Introduction to Ada

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

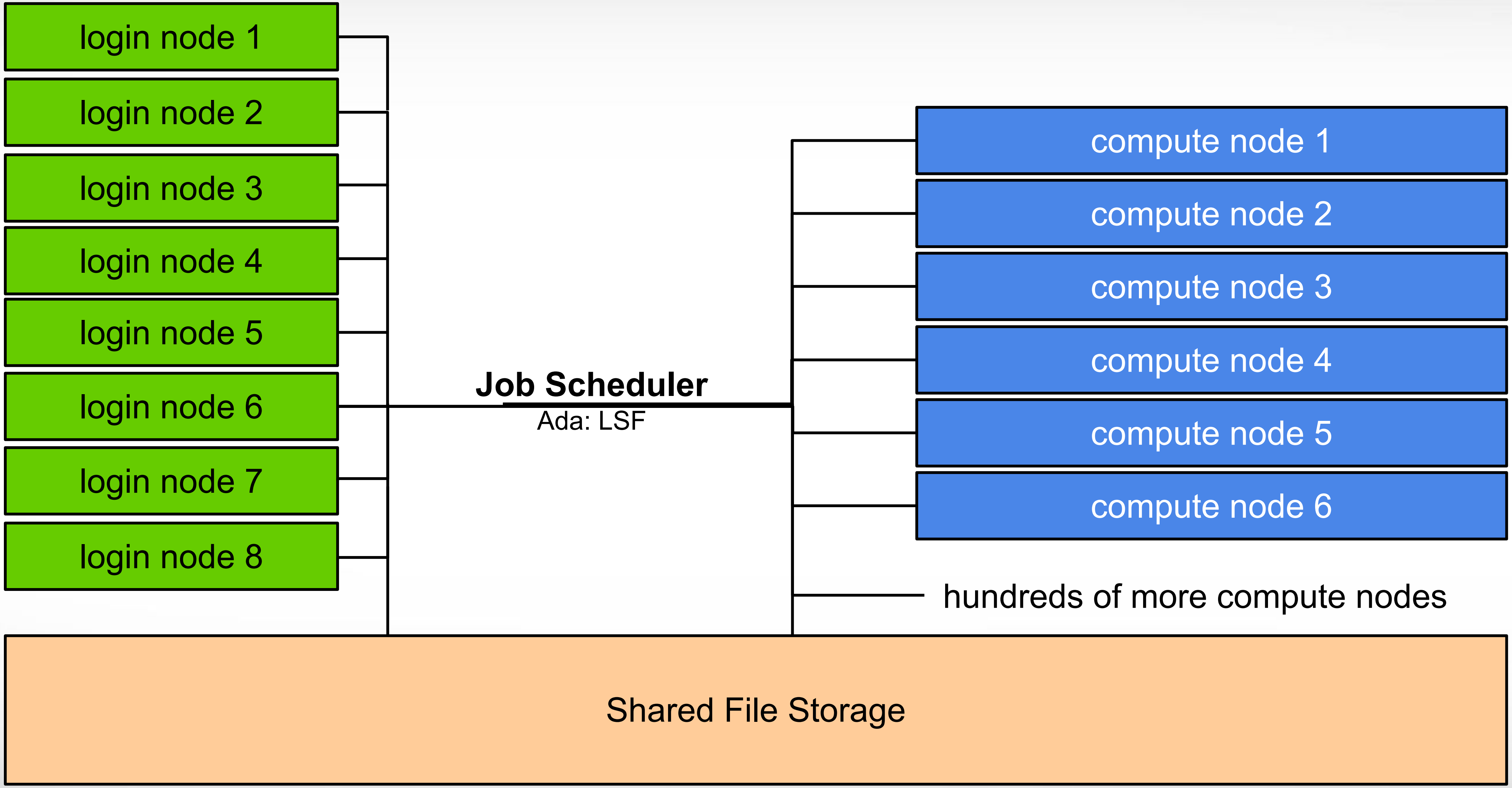
Spring 2021

Usage Policies
(Be a good compute citizen)

- It is illegal to share computer passwords and accounts by state law and university regulation
- It is prohibited to use Ada in any manner that violates the United States export control laws and regulations, EAR & ITAR
- Abide by the expressed or implied restrictions in using commercial software
  - almost all bioinformatics software on Ada is open source

https://hprc.tamu.edu/policies

HPC Diagram



Ada Compute nodes on HPRC clusters are not connected to the internet

Ada Clusters Specs

Compute Nodes	860	cores per node	CPU speed
Total Cores	17,500	-	-
64GB memory nodes	811	20 cores per node	2.50 GHz
256 GB memory nodes	26	20 cores per node	2.50 GHz
1TB memory nodes*	11	40 cores per node	2.27 GHz
2TB memory nodes*	4	40 cores per node	2.27 GHz
login nodes	8	20 cores per node	2.50 GHz
Job Scheduler	LSF	-	-
File System	4 PB raw	-	-
Production Date	Sep 2014	-	-

