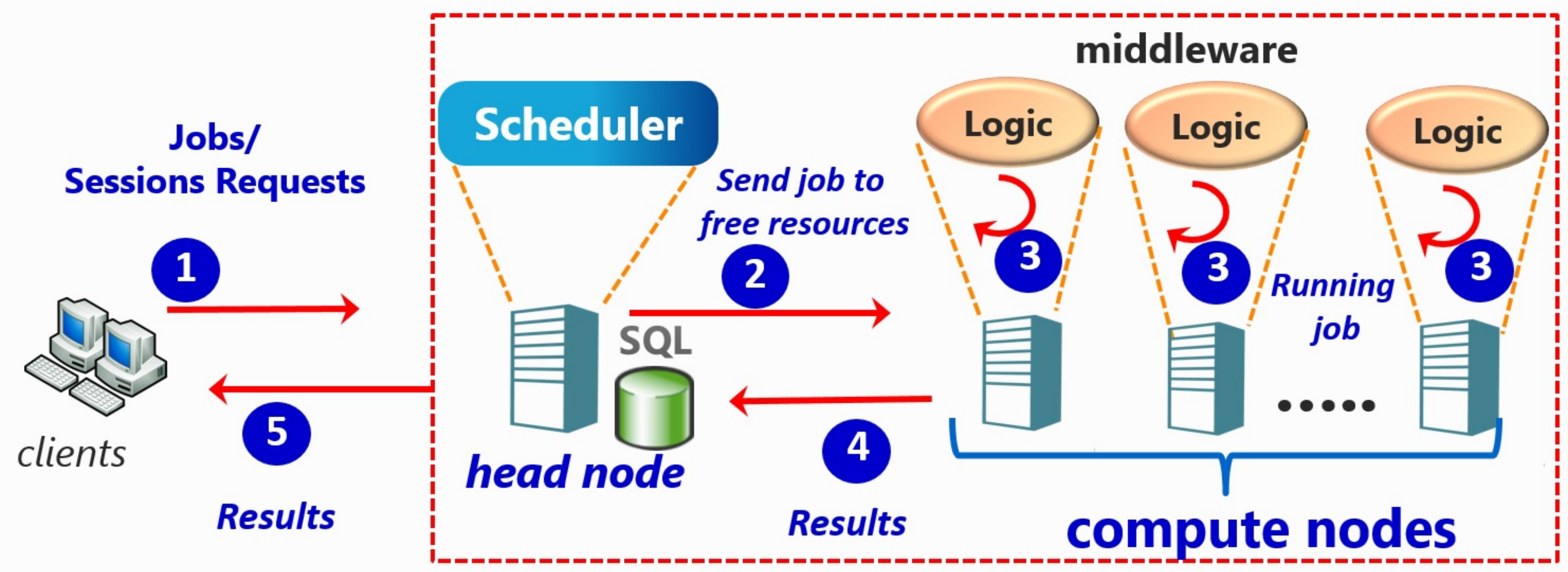
2017 HPC @UofA

Introduction to Phoenix

Objectives:

- Understanding Phoenix
- Submission of the first job
- Account management

Phoenix Components



Phoenix Specifications

5120 cores in 160 nodes – 32 cores each

125 GB each

144 K80 GPUs

High speed internal network

Head vs Computing Node

- Head node
  - Login on
  - File transfer
  - Compiling
  - Prepare environment
  - Quick executions
  - Job submission
  - Simple pre-post processing (ideally to be done on your local workstation)

- Computing node
  - No direct access to
  - Execute jobs
  - Highest computing power

The scheduler
(slurm)

Control available resources
Assign priority according fair-share system
Control job execution

Be clever with your job requirements
99% of longer waiting times are due to
bad requests

- 10 hours if you program needs only 1
- 32 GB if you program works with 1GB
- Many nodes/cores if your code is NOT PARALLEL

Common Mistakes

SLURM Fairshare algorithm

- Each account has a numerical allocation (in SU)
- Scheduler compares relative usage vs allocation

