

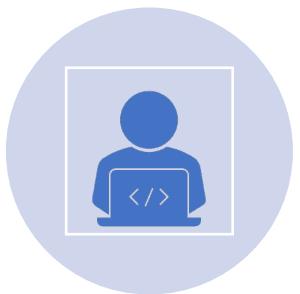
Introduction to supercomputing

By Kevin Fotso

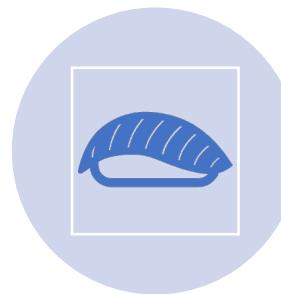
Recap on the Shell



What is a shell?

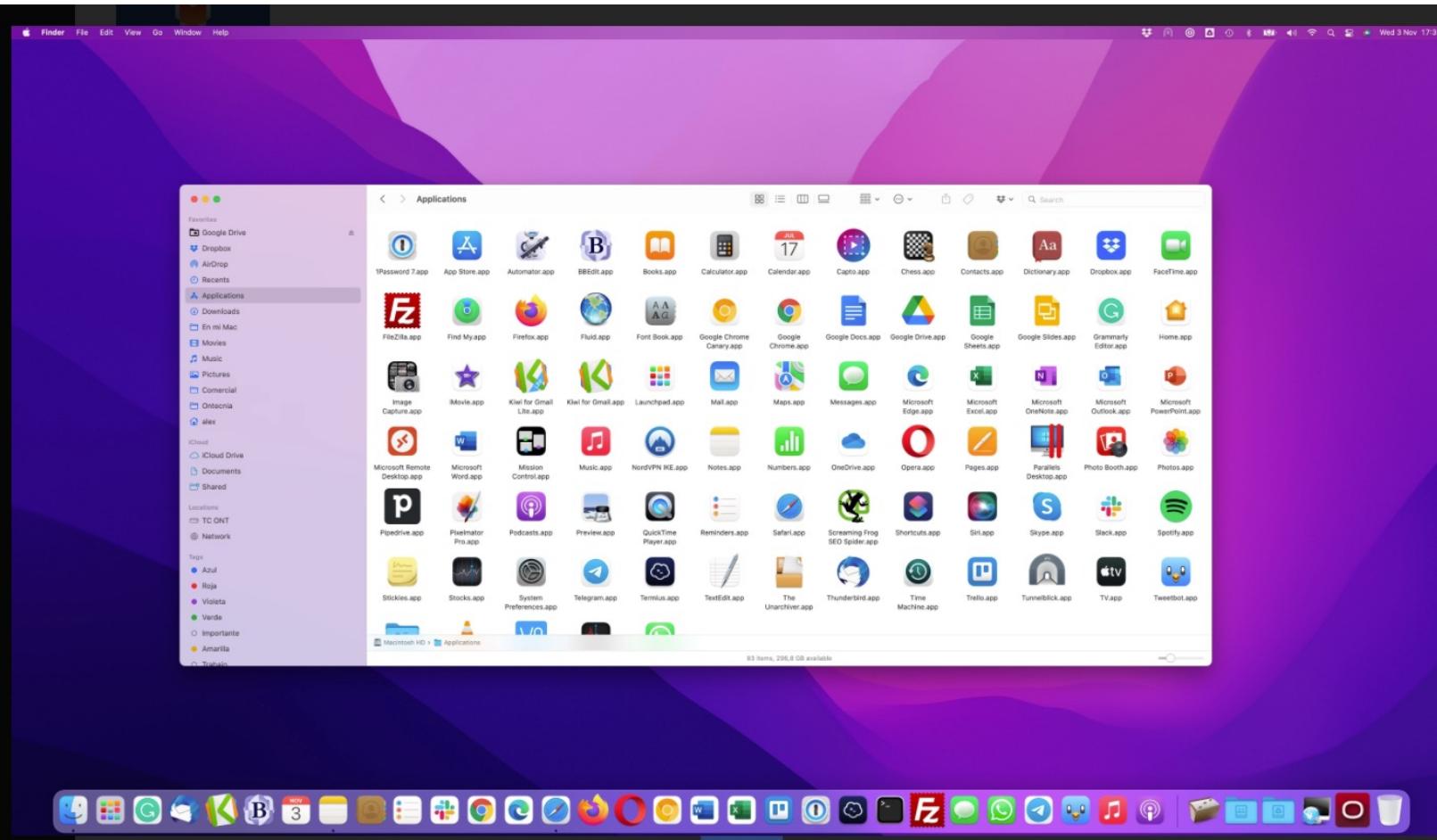


The shell is a program that takes input commands (from the mouse or keyboard) and translates them to the OS to perform.



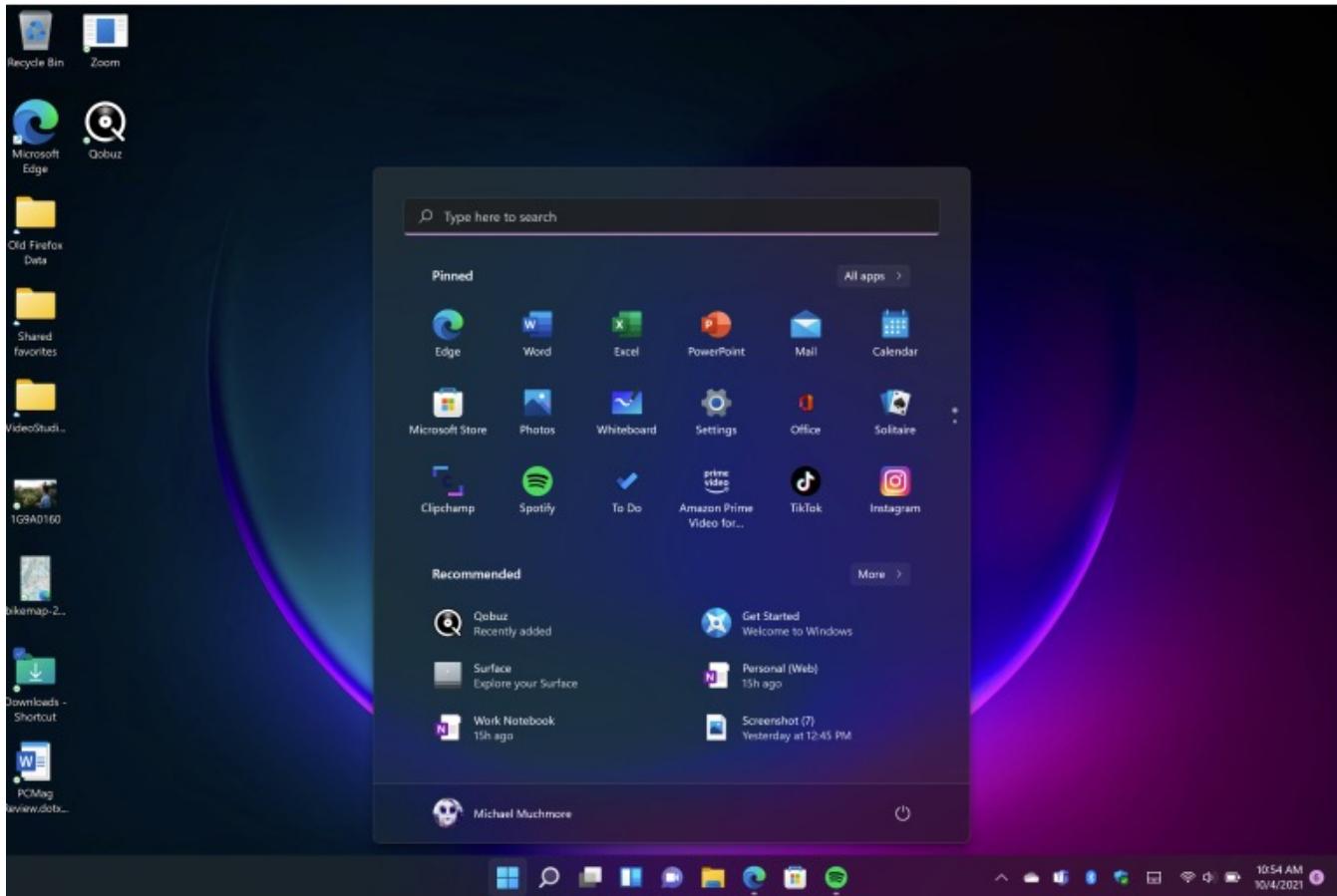
There are graphical shells and linux shells.