* use large memory nodes only for large memory jobs

https://hprc.tamu.edu/resources

Accessing Ada

- SSH command is required for accessing Ada / Terra:
  - On campus: `ssh NetID@ada.tamu.edu`
  - Off campus:
    - Set up and start VPN (Virtual Private Network): http://u.tamu.edu/VPnetwork
    - Then: `ssh NetID@ada.tamu.edu`
  - SSH programs for Windows:
    - MobaXTerm (preferred, includes SSH and X11)
- Access through https://portal.hprc.tamu.edu (Menu “Clusters” => “Ada Shell Access”)
- Ada has 8 login nodes. Check the bash prompt to see which you log into.
  -
    ```
    [NetID@ada1 ~]$
    ```
- Login sessions that are idle for **60** minutes will be closed automatically
- Processes run longer than **60** minutes on login nodes will be killed automatically.
- **Do not use more than 8 cores on the login nodes!**
- **Do not use the sudo command.** Contact us for assistance installing software.

https://hprc.tamu.edu/wiki/HPRC:Access

File Transfers with **Ada**

- Simple File Transfers:
  - scp: command line (Linux, MacOS)
  - rsync: command line (Linux, MacOS); ***can resume transfer***
  - MobaXterm: GUI (Windows)
  - WinSCP: GUI (Windows)
  - Portal: https://portal.hprc.tamu.edu
- Bulk data transfers (large files):
  - Use fast transfer nodes
    - data transfer processes will not timeout at 60 minutes
    - `ssh ada-ftn1.tamu.edu` OR `ssh ada-ftn2.tamu.edu`
    - Globus Connect (https://hprc.tamu.edu/wiki/SW:GlobusConnect)
    - GridFTP

https://hprc.tamu.edu/wiki/HPRC:FileTransfers

File Systems and User Directories

Directory	Environment Variable	Space Limit	File Limit	Intended Use
/home/$USER	$HOME	10 GB	10,000	Small to modest amounts of processing.
/scratch/user/$USER	$SCRATCH	1 TB	250,000	Temporary storage of large files for on-going computations. Not intended to be a long-term storage area.
/tiered/user/$USER	$ARCHIVE	10 TB	50,000	Intended to hold valuable data files that are not frequently used

- **`$HOME`** and **`$SCRATCH`** directories are not shared between Ada and Terra clusters.
- View usage and quota limits using the command: **`showquota`**
- Quota and file limit increases will only be considered for scratch and tiered directories
- Request a group directory for sharing files.
- Do not share your home, scratch, tiered directories.

https://hprc.tamu.edu/wiki/Ada:Filesystems_and_Files

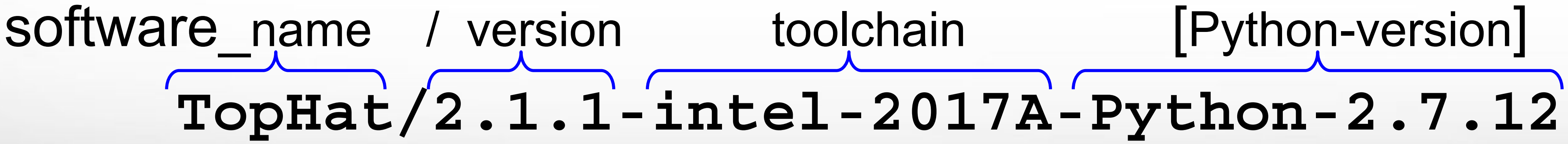
Software

- See the Software wiki page for instructions and examples
  - https://hprc.tamu.edu/wiki/SW
  - https://hprc.tamu.edu/software/ada
  - https://hprc.tamu.edu/wiki/Bioinformatics
- License-restricted software
  - Contact license owner for approval
- Contact us for software installation help/request
  - User can install software in their $SCRATCH dir
  - Do not run the "*sudo*" command when installing software

Computing Environment

- Paths:
  - $PATH: for commands (eg. /bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/netid/bin)
  - $LD_LIBRARY_PATH: for libraries
  - See your $PATH variable with the command `echo $PATH`
- There is a lot of software, many versions, and many paths to manage

  .... How do you manage all these software versions?
- The solution (lmod) which uses the command: **`module`**
- Almost all software, application, library, etc. is available as a module.
  - Module names have the format:



software_name / version toolchain [Python-version]

```
TopHat/2.1.1-intel-2017A-Python-2.7.12
```

https://hprc.tamu.edu/wikiAda:Computing_Environment#Modules

Application Modules

- Installed applications are available as modules which are available to all users
  - (except for restricted modules)
- It's a good habit to purge unused modules before loading new modules.
- It is highly recommended to load a specific software version instead of the defaults
- **Avoid loading modules in your `~/.bashrc`**

```
module avail                # list all available modules (sometimes it is very slow)
                            # space bar down, page up/down, q to quit
                            # / for case sensitive search (similar to a UNIX man page)

module spider boost         # case insensitive search for modules with ‘boost’ in name

module keyword graphics     # search module descriptions for keyword ‘graphics’
                            # some graphics modules may be missed if
                            # keyword is not found in description (case insensitive)
```

https://hprc.tamu.edu/wiki/Ada:Computing_Environment#Modules

Module Loading Exercise

1.
   ```
   module list                 # list all loaded modules
   ```
2.
   ```
   module spider trinity       # search for available module names matching trinity
                               # not case sensitive unless an exact match is found
   ```
3.
   ```
   module load Trinity/2.5.1-GCCcore-6.3.0-Perl-5.24.0    # load specific module version
                                                          # type Trinity/2.5 then hit tab key
                                                          # or copy and paste
   ```
4.
   ```
   module list                 # list all loaded modules
   ```
5.
   ```
   module purge                # remove all loaded modules
   ```

Modules and Toolchains

- Load modules with the same toolchains in your job scripts
- The **`2018b`** and **`GCCcore-7.3.0`** toolchain versions are recommended
    - `GCCcore/7.3.0` **and** `(intel/2018b` **or** `iomkl/2018b` **or** `foss/2018b)`

- Avoid loading modules in your .bashrc and .bash_profile files
- Avoid mixing toolchains if loading multiple modules in the same job script

```
module load HISAT2/2.0.4-foss-2016b
module load TopHat/2.1.1-intel-2017A-Python-2.7.12
module load Cufflinks/2.2.1-intel-2015B
```

- Same rule applies to compilers and libraries.

The GCCcore Toolchain

- To minimize the number of software builds, the GCCcore7.3.0 toolchain modules can be loaded alone or with any one of the following 2018b toolchains
  - `intel/2018b`
  - `iomkl/2018b`
  - `foss/2018b`
- Example of loading a GCCcore-7.3.0 module with a 2018b module

```
module load BEDTools/2.29.0-GCCcore-7.3.0
module load AdapterRemoval/2.2.2-foss-2018b
```

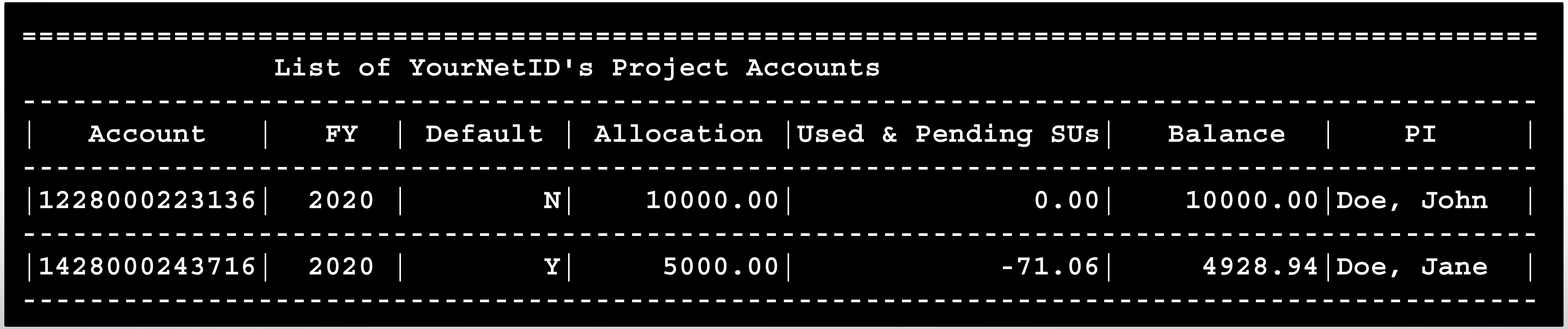
- See list of compatible toolchains

```
toolchains
```

Consumable Computing Resources

- Resources specified in a job file:
  - Processor cores
  - Memory
  - Wall time
  - GPU
- Service Unit (SU) - defined as one CPU core usage for one hour
  - Use "**`myproject`**" to see your balance