  If usage < allocation, job priority is increased

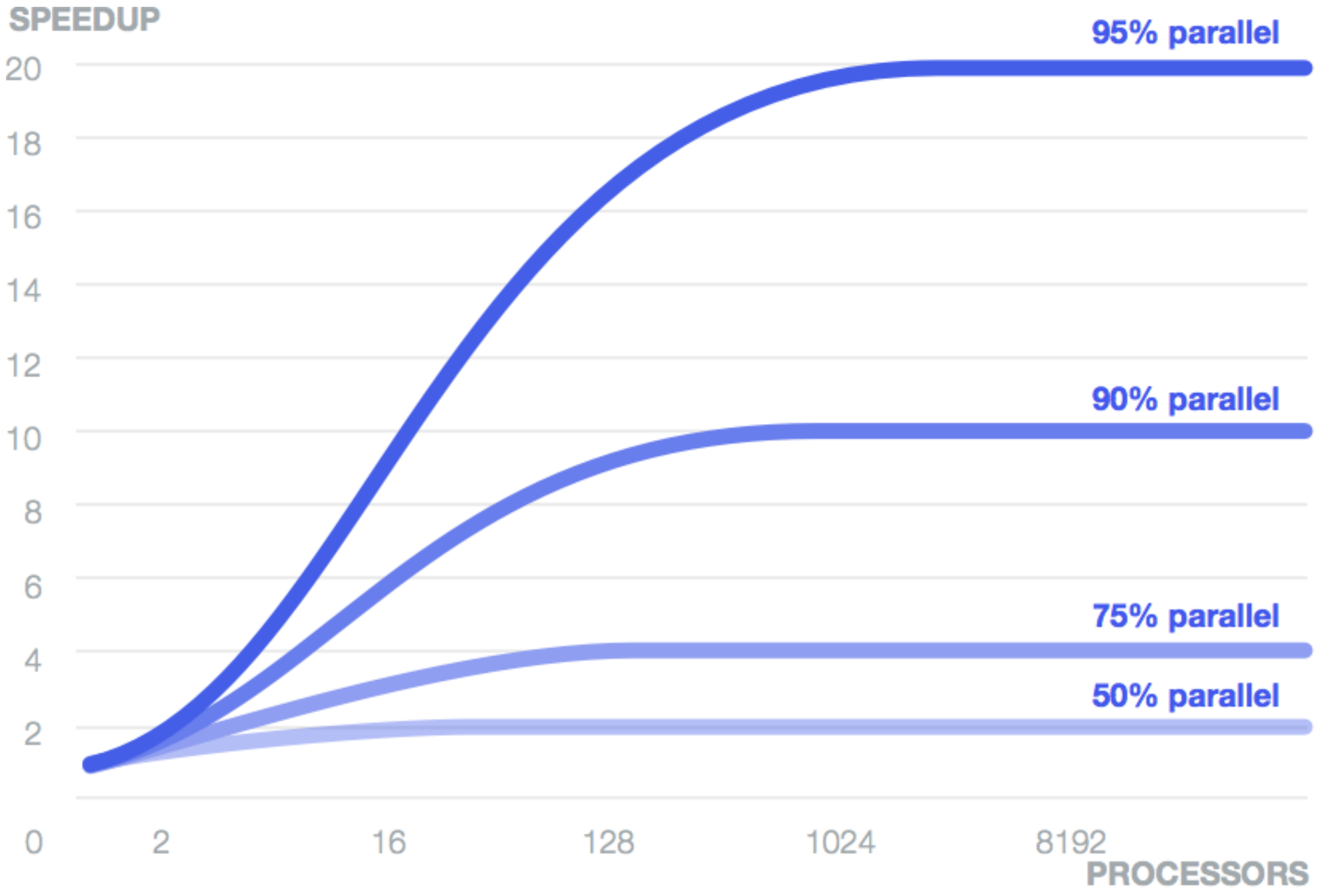
  If usage > allocation, job priority is decreased
- Higher priority jobs run first
- Other factors can also affect when a job starts

  Job only starts if all required resources are available

  E.g. wall-time, CPUs, GPUs, memory requested

  Important to match resource request to job requirement!

Speed Up


SPEEDUP
20
18
16
14
12
10
8
6
4
2
0
2
16
128
1024
8192
PROCESSORS
95% parallel
90% parallel
75% parallel
50% parallel

Storage:

- $HOME (/home/aXXXXXXX)
  - Account configurations
  - Important things (backup)
  - Small quota ~ 5-10GB

Storage:

- FASTDIR
  - /fast/users/aXXXXXXX
  - Inputs and outputs files
  - Executables
  - Jobs

  - Must be your working folder

Tutorials and Fun

What to expect from this Tut?