Graphical shells (1)



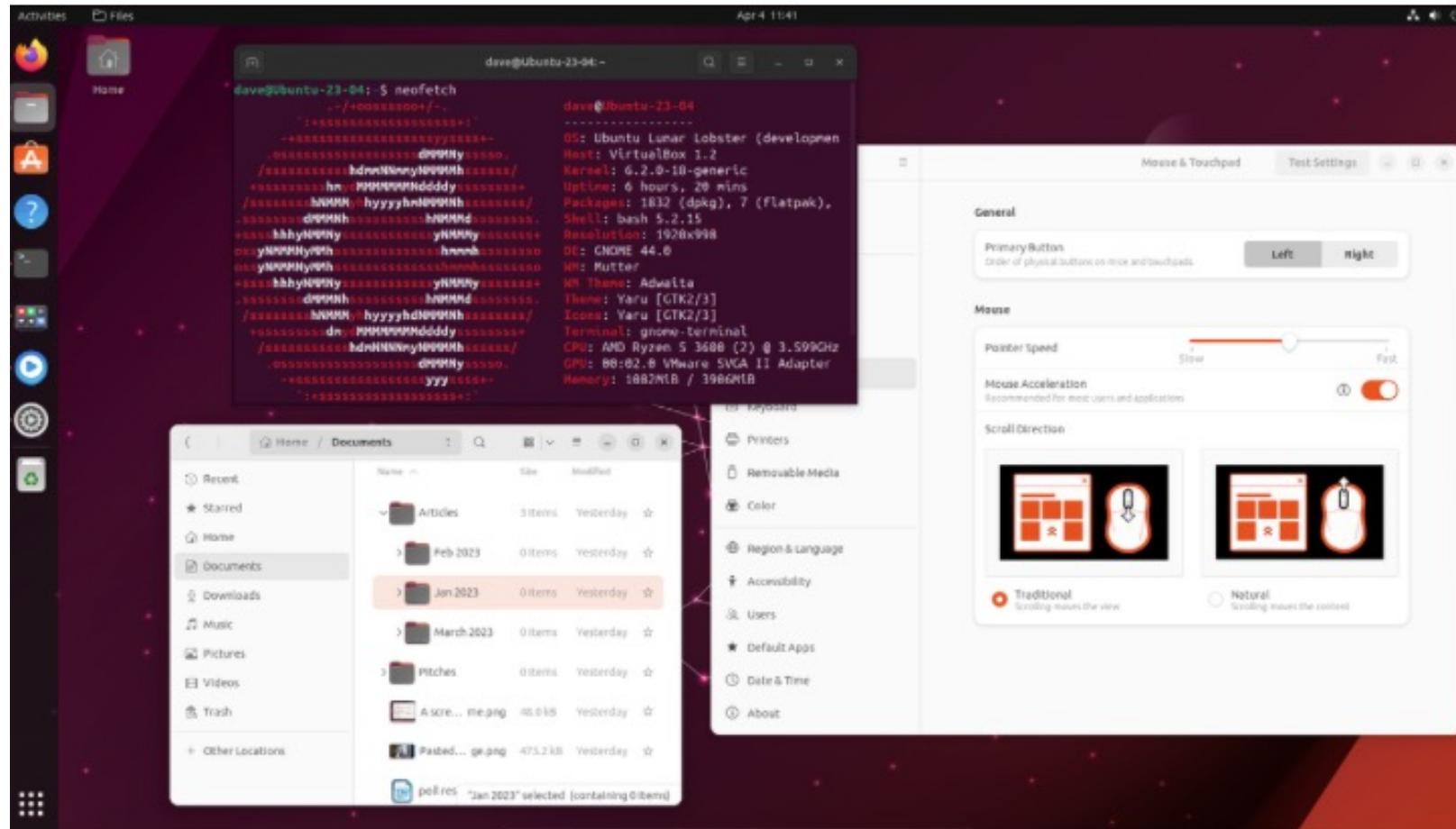
Macbook OS

Graphical shells (2)