```
myproject
```



```
========================================================================================
              List of YourNetID's Project Accounts
----------------------------------------------------------------------------------------
|   Account    |   FY   | Default | Allocation |Used & Pending SUs|   Balance   |    PI     |
----------------------------------------------------------------------------------------
|1228000223136|  2020  |        N|    10000.00|              0.00|     10000.00|Doe, John  |
----------------------------------------------------------------------------------------
|1428000243716|  2020  |        Y|     5000.00|            -71.06|      4928.94|Doe, Jane  |
----------------------------------------------------------------------------------------
```

https://hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit

Ada: Examples of SUs charged based on Job Cores, Time and Memory Requested

A Service Unit (SU) on Ada is equivalent to one core or **2500** MB memory usage for one hour.

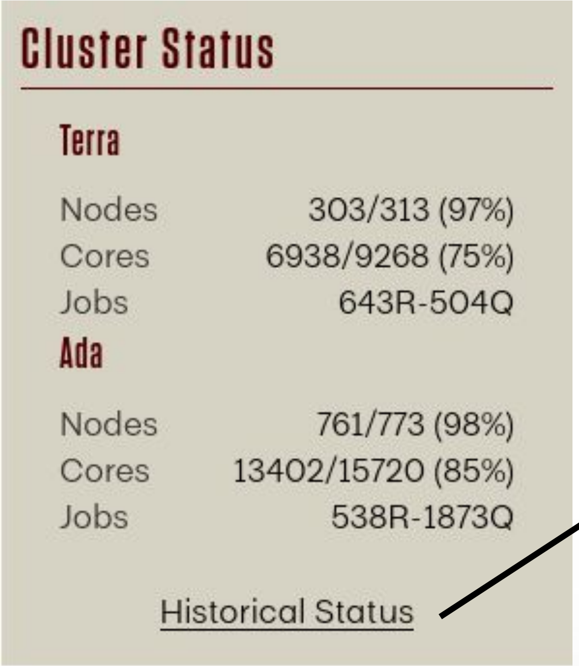
	Number of Cores	MB of memory per core	Total Memory (GB)	Hours	SUs charged
1.	1	2500	2.5	1	1
2.	1	2600	2.6	1	2
3.	1	50000	50	1	20
4.	20	2500	50	1	20
5.	20	2700	54	1	20

- SUs are charged at job submit time and the job's unused SUs are reimbursed if job finishes early

https://hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit

Historical HPRC Cluster Usage

hprc.tamu.edu



Cluster Status

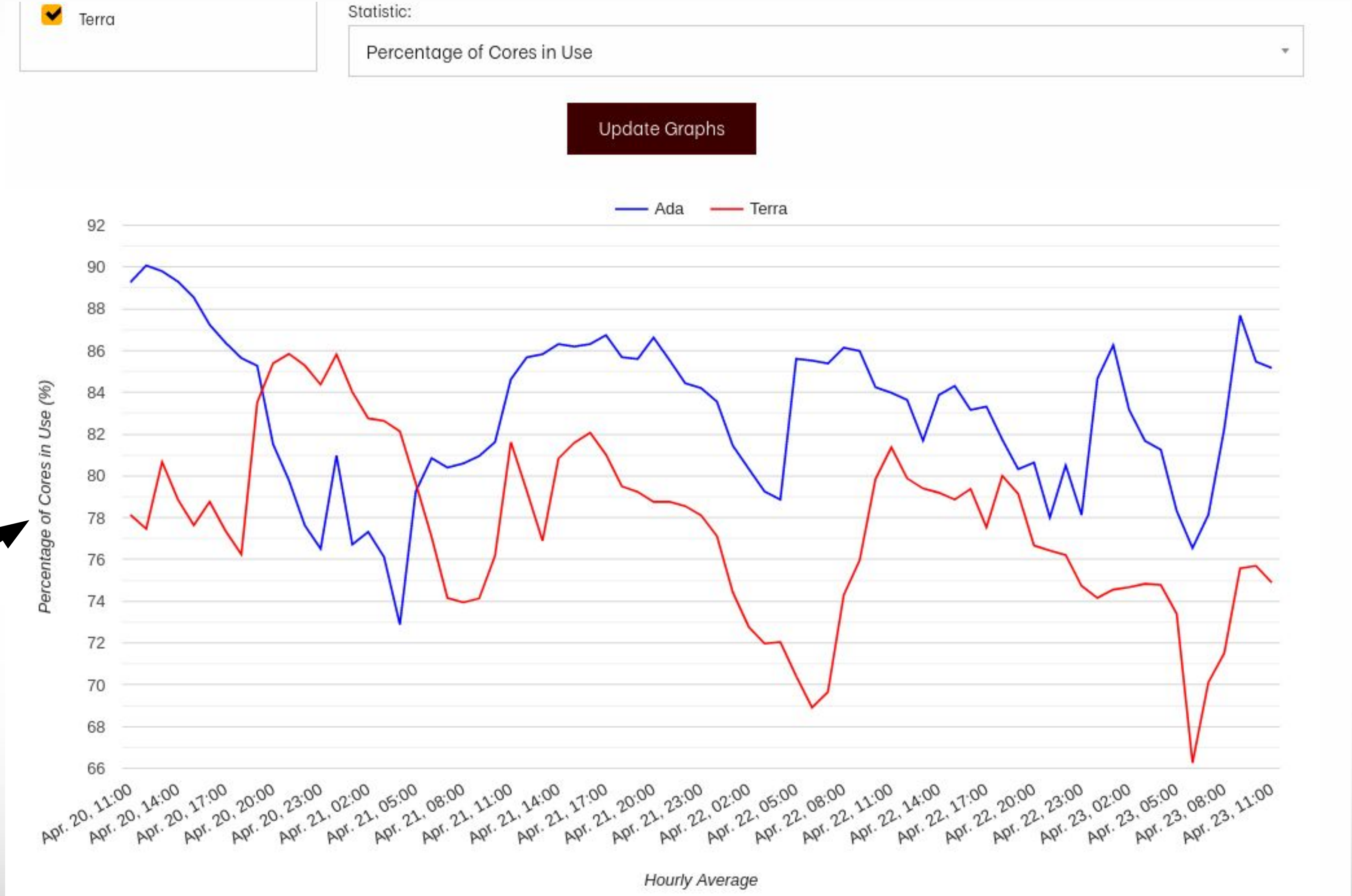
Terra

Nodes	303/313 (97%)
Cores	6938/9268 (75%)
Jobs	643R-504Q

Ada

Nodes	761/773 (98%)
Cores	13402/15720 (85%)
Jobs	538R-1873Q

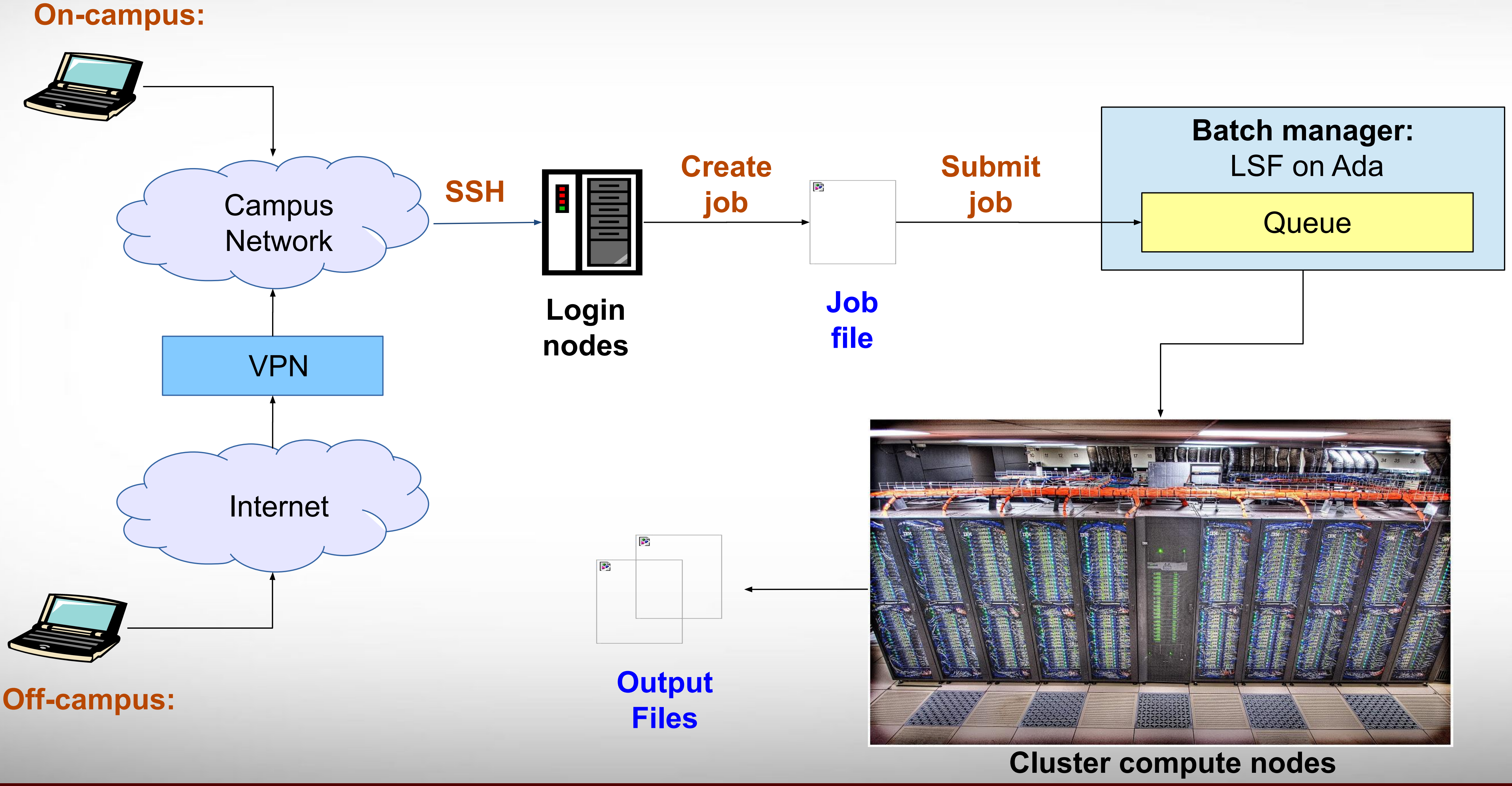
Historical Status



☑ Terra

Statistic: Percentage of Cores in Use

Batch Computing on HPRC Clusters



On-campus:

Campus Network

VPN

Internet

Off-campus:

SSH

Login nodes

Create job

Job file

Submit job

Batch manager:
LSF on Ada

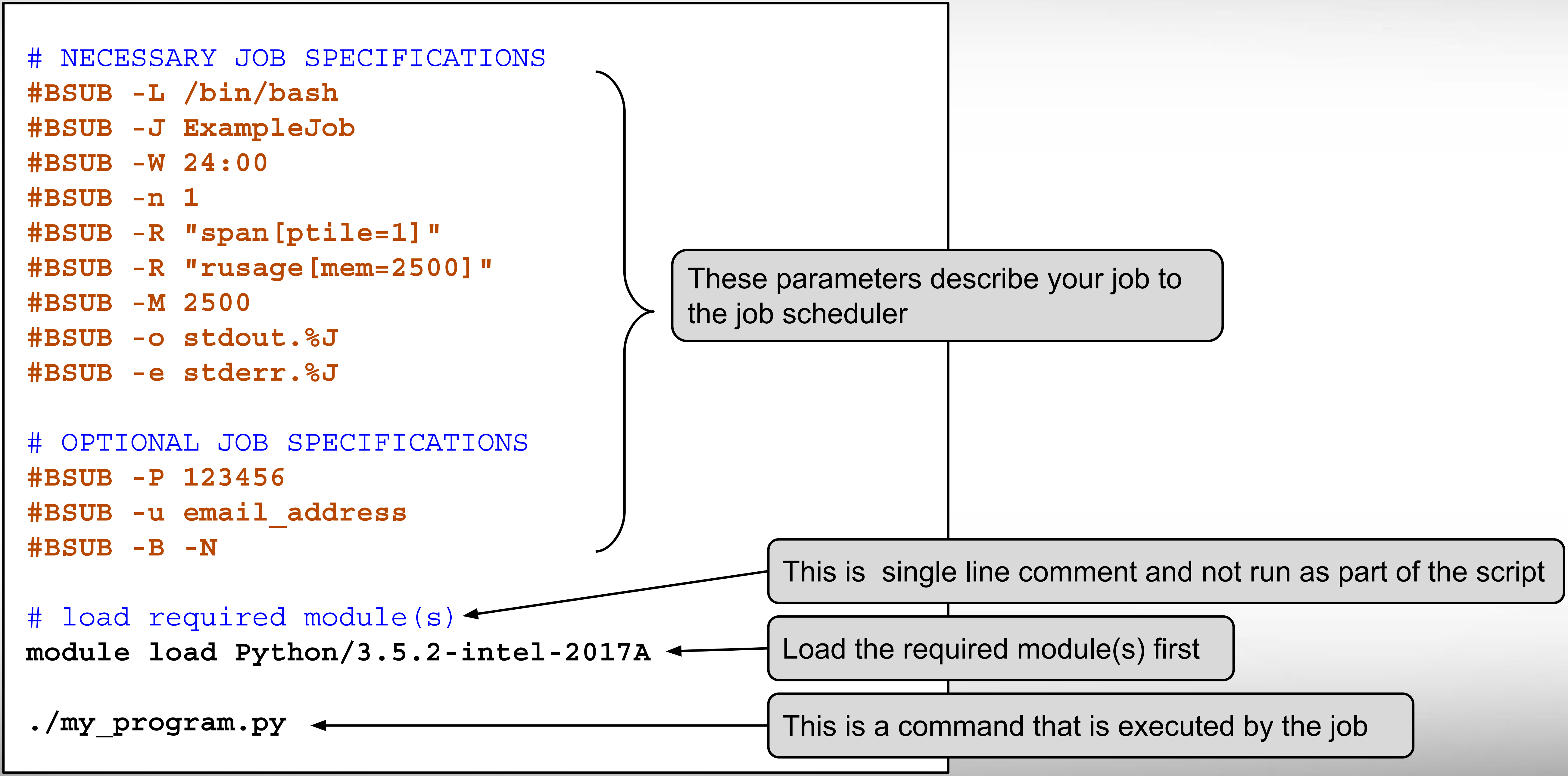
Queue

Cluster compute nodes

Output Files

Batch Job Scripts

Sample Job Script Structure