- Log on Phoenix from command line
- Transfer files to and from Phoenix
- Submit jobs
- Managing my account

Tutorial #1a – How to connect to Phoenix

SSH client to login

Copy software from: https://goo.gl/u4TW34

- Windows:
  - **Bitvise**
  - Putty

- Linux/OSx:
  - ssh

Head node:

phoenix.adelaide.edu.au

```
ssh aXXXXXXX@phoenix.adelaide.edu.au
```

Tutorial #1b – How to transfer files to Phoenix

SCP client for file transferring:

- Windows:
  - **Bitvise**
  - WinSCP
  - Filezilla
- Linux/OSx:
  - scp

Head node:

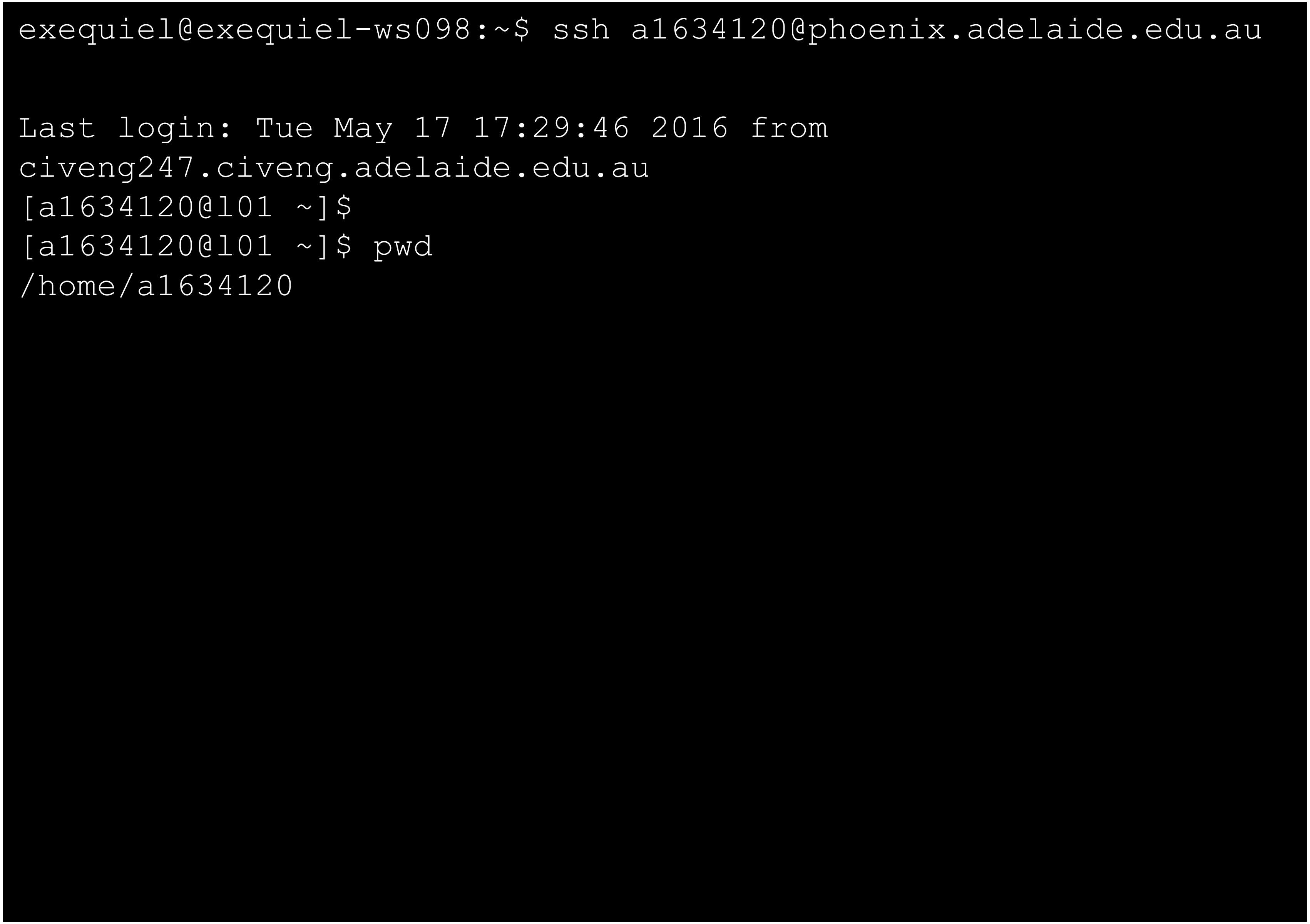
phoenix.adelaide.edu.au

From your computer to Phoenix:

```
scp FILES aXXXXXXX@phoenix.adelaide.edu.au:/fast/users/aXXXXXXX
```