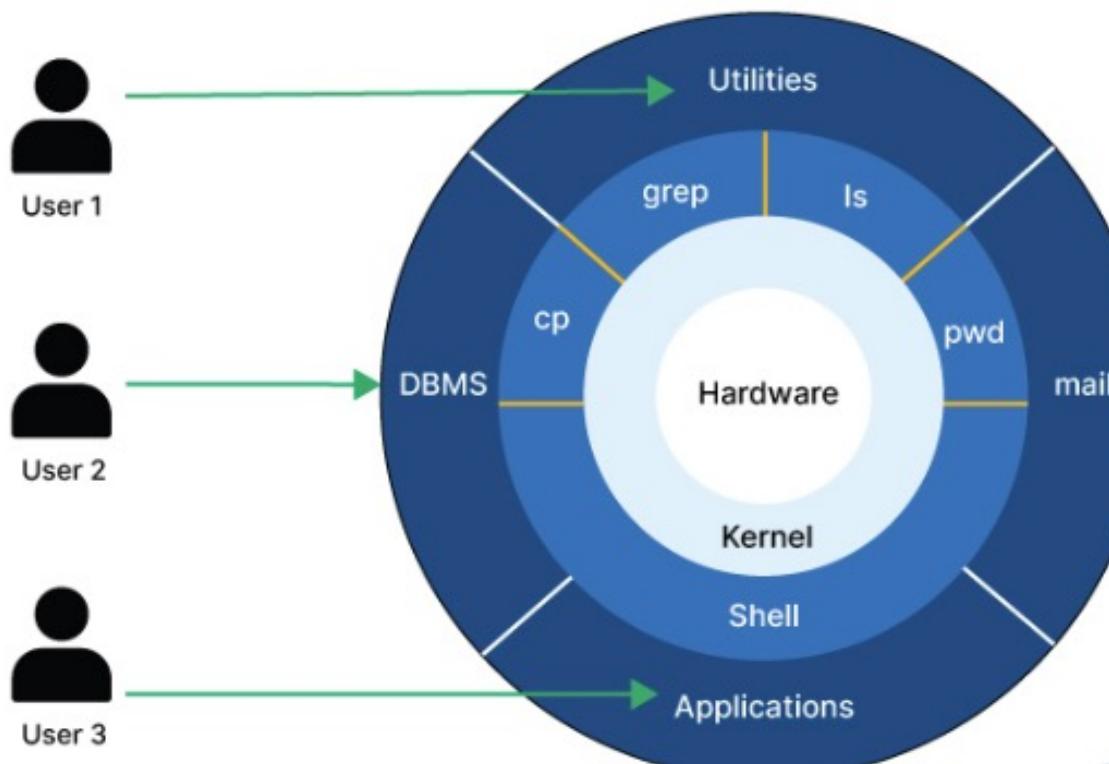
Windows

Graphical shells (3)



Ubuntu (Linux)

Linux shell



- The kernel manages resource access to the hardware.
- The OS is made of the Kernel + the useful programs that interact with it to get access to resources
- The shell is where the user interacts with the OS

BASH shell

- Stands for Bourne Again Shell.
- Created in 1989 by Brian Fox

Overview of the shell prompt (1)

Welcome to University of Colorado Boulder Research Computing!

Full documentation is available in our user guide at <https://www.rc.colorado.edu/support/user-guide>. If you have a question that's not answered there, contact us at rc-help@colorado.edu.

A number of directories have been created for you already:

- * `/home/\$USER`, your home directory
- * `/projects/\$USER`, your project directory

Run the command `module avail` to see a list of available software.

To prevent this README from being displayed at login, edit your `~/.bash_profile` or `~/.login` files.

Welcome to CU-Boulder Research Computing.

- * Website <http://colorado.edu/rc>
- * Questions? rc-help@colorado.edu
- * Subscribe to system announcements: <https://curc.statuspage.io/>
- * Please type `rc-help` for the Acceptable Use Policy and a short help page.

You are using login node: login-ci1

Users who had jobs in the queue prior to the planned maintenance should check to confirm these jobs are still queued. Some jobs, particularly those scheduled since midnight today (Wed June 7), may have been canceled during the maintenance period.

[kfotso@xsede.org@login-ci1 ~]\$ █

Overview the shell prompt (2)

```
Welcome to University of Colorado Boulder Research Computing!
```

```
Full documentation is available in our user guide at  
https://www.rc.colorado.edu/support/user-guide. If you have a question  
that's not answered there, contact us at rc-help@colorado.edu.
```

```
A number of directories have been created for you already:
```

- * `/home/\$USER`, your home directory
- * `/projects/\$USER`, your project directory

```
Run the command `module avail` to see a list of available software.
```

```
To prevent this README from being displayed at login, edit your  
.bash_profile or .login files.
```

```
Welcome to CU-Boulder Research Computing.
```

- * Website <http://colorado.edu/rc>
- * Questions? rc-help@colorado.edu
- * Subscribe to system announcements: <https://curc.statuspage.io/>
- * Please type rc-help for the Acceptable Use Policy and a short help page.

```
You are using login node: login-ci1
```

```
Users who had jobs in the queue prior to the planned maintenance should check  
to confirm these jobs are still queued. Some jobs, particularly those scheduled  
since midnight today (Wed June 7), may have been canceled during the  
maintenance period.
```



CURC messaging

Overview the shell prompt (3)

```
[kfotso@xsede.org@login-ci1 ~]$
```



Username



Hostname

Overview the shell prompt (4)

```
[kfotso@xsede.org@login-ci1 ~]$ █
```



- ~ shows your current working directory.
- ~ (tilde) stands for your home directory in the filesystem tree

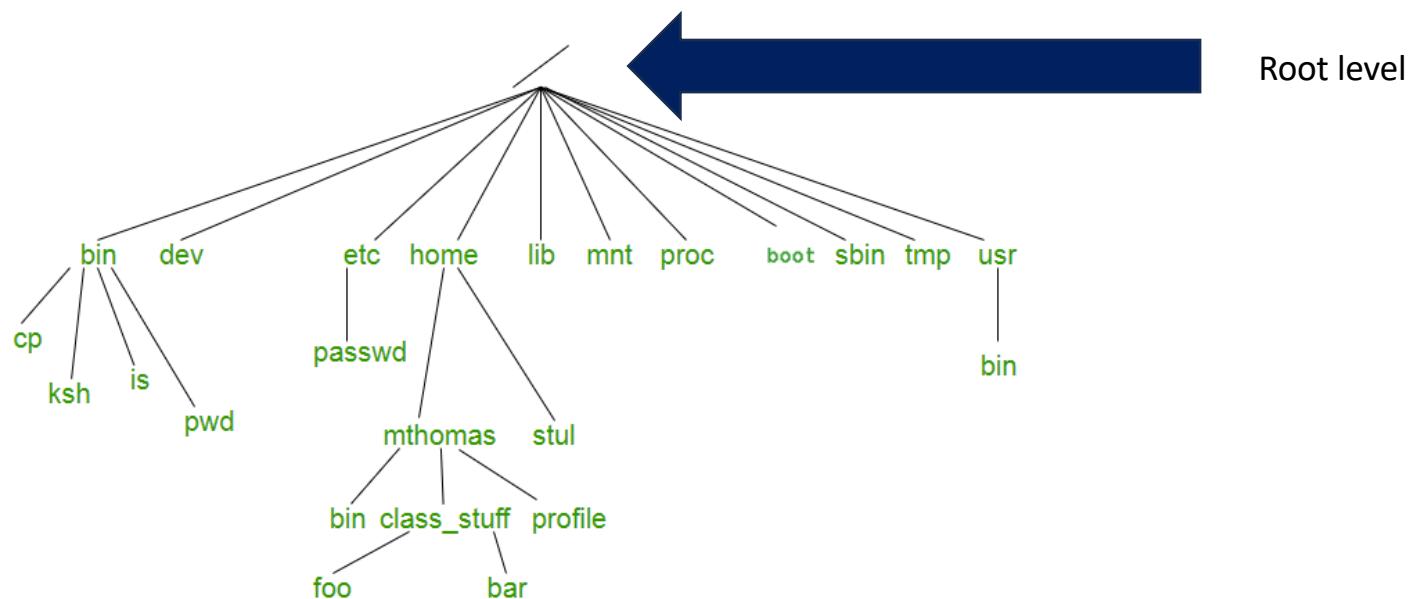
Overview the shell prompt (5)

```
[kfotso@xsede.org@login-ci1 ~]$ █
```