```
# NECESSARY JOB SPECIFICATIONS
#BSUB -L /bin/bash
#BSUB -J ExampleJob
#BSUB -W 24:00
#BSUB -n 1
#BSUB -R "span[ptile=1]"
#BSUB -R "rusage[mem=2500]"
#BSUB -M 2500
#BSUB -o stdout.%J
#BSUB -e stderr.%J

# OPTIONAL JOB SPECIFICATIONS
#BSUB -P 123456
#BSUB -u email_address
#BSUB -B -N

# load required module(s)
module load Python/3.5.2-intel-2017A

./my_program.py
```

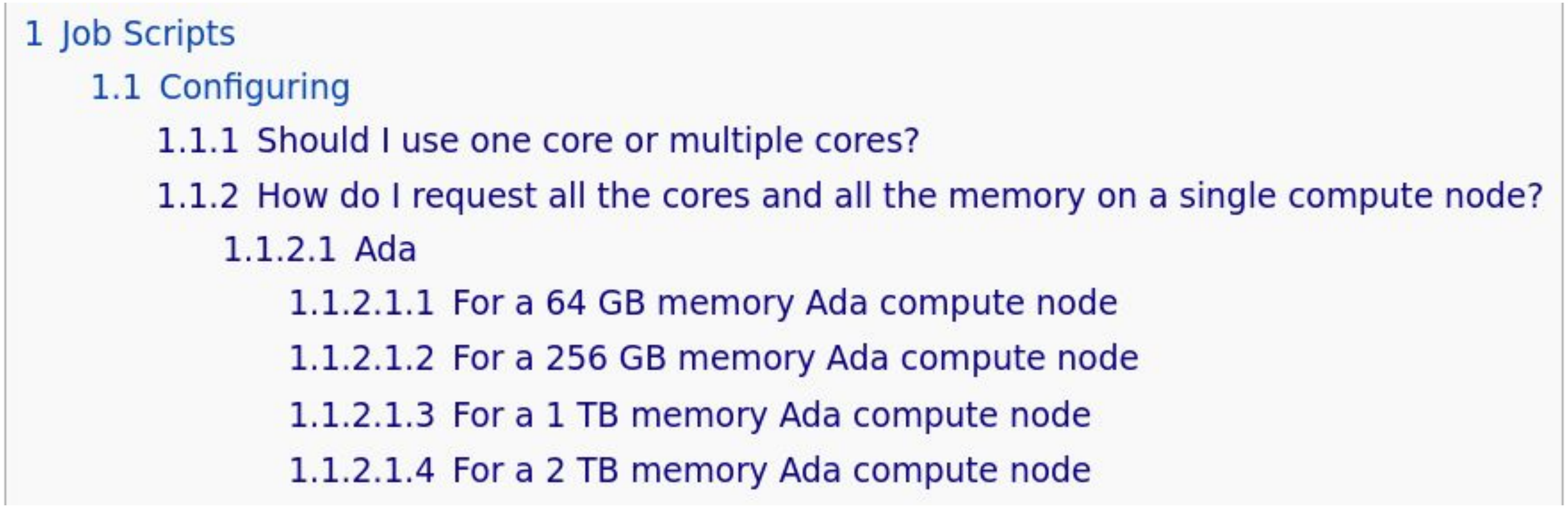
These parameters describe your job to the job scheduler

This is single line comment and not run as part of the script

Load the required module(s) first

This is a command that is executed by the job

See the HPRC Wiki on how to request all cores and all memory on a compute node



1 Job Scripts
- 1.1 Configuring
  - 1.1.1 Should I use one core or multiple cores?
  - 1.1.2 How do I request all the cores and all the memory on a single compute node?
    - 1.1.2.1 Ada
      - 1.1.2.1.1 For a 64 GB memory Ada compute node
      - 1.1.2.1.2 For a 256 GB memory Ada compute node
      - 1.1.2.1.3 For a 1 TB memory Ada compute node
      - 1.1.2.1.4 For a 2 TB memory Ada compute node

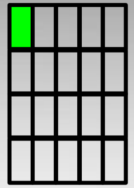
- Use all cores if the software you are using supports multi-core unless the software recommends fewer cores
- Use all cores if you are using all the memory even if the software only supports running on a single core

https://hprc.tamu.edu/wiki/Bioinformatics:FAQ

Example Batch Job Scripts

https://hprc.tamu.edu/wiki/Ada:Batch_Processing_LSF#Job_File_Examples

Ada Job File (Serial Example)



```
# NECESSARY JOB SPECIFICATIONS
#BSUB -L /bin/bash                # Uses bash to initialize the job's execution environment.
#BSUB -J ExampleJob1              # Set the job name to "ExampleJob1"
#BSUB -W 24:00                    # Set the wall clock limit to 24hr
#BSUB -n 1                        # Request 1 core
#BSUB -R "span[ptile=1]"          # Request 1 core per node.
#BSUB -R "rusage[mem=2500]"       # Request 2500MB per process (CPU) for the job
#BSUB -M 2500                     # Set the per process enforceable memory limit to 2500MB.
#BSUB -o stdout.%J                # Send stdout to "stdout.[jobID]"
#BSUB -e stderr.%J                # Send stderr to "stderr.[jobID]"

# OPTIONAL JOB SPECIFICATIONS
#BSUB -P 123456                  # Set billing account to 123456
#BSUB -u email_address           # Send all emails to email_address
#BSUB -B -N                      # Send email on job begin (-B) and end (-N)