From Phoenix to your computer:

```
scp aXXXXXXX@phoenix.adelaide.edu.au:/fast/users/aXXXXXXX .
```



```
exequiel@exequiel-ws098:~$ ssh a1634120@phoenix.adelaide.edu.au


Last login: Tue May 17 17:29:46 2016 from
civeng247.civeng.adelaide.edu.au
[a1634120@l01 ~]$
[a1634120@l01 ~]$ pwd
/home/a1634120
```

Tutorial #1c – linux commands

I want to	Linux command
List files	ls
Know where I am	pwd
Copy files	cp
Delete files	rm
Create a folder	mkdir
Delete a folder	rmdir
Move to a folder	cd
Display file content	cat; more; less
Edit a text file	vi; nano
Define a variable	VAR=VALUE
Export a variable	export VAR=VALUE
Check shell	echo $SHELL; echo $0

Tutorial #2 – Modules

- Modules system features:
  - Easy way to use installed software
  - Automatic dependencies
  - You could install your own software as personal modules!

Modules commands

Command	Description
avail	List modules
load	Load modules
unload	Unload modules
swap	Changing versions
purge	Fresh start (unload all)
list	List of loaded modules
show	Show detail of a module

Modules commands

Command	Example
avail	module avail matlab
load	module load Python/2.7.11-foss-2016uofa
unload	module unload R/3.3.0-foss-2016uofa
swap	module swap matlab/2016b matlab/2016a
purge	module purge
list	module list
show	module show matlab/2016b

Tutorial #2a – Changing shell

```
$ echo $SHELL

$ module load uofa_util

$ phoenix-shell -s /bin/bash
```

Software on Phoenix

Compilers:

- C/C++ (GNU and Intel)
- Fortran (GNU, Intel and PGI)
- C# (mono)
- Python
- Matlab
- Java
- R
- Perl

Modules for Bioinformatics

- Blast
- Bowtie
- CENSOR
- Cufflinks
- MUSCLE
- QIIME
- Tophat
- Velvet

Modules for Engineering

- Abaqus (Finite element analysis)
- Ansys (Engineering simulation)
- CST (3D electromagnetic EDA solutions)
- Ls-dyna (Finite element)
- Gurobi (optimisation)
- Theano (Fast computing)
- Comsol (Multiphysics)
- GSLIB (Geostatistics)

Toolchains

Set of compiler + libraries:

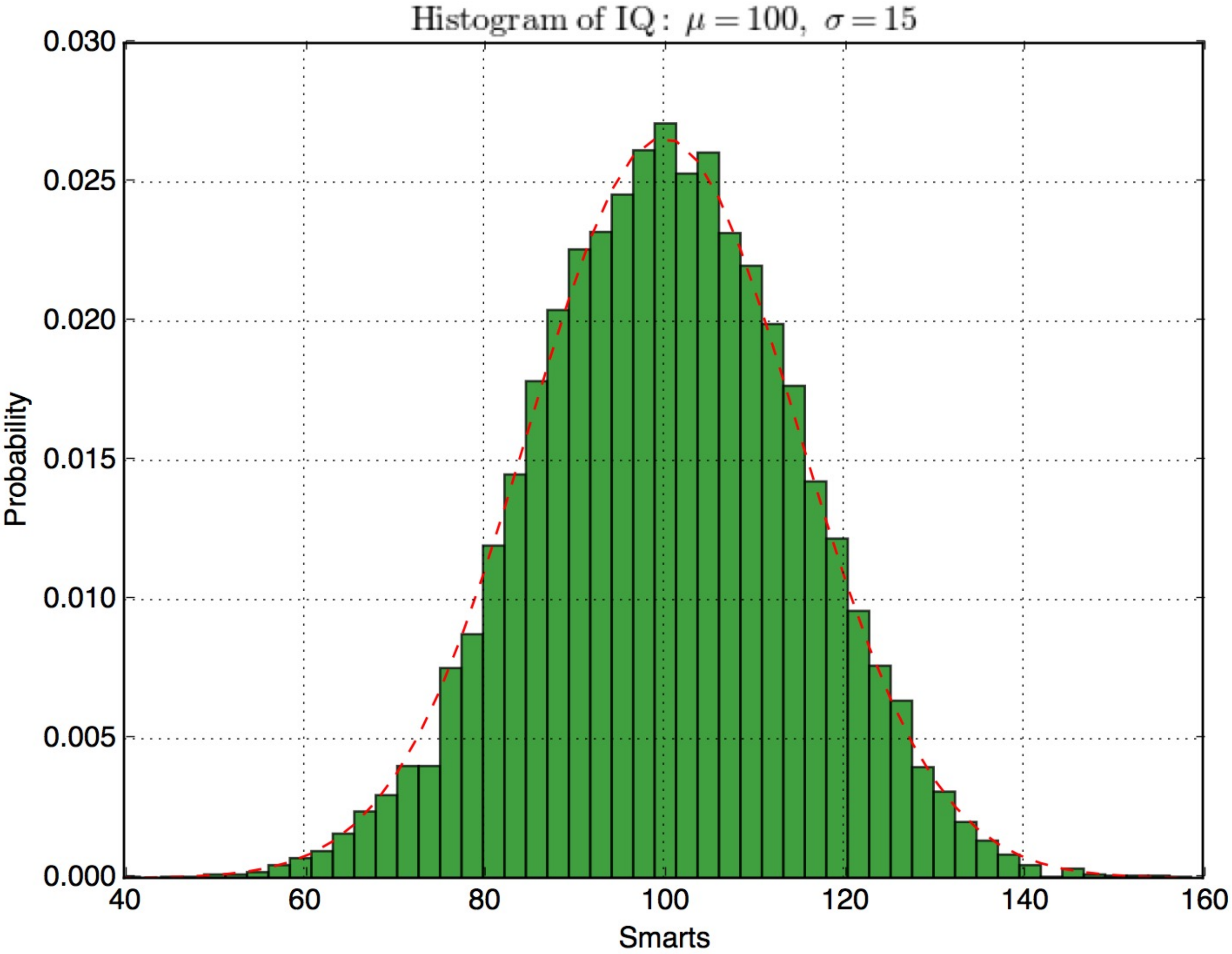
- FOSS:
  - GNU compiler
  - FFTW (Fast Fourier transform)
  - Open BLAS (BLAS + LAPACK)
  - ScaLAPACK (HPC and distributed LAPACK)
  - OpenMPI (distributed)

Toolchains

Set of compiler + libraries:

- Intel:
  - Intel compilers (C/C++/Fortran)
  - MKL (Mathematical libraries)
  - OpenMPI (distributed computing)

Tutorial #3 – Python program



Tutorial #3 – Steps

1. Make a folder in your $HOME
2. Copy from /apps/examples/python the python program “histo.py”
3. Find and load Python/2.7.11
4. Execute the script
5. Copy the histogram to your PC from Phoenix and compare

Tutorial #3 – Preparing for Job submission

1. Find all modules you need
2. Setup inputs and outputs folders
3. Define some environmental variables
4. A quick execution on head node
5. If all is ok, write a script with steps 1-4
6. Logoff-logon and re-run script (repeatability)

Borrow and adapt
a simple script from our wiki:

https://wiki.adelaide.edu.au/hpc

Or just copy it from
/apps/examples/python

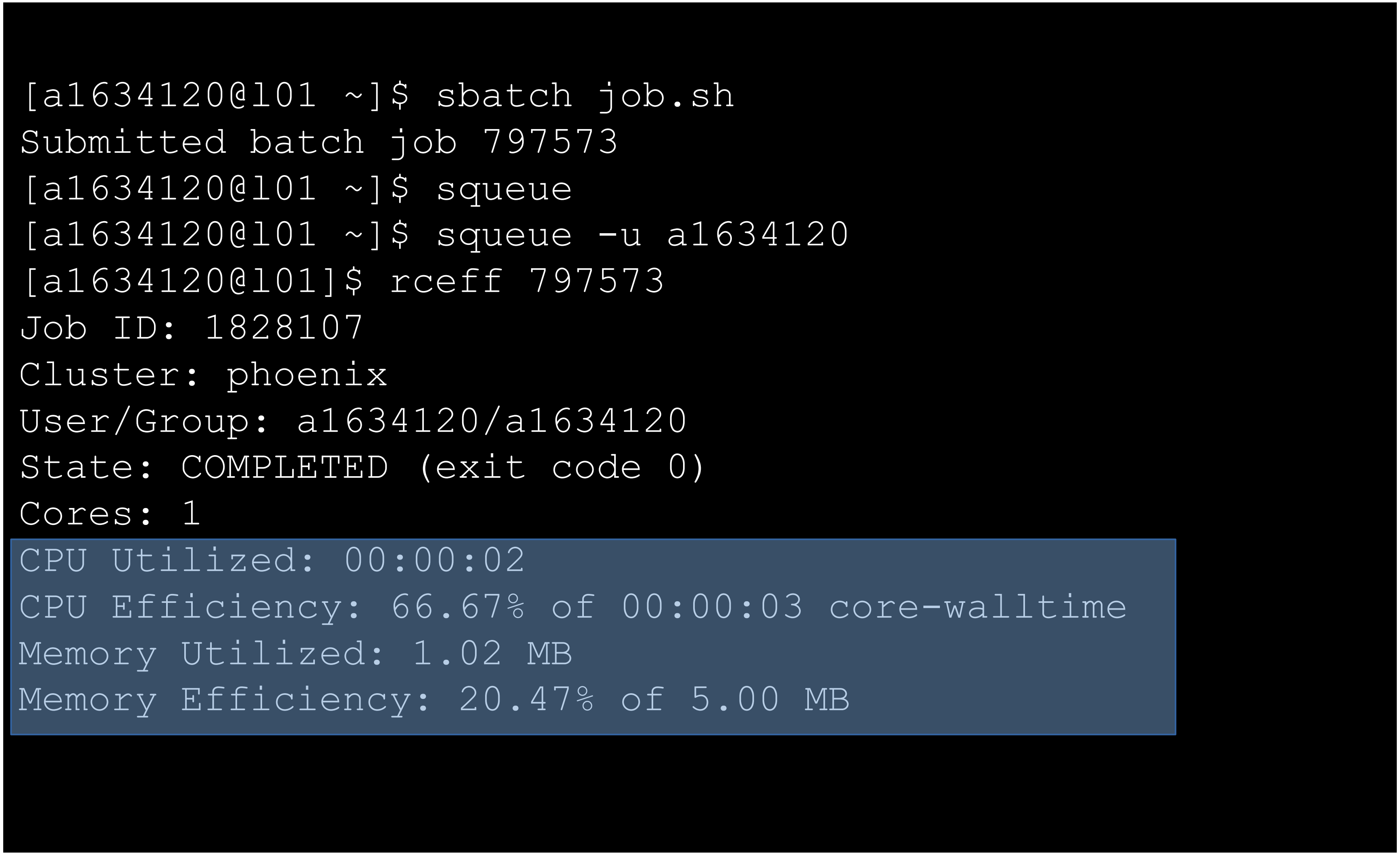
Tutorial #4 – Job submission and monitoring

```
#!/bin/bash
#SBATCH -p batch
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --time=00:01:00
#SBATCH --mem=5MB


# Notification configuration
#SBATCH --mail-type=ALL
#SBATCH --mail-user=firstname.lastname@adelaide.edu.au


module load Python/2.7.11-foss-2016uofa
python histo.py
```