\$ shows that you are a standard user and not administrator (root)
would have meant that you are a standard administrator

Understanding the linux filesystem tree



Basic bash shell commands(1)

```
[kfotso@xsede.org@login-ci1 ~]$ pwd  
/home/kfotso@xsede.org  
[kfotso@xsede.org@login-ci1 ~]$ █
```

- `pwd` stands for "print working directory"
- Very useful to locate your position in the filesystem tree

Basic bash shell commands(2)

```
[kfotso@xsede.org@login-ci1 ~]$ ls  
'$TMPDIR'  
All_PP_nomeds_chr1_01_results.log  
check_nodes.txt  
config.log  
file  
[kfotso@xsede.org@login-ci1 ~]$  
job.log          output{1..36}      slurm-1169215.out  
log.file        README.mdwn       slurm-1288904.out  
make.log         slurm-1047744.out  slurm-1330575.out  
nodelist.txt    slurm-1152814.out  slurm-1330585.out  
output1          slurm-1165091.out  slurm-1330591.out
```

- `ls` stands for “list directory contents”
- Can run “`man ls`” to learn more about a bash shell related command.

Basic bash shell commands(3)

```
[kfotso@xsede.org@login-ci1 ~]$ tree
.
+-- $TMPDIR
|   |-- All_PP_nomeds_chr1_01_results.log
|   |-- check_nodes.txt
|   |-- config.log
|   |-- file
|   |-- job.log
|   |-- log.file
|   |-- make.log
|   |-- nodelist.txt
|   |-- output1
|   |-- output{1..36}
|   |-- README.mdwn
|   |-- SLURM_SUBMIT_DIR
|   |-- spack-build-env.txt
|   |-- spack-build-out.txt
|   |-- test
|   `-- TMPDIR
.
5 directories, 12 files
[kfotso@xsede.org@login-ci1 ~]$
```

- “.” the dot mean your current working directory
- tree - list content of directories in a tree-like format
- It also lists number of directories and files



Understanding absolute vs relative paths (1)

- The file path is defined as a human-readable representation of a file or a folder's location on a computer system
- An absolute path informs a user location on the filesystem from root “/”
- A relative path informs a user location of the filesystem from the current directory “.”

Understanding absolute vs relative paths (2)

```
[kfotso@xsede.org@login-ci1 software]$ pwd  
/projects/kfotso@xsede.org/software  
[kfotso@xsede.org@login-ci1 software]$ █
```

- Example of an **absolute path** from “/” using `pwd`

Understanding absolute vs relative paths (4)

```
[kfotso@xsede.org@login-ci1 software]$ tree  
.  
└── anaconda  
    └── envs  
        └── ccc-env  
            └── bin  
                └── 2to3 -> 2to3-3.9
```

- Overview of the files from “.”

```
[kfotso@xsede.org@login-ci1 software]$ ls ./anaconda/envs/ccc-env/bin/  
2to3      ipython   lzmore          python3           tput  
2to3-3.9  ipython3  matplotlib     python3.9        tset  
captoinfo  lzcat    ncursesw6-config  python3.9-config unlzma
```

- Listing the relative path from “.”

Understanding absolute vs relative paths (5)

```
[kfotso@login-ci1 software]$ ls /projects/kfotso@xsede.org/software/anaconda/envs/ccc-env/bin/
2to3      lzgrep      pygmentize    unxz
2to3-3.9   lzfgrep     python       wheel
captoinfo  lzgrep      python3      wish
clear      lzless      python3.9   wish8.6
```

- Same listing of files could have been done as absolute path from “/”

Basic bash shell commands(4)

```
[kfotso@xsede.org@login-ci1 ~]$ ls  
'$TMPDIR'  
All_PP_nomeds_chr1_01_results.log job.log      output{1..36}        test  
check_nodes.txt    log.file      README.mdwn      TMPDIR  
config.log        make.log      SLURM_SUBMIT_DIR  
file              nodelist.txt spack-build-env.txt  
                  output1       spack-build-out.txt
```

```
[kfotso@xsede.org@login-ci1 ~]$ ls -a  
'.'  
'..'  
'$TMPDIR'  
All_PP_nomeds_chr1_01_results.log  .horovod     .python_history  
.bash_history          .ipython     .RData  
                           .java      README.mdwn  
                           job.log     .Rhistory  
                           .jupyter   .shrc
```

- “-a” stands for “all” so it lists hidden files as well

Basic bash shell commands(5)

```
[kfotso@xsede.org@login-ci1 ~]$ ls -l
total 1432
drwxr-xr-x. 2 kfotso@xsede.org kfotsopgrp@xsede.org      0 May  1 01:57 '$TMPDIR'
-rw-r--r--. 1 kfotso@xsede.org kfotsopgrp@xsede.org    1052 Apr 27 13:31 All_PP_nomad
s_chr1_01_results.log
-rw-r--r--. 1 kfotso@xsede.org kfotsopgrp@xsede.org 249155 Apr 13 18:19 check_nodes.
txt
```



File permission	Number of links	Owner name	Group name	File size	Time of last modification	File or directory

- “-l” stands for “long listing”.

File permission

-rw-r--r--.



“_” means file
Permission for owner
Permission for group
Permission for other

Storage (1)

```
[kfotso@xsede.org@login-ci1 ~]$ curc-quota
```

	Used	Avail	Quota	Limit
/home/kfotso@xsede.org	434M	1.6G	2.0G	
/projects/kfotso@xsede.org	100G	151G	250G	
/scratch/alpine1	26368G	4352G	30720G	

- curc-quota shows disk usage

Storage (2)

```
[kfotso@xsede.org@login-ci1 ~]$ du -sh /scratch/alpine/kfotso@xsede.org/data  
1.0K  /scratch/alpine/kfotso@xsede.org/data
```

- du stands for disk usage

Storage (3)

Filesystem	Size	Used	Avail	Use%	Mounted
on devtmpfs	3.8G	0	3.8G	0%	/dev
tmpfs	3.8G	4.0K	3.8G	1%	/dev/shm
tmpfs	3.8G	393M	3.5G	11%	/run
tmpfs	3.8G	0	3.8G	0%	/sys/fs
/cgroup					
/dev/mapper/vg_root-lv_root	145G	4.6G	141G	4%	/
/dev/sdb1	7.8G	84K	7.4G	1%	/tmp
/dev/sda2	1014M	343M	672M	34%	/boot
/dev/sda1	200M	5.8M	195M	3%	/boot/efi
/dev/mapper/vg_root-lv_var	8.0G	1.9G	6.2G	23%	/var
sgate1-data.rc.int.colorado.edu:/gpfs/summit/scratch	463G	11G	453G	3%	/scratch/summit
c3nsd1.rc.int.colorado.edu:/gpfs/alpine1/scratch	1.9P	1.2P	661T	65%	/scratch/alpine
isilon1-data.rc.int.colorado.edu:/ifs/curc/sw	433T	260T	159T	63%	/curc/software
isilon1-data.rc.int.colorado.edu:/ifs/curc/slurm	433T	260T	159T	63%	/curc/slurm
isilon1-data.rc.int.colorado.edu:/ifs/curc/home	433T	260T	159T	63%	/home