# load required module(s)
module load Python/3.5.2-intel-2017A

# run your program
./my_program.py
```

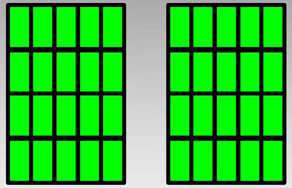
SUs = ?

Ada Job File (multi core, single node)

```
# NECESSARY JOB SPECIFICATIONS
#BSUB -L /bin/bash                # Use bashto initialize the job's execution environment.
#BSUB -J ExampleJob2              # Set the job name to "ExampleJob2"
#BSUB -W 24:00                    # Set the wall clock limit to 24hr and 0min
#BSUB -n 20                       # Request 20 cores total for the job
#BSUB -R "span[ptile=20]"         # Request 20 cores per node.
#BSUB -R "rusage[mem=2700]"       # Request 2700MB per process (CPU) for the job
#BSUB -M 2700                     # Set the per process enforceable memory limit to 2700MB.
#BSUB -o stdout.%J                # Send stdout to "stdout.[jobID]"
#BSUB -e stderr.%J                # Send stderr to "stderr.[jobID]"

# OPTIONAL JOB SPECIFICATIONS
#BSUB -P 123456              # Set billing account to 123456  #find your account with "myproject"
#BSUB -u email_address            # Send all emails to email_address
#BSUB -B -N                       # Send email on job begin (-B) and end (-N)

# load required module(s)
module load intel/2017A

# run your program
./my_multicore_program
```

SUs = ?

Ada Job File (multi core, multi node)



```
# NECESSARY JOB SPECIFICATIONS
#BSUB -J ExampleJob3               # Set the job name to "ExampleJob3"
#BSUB -L /bin/bash                 # Use bash to initialize the job's execution environment.
#BSUB -W 24:00                     # Set the wall clock limit to 24hr
#BSUB -n 40                        # Request 40 cores total for the job
#BSUB -R "span[ptile=20]"          # Request 20 cores per node.
#BSUB -R "rusage[mem=2500]"        # Request 2500MB per process (CPU) for the job
#BSUB -M 2500                      # Set the per process enforceable memory limit to 2500MB.
#BSUB -o stdout.%J                 # Send stdout to "stdout.[jobID]"
#BSUB -e stderr.%J                 # Send stderr to "stderr.[jobID]"

# OPTIONAL JOB SPECIFICATIONS
#BSUB -P 123456                  # Set billing account to 123456 #find your account with " myproject"
#BSUB -u email_address           # Send all emails to email_address
#BSUB -B -N                      # Send email on job begin (-B) and end (-N)

# load required module(s)
module load intel/2017A

# run your program
./my_multicore_multinode_program
```

SUs = ?

Job Submission and Tracking

LSF command	Description
bsub < *jobfile.sh*	Submit jobfile.sh to batch system
bjobs -u *user_name* **bjobs -l** *job_id*	List jobs for username List job details including Max Memory used
bkill *job_id*	Kill a job
bhist -l *job_id*	Show information for a job (can be when job is running or recently finished)
lnu -l -j *job_id*	Show resource usage for a job

https://hprc.tamu.edu/wiki/HPRC:Batch_Translation

Submitting Your Job and Check Job Status

Submit job

```
bsub < run_trimmomatic.sh
```

```
Verifying job submission parameters...
Verifying project account...
     Account to charge:   082792010838
         Balance (SUs):      4871.5983
         SUs to charge:         24.000
Job <2470599> is submitted to default queue <sn_short>.
```

Check status

(dash lower case l as in *list*)

more detailed; shows Max Memory:

```
bjobs -l 2470599
```

```
bjobs
```

```
JOBID    STAT USER        QUEUE    JOB_NAME  NEXEC_HOST SLOTS RUN_TIME       TIME_LEFT
2470599  RUN  mynetid     sn_short trimmo    1          1     0 second(s)    23:59 L
```

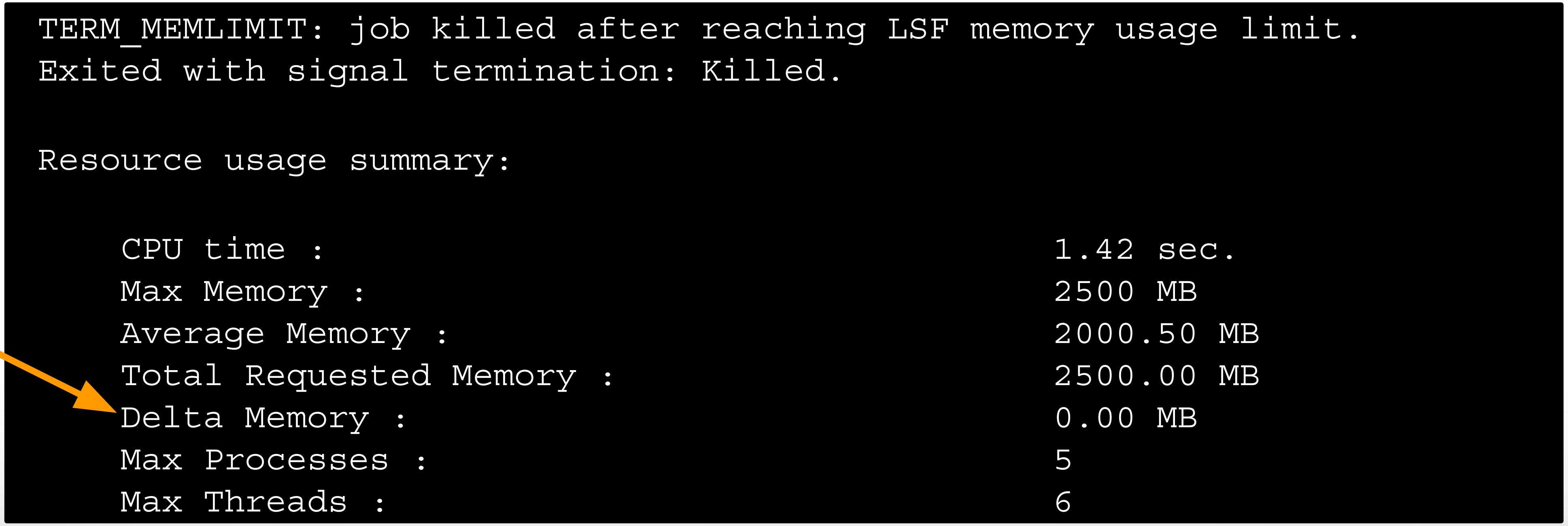
Debug job failures

- Debug job failures using the stdout and stderr files

The stdout file was created by this parameter in your job script file. %J = jobid

