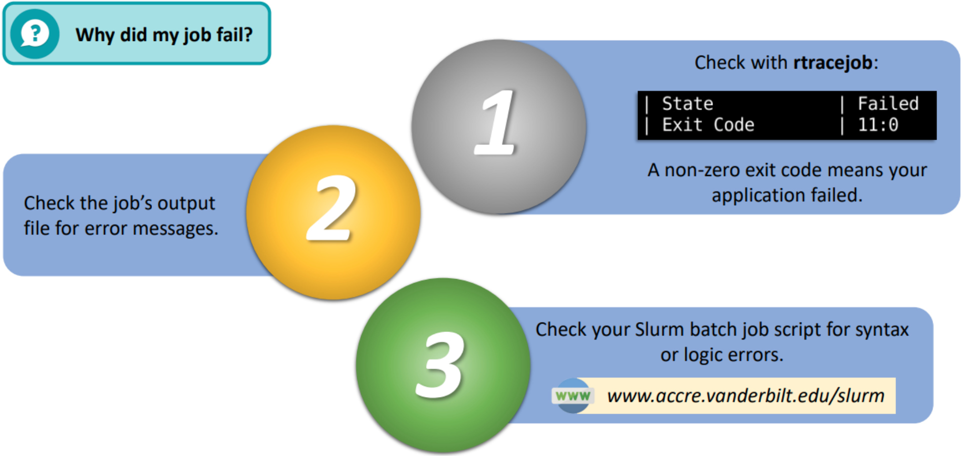
rtracejob *jobid*

- Print requested and utilized resources (and more) for the given *jobid*.



```
+------------------+------------------------------+
|  User: vanzod    |        JobID: 9837216        |
+------------------+------------------------------+
| Account          | accre                        |
| Job Name         | test_job                     |
| State            | Completed                    |
| Exit Code        | 0:0                          |
| Wall Time        | 3-00:00:00                   |
| Requested Memory | 40Gn                         |
| Memory Used      | 40333256K                    |
| CPUs Requested   | 8                            |
| CPUs Used        | 8                            |
| Nodes            | 1                            |
| Node List        | vmp372                       |
| Wait Time        | 5.2 minutes                  |
| Run Time         | 452.0                        |
| Submit Time      | Mon Aug  8 09:14:53 2016     |
| Start Time       | Mon Aug  8 09:14:55 2016     |
| End Time         | Mon Aug  8 16:46:56 2016     |
+------------------+------------------------------+
| Today's Date     | Mon Aug  8 16:51:13 2016     |
+------------------+------------------------------+
```

Check the exit code of finished jobs



Why did my job fail?

1 Check with **rtracejob**:

```
| State          | Failed
| Exit Code      | 11:0
```

A non-zero exit code means your application failed.

2 Check the job's output file for error messages.

3 Check your Slurm batch job script for syntax or logic errors.

www.accre.vanderbilt.edu/slurm