- df shows the filesystem disk space usage

Basic bash shell commands(6)

```
[kfotso@xsede.org@login-ci1 ~]$ cd /scratch/alpine/$USER  
[kfotso@xsede.org@login-ci1 kfotso@xsede.org]$ █
```

- “cd” means change my working directory

Basic bash shell commands(7)

```
[kfotso@xsede.org@login-ci1 kfotso@xsede.org]$ pwd  
/projects/kfotso@xsede.org  
[kfotso@xsede.org@login-ci1 kfotso@xsede.org]$ cp /projects/ssills24@xsede.org/new_reduce_multi.slurm .
```



Source



Destination

- Copy file to my current working directory

Basic bash shell commands(8)

```
[kfotso@xsede.org@login-ci1 software]$ cp -r mkdir /projects/kfotso@xsede.org/  
[kfotso@xsede.org@login-ci1 software]$ █
```

- Copy folder to my project directory.
- “-r” means recursive

Basic bash shell commands(9)

- Copy folder to my project directory as “mkdir-from-Sam”

```
[kfotso@xsede.org@login-ci1 software]$ cp -r mkdir /projects/kfotso@xsede.org/mkdir-from-Sam  
[kfotso@xsede.org@login-ci1 software]$ █
```

Basic bash shell commands(10)

```
[kfotso@xsede.org@login-ci1 software]$ mv mkdir mkdir-renamed  
[kfotso@xsede.org@login-ci1 software]$ █
```

- Renaming a folder or file. “mv” means move.

Basic bash shell commands(11)

```
[kfotso@xsede.org@login-ci1 software]$ cat demonstration  
This is the content from the demonstration file  
[kfotso@xsede.org@login-ci1 software]$ █
```

- "cat" stands for concatenate
- Outputs content of a file
- You can play with the nano command as well

Basic bash shell commands(12)

```
#!/bin/bash -l

# Run this file using 'sbatch reduce.sbatch.summit'
#
#SBATCH --account=amc-general
#SBATCH --partition=amilan

# Give this job a name
#SBATCH --job-name=reduce-sim

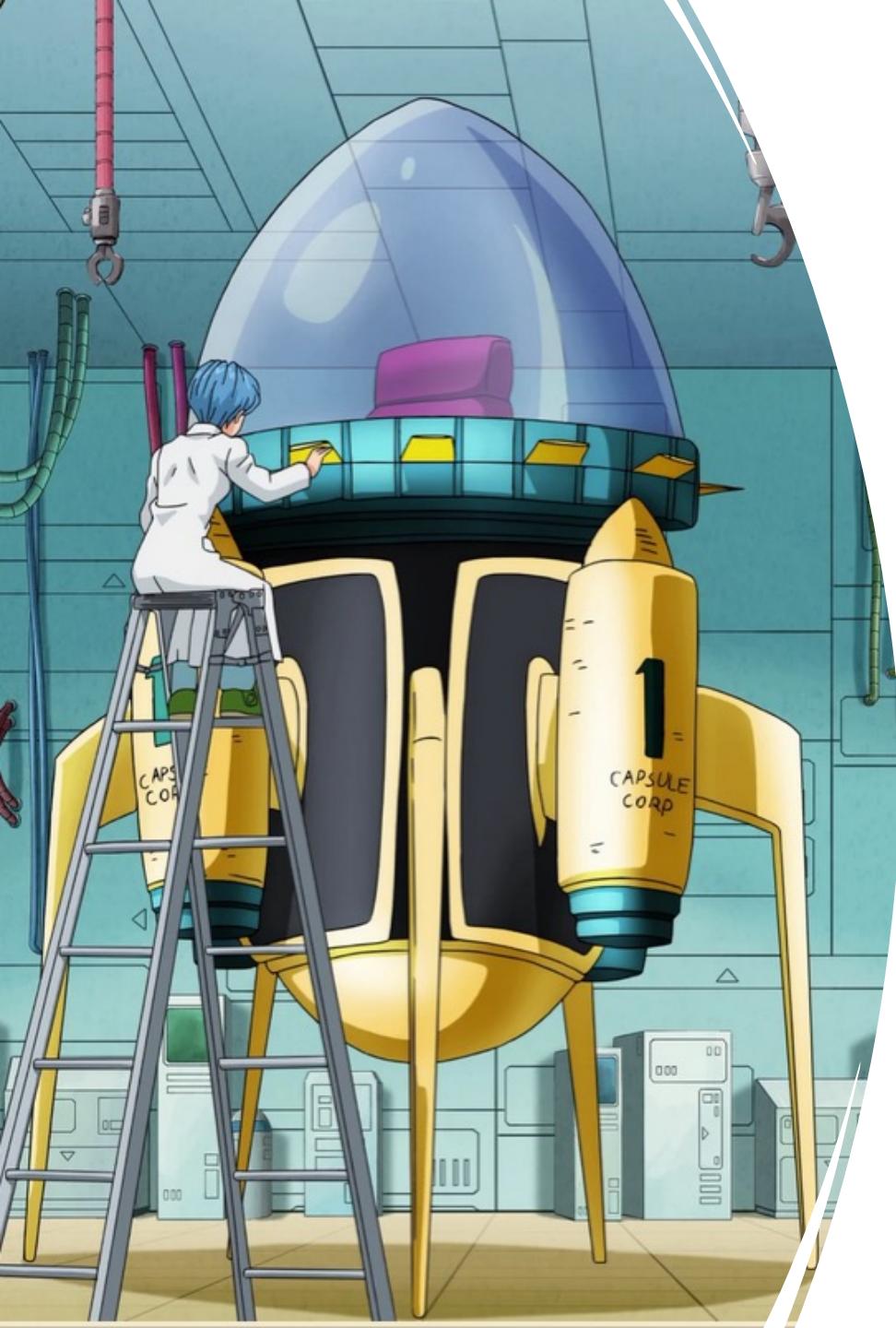
# Join standard output and error to a single file
#SBATCH --output=reduce.qlog

# Request time needed for job to run (default: 12 hours)
#SBATCH --time=00:40:00
#SBATCH --qos=normal

# Send an email when the job begins and when it ends running
#SBATCH --mail-type=BEGIN,FAIL,END

# Whom to send the email to
```