```
#BSUB -o stdout.%J
```

Job ran out of requested memory



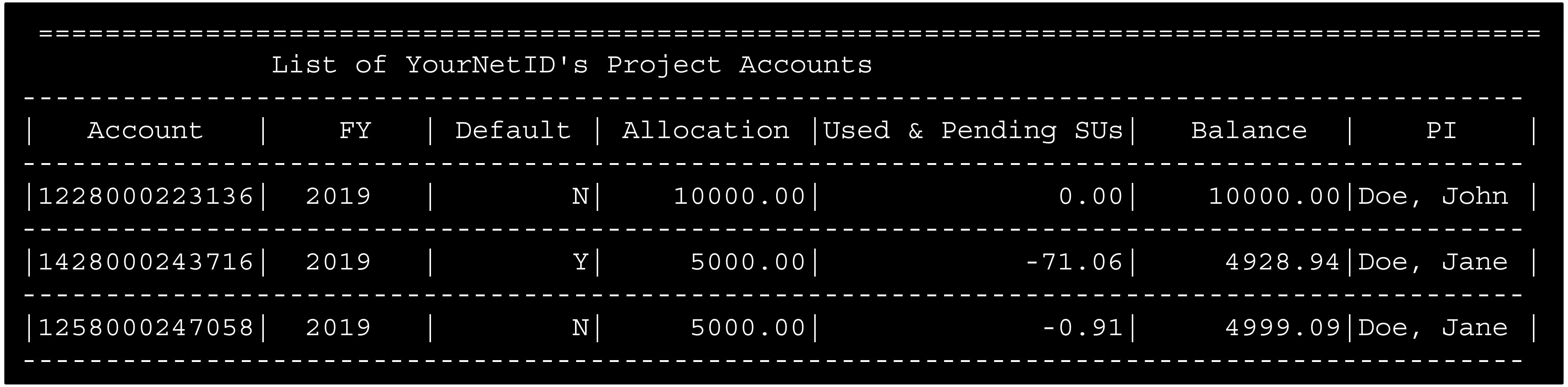
```
TERM_MEMLIMIT: job killed after reaching LSF memory usage limit.
Exited with signal termination: Killed.

Resource usage summary:

    CPU time :                                      1.42 sec.
    Max Memory :                                    2500 MB
    Average Memory :                                2000.50 MB
    Total Requested Memory :                        2500.00 MB
    Delta Memory :                                  0.00 MB
    Max Processes :                                 5
    Max Threads :                                   6
```

Make the necessary adjustments to BSUB parameters in your job script and resubmit the job

Check your Service Unit (SU) Balance

- List the SU Balance of your Account(s)

```
myproject
```



```
 =====================================================================================
               List of YourNetID's Project Accounts
-------------------------------------------------------------------------------------
|   Account    |    FY     | Default | Allocation |Used & Pending SUs|   Balance   |     PI    |
-------------------------------------------------------------------------------------
|1228000223136|  2019     |        N|    10000.00|              0.00|     10000.00|Doe, John  |
-------------------------------------------------------------------------------------
|1428000243716|  2019     |        Y|     5000.00|            -71.06|      4928.94|Doe, Jane  |
-------------------------------------------------------------------------------------
|1258000247058|  2019     |        N|     5000.00|             -0.91|      4999.09|Doe, Jane  |
-------------------------------------------------------------------------------------
```

- To specify a project ID to charge in the job file
  - **#BSUB -P Account#**
- Run "**myproject -d Account#**" to change default project account
- Run "**myproject -h**" to see more options

https://hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit
https://hprc.tamu.edu/wiki/HPRC:AMS:UI

Job submission issue: insufficient SUs

```
$ bsub < myjob
Verifying job submission parameters...
Verifying project account...
     Account to charge:   082792010838
         Balance (SUs):       342.5322
         SUs to charge:       480.0000
         --------------------------------------------------------------------------------
        |ERROR! Your project account does not have sufficient balance to submit your job! |
         --------------------------------------------------------------------------------
Request aborted by esub. Job not submitted.
```

- What to do if you need more SUs
  - Wait for your currently running jobs to finish to get reimbursed for unused runtime
  - Ask your PI to transfer SUs to your account

https://hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit
https://hprc.tamu.edu/wiki/HPRC:AMS:UI

Job Environment Variables

- **Ada:**
  - **$LSB_JOBID** = job id
  - **$SCRATCH** = /scratch/user/NetID
  - **$TMPDIR** = /work/$LSB_JOBID.tmpdir
    - $TMPDIR is local to each assigned compute node for the job and is about 750 GB
    - Use of $TMPDIR is recommended for jobs that use many small temporary files
    - Do not use $TMPDIR for software that has checkpoints to restart where it left off

https://hprc.tamu.edu/wiki/Ada:Batch_Processing_LSF#Environment_Variables
https://hprc.tamu.edu/wiki/Terra:Batch#Environment_Variables

Common Job Problems

- Control characters (`^M`) in job files or data files edited with Windows editor
  - remove the `^M` characters with: `dos2unix my_job_file`
- Did not load the required module(s)
- Insufficient walltime specified in `#BSUB -W`
- Insufficient memory specified in `#BSUB -M` and `-R "rusage[mem=xxx]"`
- Running OpenMP jobs across multiple compute nodes
- Insufficient SU: See your SU balance: `myproject`
- Insufficient disk or file quotas: check quota with `showquota`
- Using GUI-based software without setting up X11 forwarding
  - Enable X11 forwarding at login `ssh -X netid@ada.tamu.edu`
  - Or use VNC app in HPRC portal

https://hprc.tamu.edu/wiki/HPRC:CommonProblems

CRLF Line Terminators

Windows editors such as Notepad will add hidden Carriage Return Line Feed (CRLF) characters that will cause problems with many applications