Tutorial #4 – Job submission

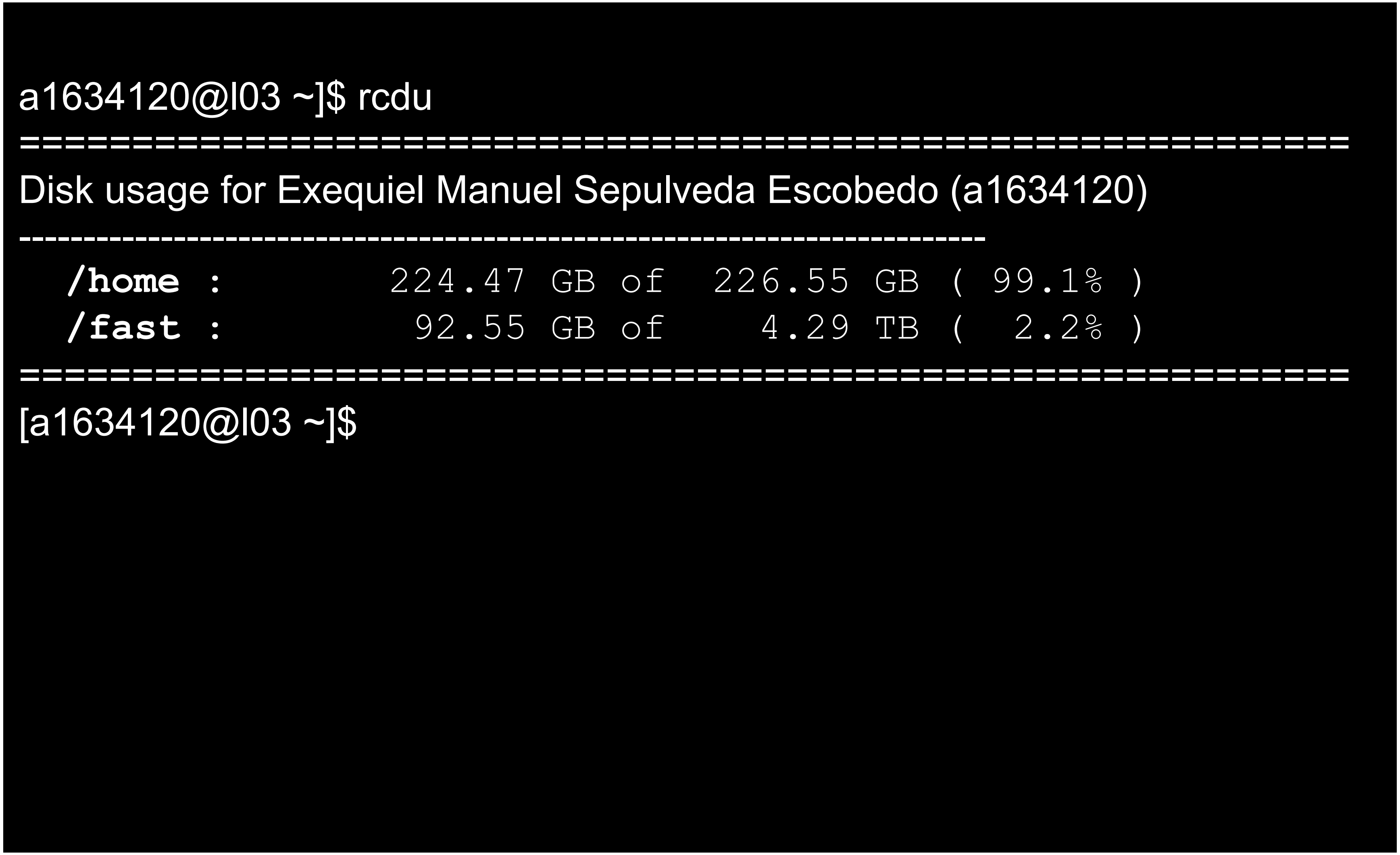


```
[a1634120@l01 ~]$ sbatch job.sh
Submitted batch job 797573
[a1634120@l01 ~]$ squeue
[a1634120@l01 ~]$ squeue -u a1634120
[a1634120@l01]$ rceff 797573
Job ID: 1828107
Cluster: phoenix
User/Group: a1634120/a1634120
State: COMPLETED (exit code 0)
Cores: 1
CPU Utilized: 00:00:02
CPU Efficiency: 66.67% of 00:00:03 core-walltime
Memory Utilized: 1.02 MB
Memory Efficiency: 20.47% of 5.00 MB
```

Tutorial #5 – Job monitoring

I want to	Linux command
Submit	sbatch
Cancel a job	scancel JOB_ID
Get job info	sacct -j JOB_ID
Get my jobs in the queue	squeue –u axxxxxxx
Get job info	scontrol show job JOB_ID
Get efficiency info	rceff JOB_ID
Get share info	sshare –u axxxxxxx

Tutorial #6 – disk usage



```
a1634120@l03 ~]$ rcdu
==========================================================
Disk usage for Exequiel Manuel Sepulveda Escobedo (a1634120)
---------------------------------------------------------------------------------
  /home :        224.47 GB of  226.55 GB ( 99.1% )
  /fast :         92.55 GB of    4.29 TB (  2.2% )
==========================================================
[a1634120@l03 ~]$
```

https://wiki.adelaide.edu.au/hpc

hpcsupport@adelaide.edu.au

Walk-up sessions:

2nd and 4th Tuesday

10:00 – 12:00

Learning Innovation Studio