- nano
new_reduce_multi.slurm
- To edit a file



I- Part 1) Elements of computing

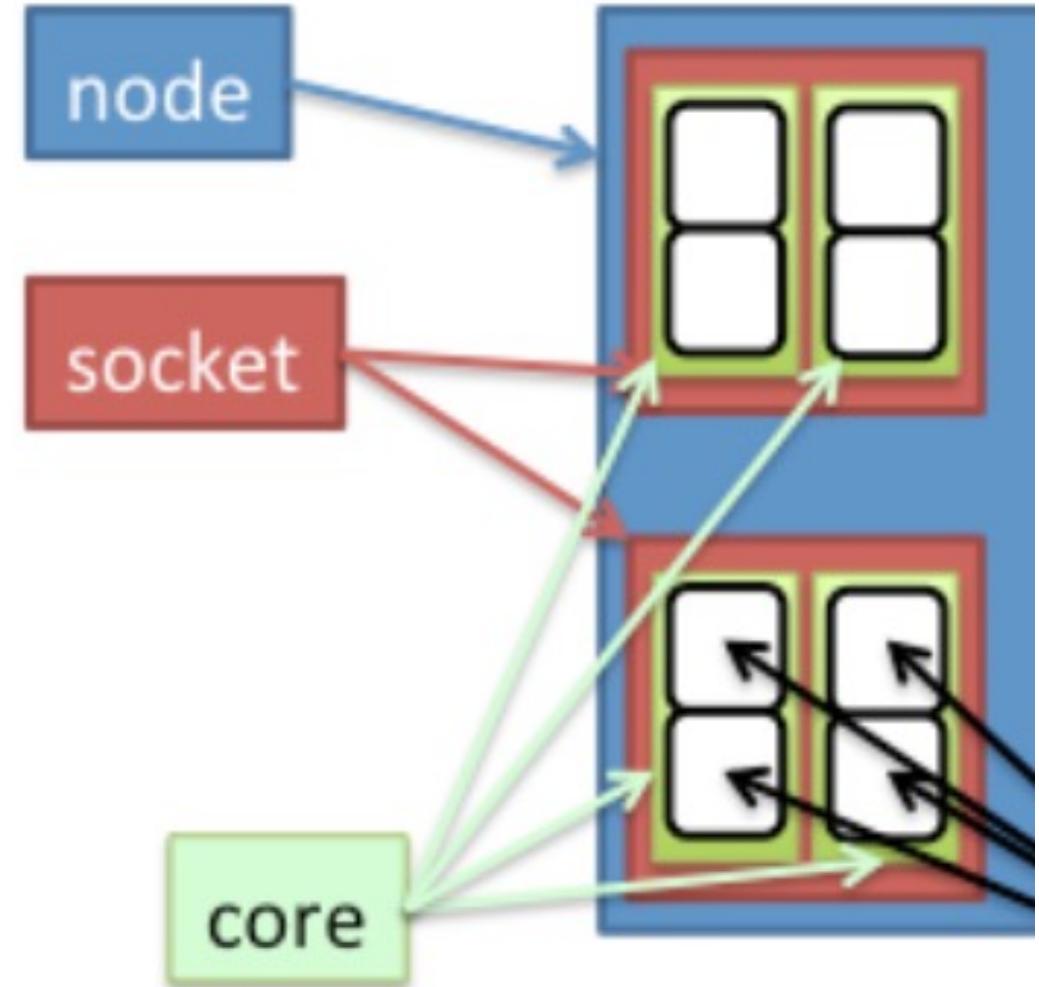
Parts of the processing chip (1)



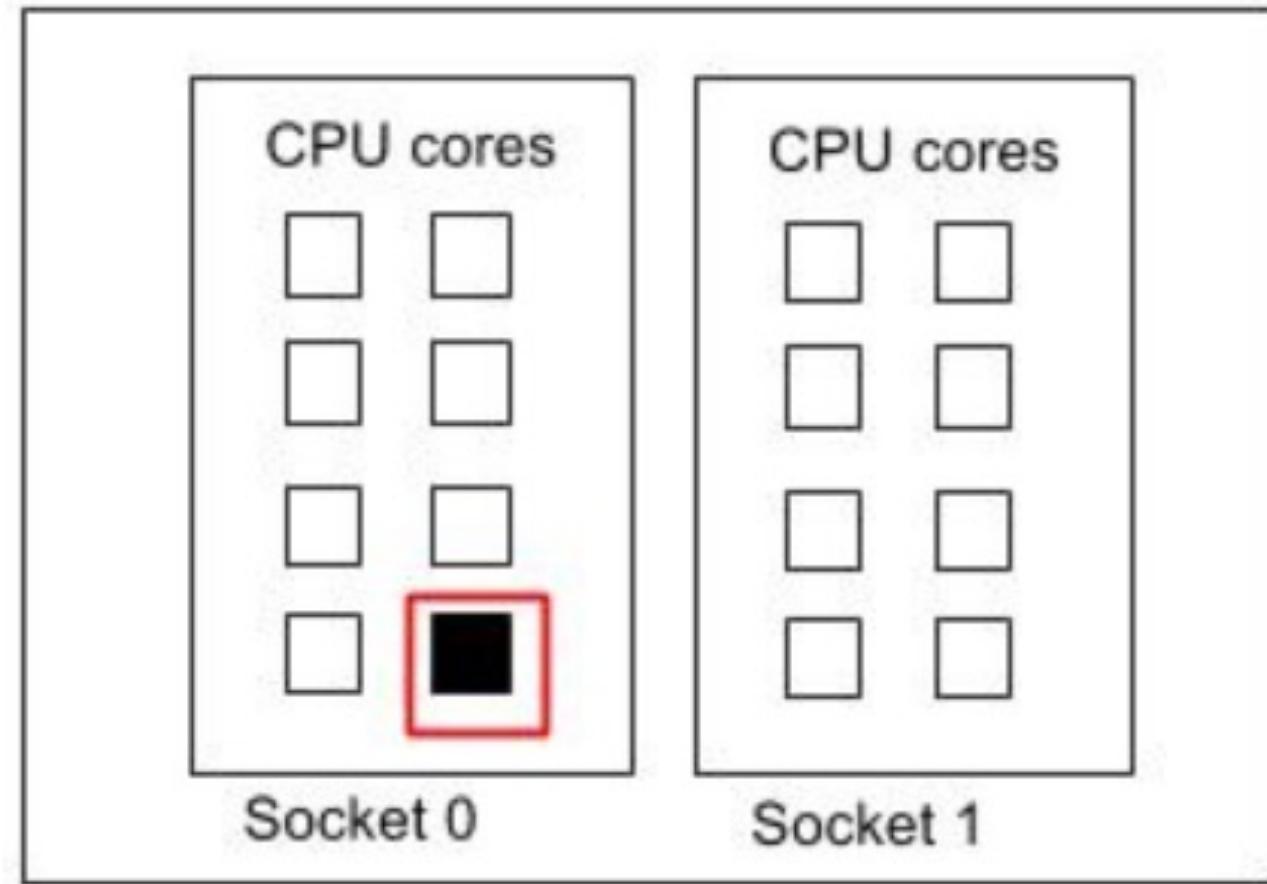
- A socket can be considered an entire computing chip in modern computers.
- Contains more than 1 core.
- On supercomputers a node usually contains more than 1 socket.