```
file dos_text.txt
```

# use `file` command to check

```
dos_text.txt: ASCII English text, with CRLF line terminators
```

```
cat -v dos_text.txt
```

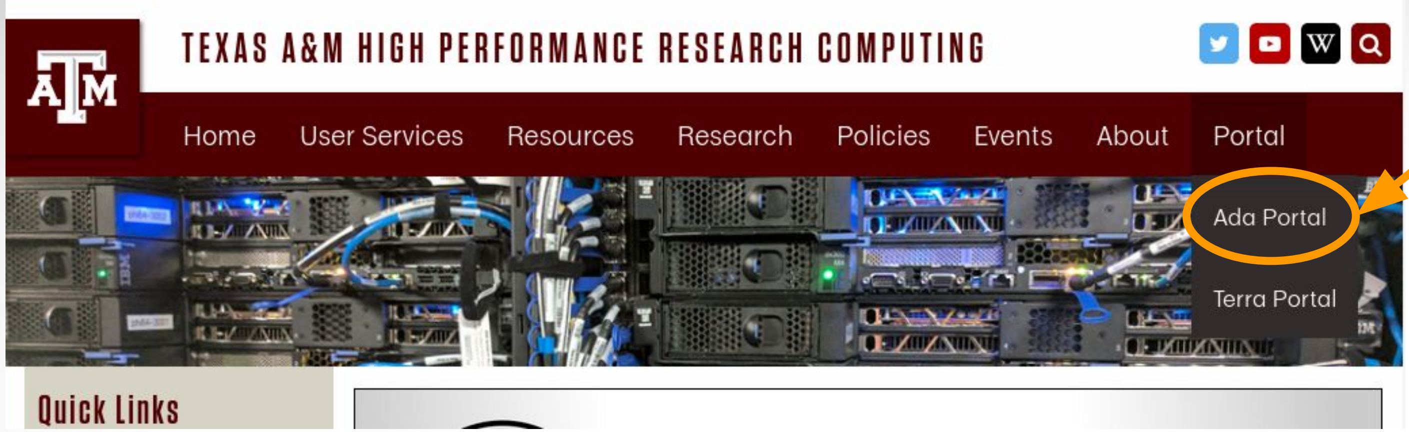
# use `cat` command to see CRLF characters

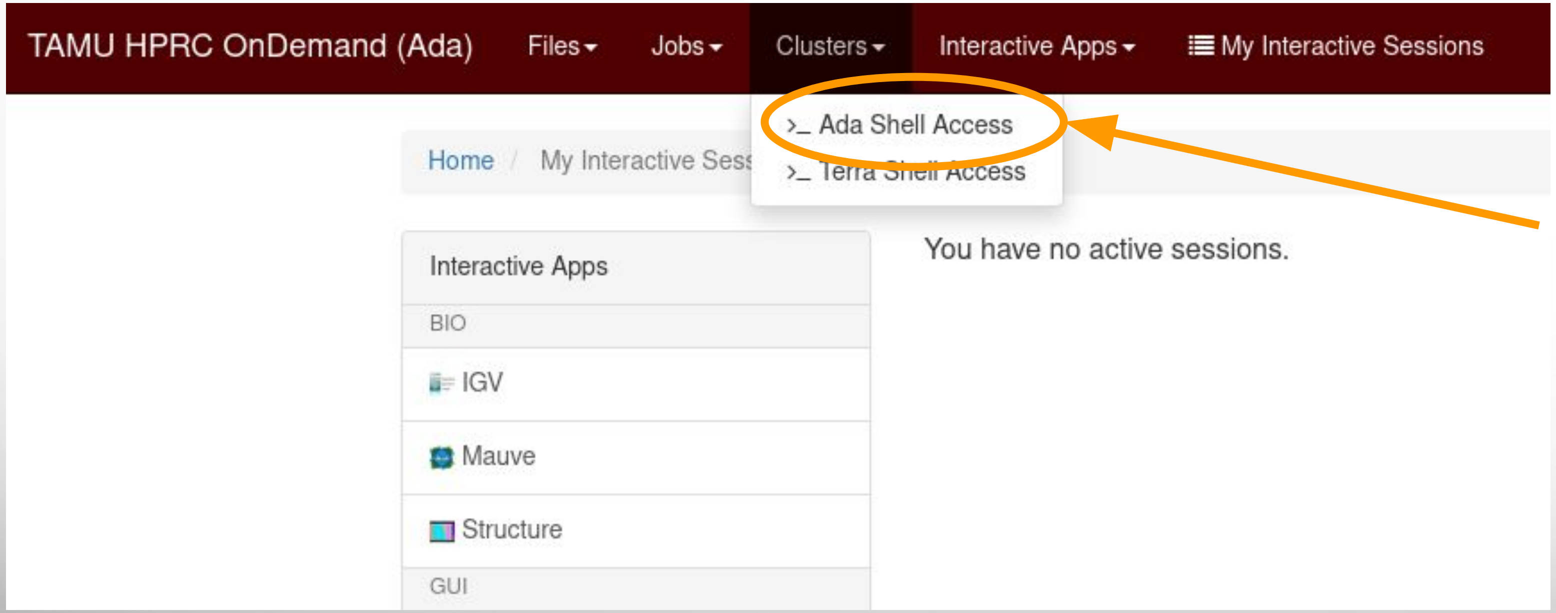
```
dos2unix dos_text.txt
file dos_text.txt
```

# use `dos2unix` command to correct

```
dos_text.txt: ASCII English text
```

HPRC Portal





You have no active sessions.

The HPRC portal allows users to do the following

- Browse files on the Ada (Curie) filesystem
- Access the ada, terra or curie UNIX command line
  - no SUs charged
- Launch jobs
  - SUs charged
- Compose job scripts
- Launch interactive GUI apps (SUs charged)
  - Ada
    - IGV
    - MATLAB
    - VNC
    - Galaxy
    - Jupyter Notebook
    - RStudio
  - Terra
- Monitor and stop running jobs and interactive sessions

HPRC Cluster Usage Notes

- Ada and Terra filesystems are not the same so you have to copy files from Ada to Terra to use Terra
- Ada uses LSF (**`#BUSB`**) scheduler and Terra uses Slurm (**`#SBATCH`**) scheduler.
    - The new cluster (fall 2020) will use the Slurm scheduler
- SUs do not transfer across clusters.
- Not all software on Ada is also available on Terra.
    - Request installation if you need a software package installed or version updated.
    - Some software is installed in a Conda environment and not as a specific module

```
module load Anaconda/3-5.0.0.1
conda env list
```

- Can test software on the login nodes but do not use more than 8 cores
    - processes get automatically killed on login node when they reach 60 minutes
    - login session gets automatically disconnected when idle for 60 minutes
- Transfer large data between Ada and Terra using the login nodes
    - transfer large data between HPRC clusters and sites off campus using the fast transfer nodes
- Carefully read the stdout and stderr files for completed and failed jobs
    - most causes of failed jobs can be found in the log files
- Search the wiki and look at the FAQ pages for a solution before sending a helpdesk request
    - https://hprc.tamu.edu/wiki/HPRC:CommonProblems
    - https://hprc.tamu.edu/wiki/Bioinformatics:FAQ

Need Help?

- First check the FAQ https://hprc.tamu.edu/wiki/HPRC:CommonProblems
  - Ada User Guide https://hprc.tamu.edu/wiki/Ada
    - Exercises https://hprc.tamu.edu/wiki/Ada:Exercises
  - Terra User Guide https://hprc.tamu.edu/wiki/Terra
    - Exercises https://hprc.tamu.edu/wiki/Terra:Exercises
- Email your questions to **help@hprc.tamu.edu**. (Managed by a ticketing system)
- Help us, help you -- we need more info
  - Which Cluster
  - UserID/NetID (*UIN is not needed!*)
  - Job id(s) if any
  - Location of your jobfile, input/output files
  - Application used if any
  - Module(s) loaded if any
  - Error messages
  - Steps you have taken, so we can reproduce the problem
- Or visit us @ 114A Henderson Hall
  - Making an appointment is recommended.

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

Thank you.

Any question?