Parts of the processing chip (2)

A core contains an Arithmetic and Logic Unit (**ALU**) + Registers

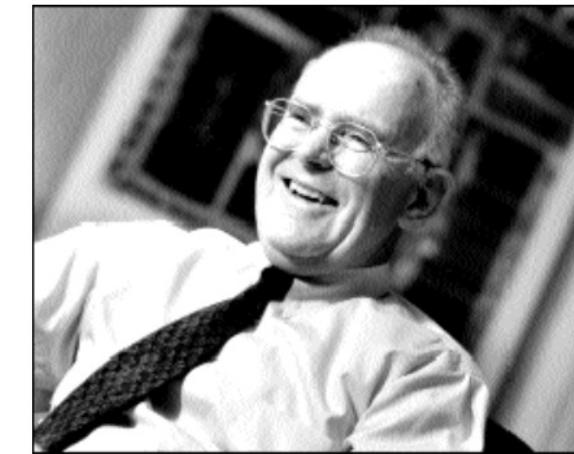
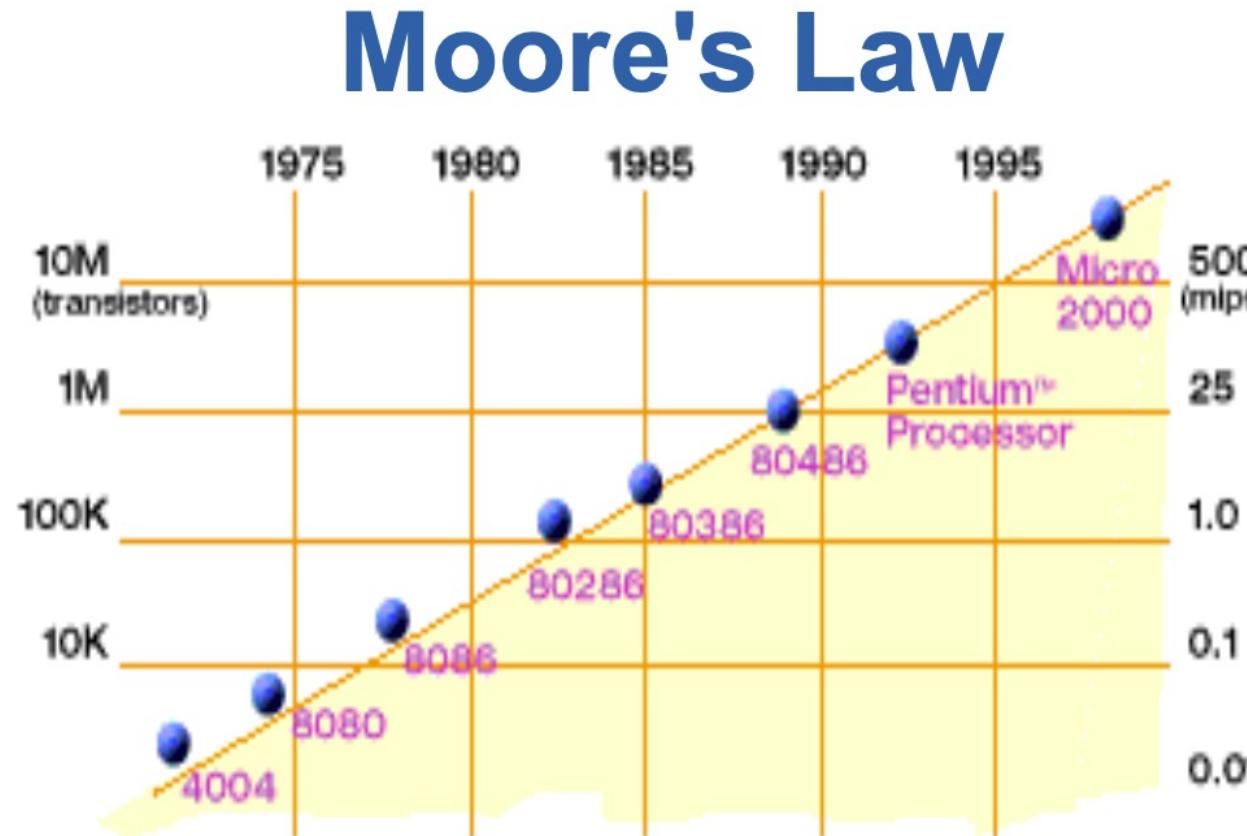


Parts of the
processing
chip (2)



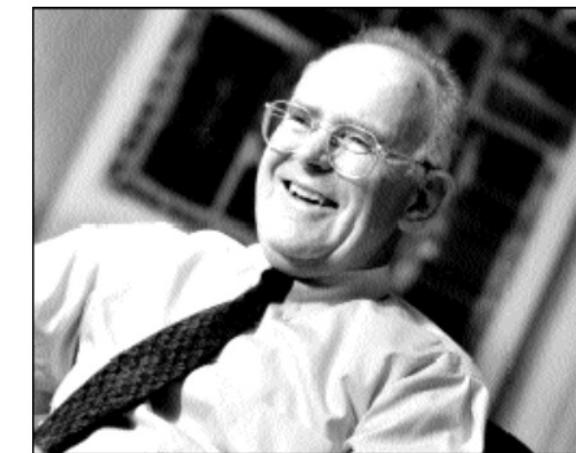
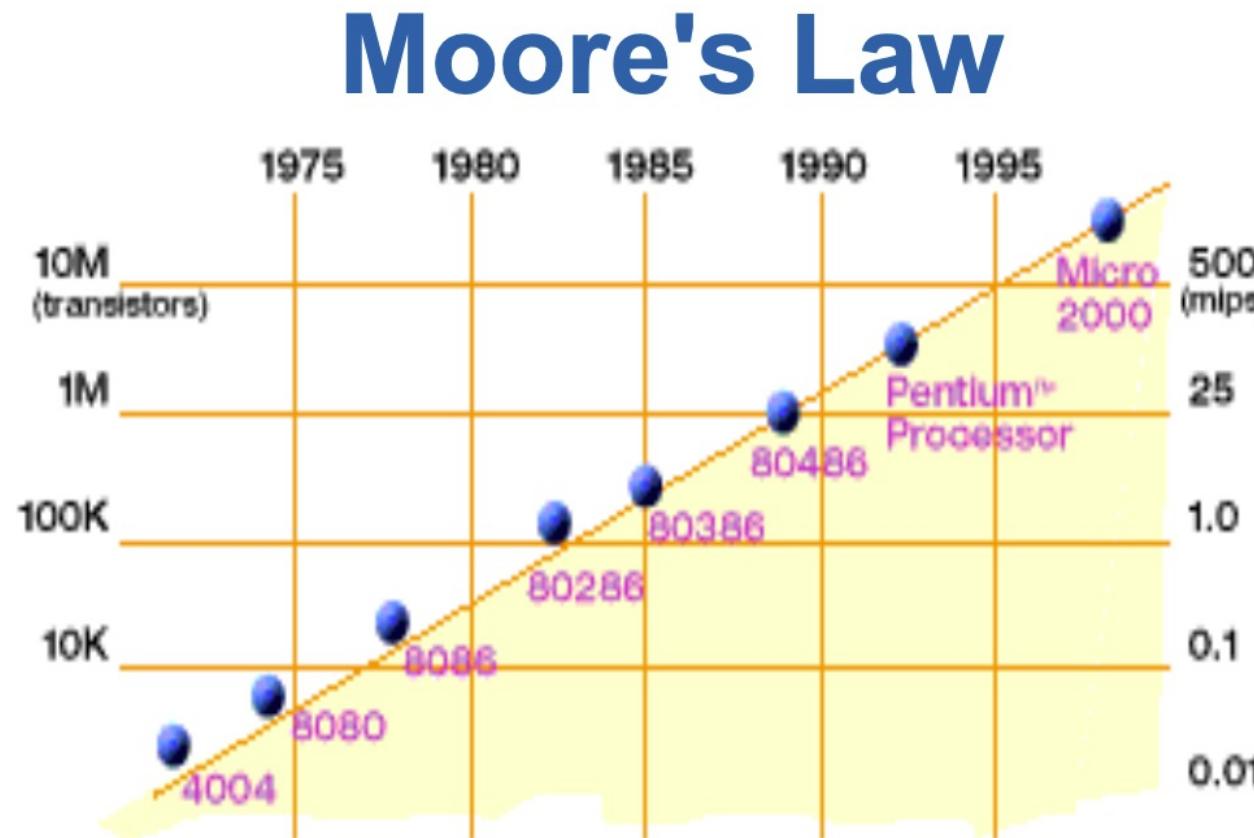
Moore's law (1)

- Transistor count doubles in a chip every 18 months.

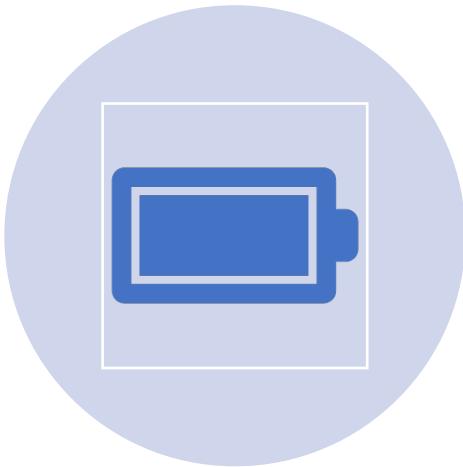


Moore's law (2)

- Worked until around 2010 where it started to plateau



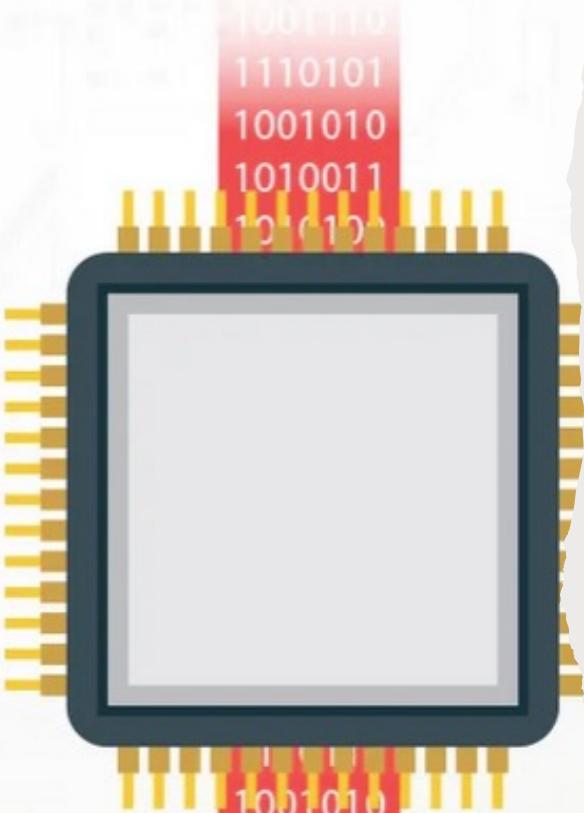
Why?



TRADITIONALLY, POWER DENSITY STAYS CONSTANT AS
CHIPS FEATURE GOT SMALLER.



HOWEVER AROUND 2010, CLOCK FREQUENCY COULD
NOT BE RAISED FURTHER BECAUSE THE HEAT DISSIPATION
OR POWER WOULD HAVE GONE TOO FAR.

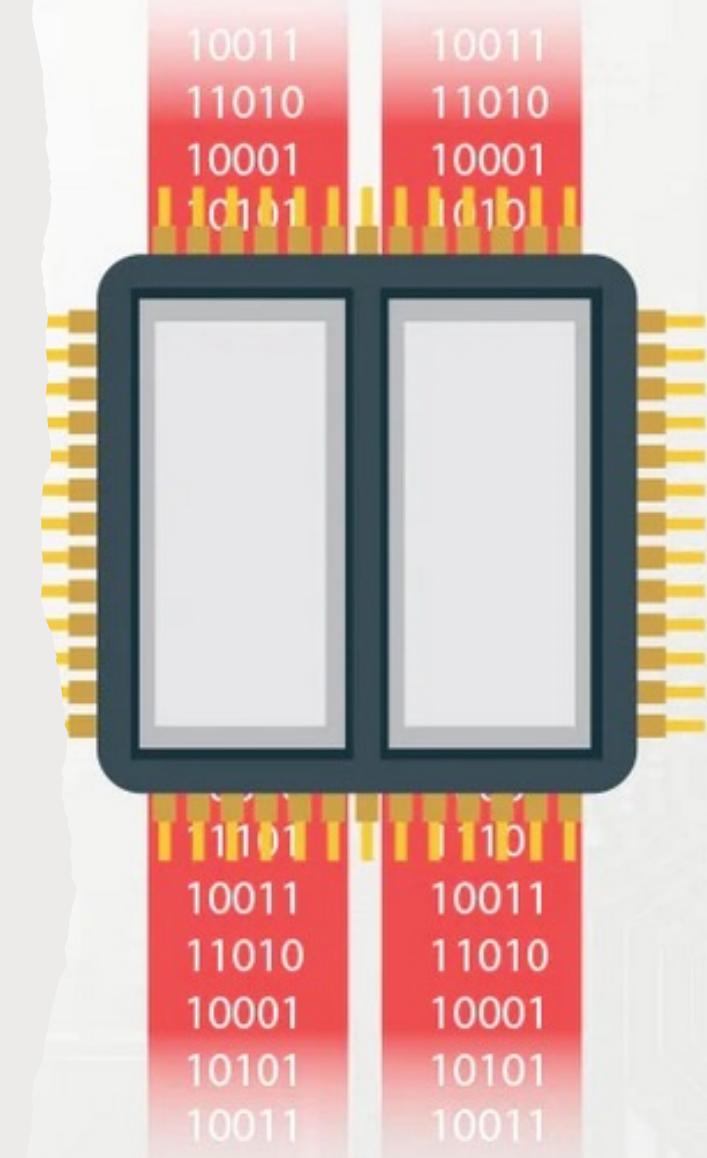


Single-core

Increase need of parallelism

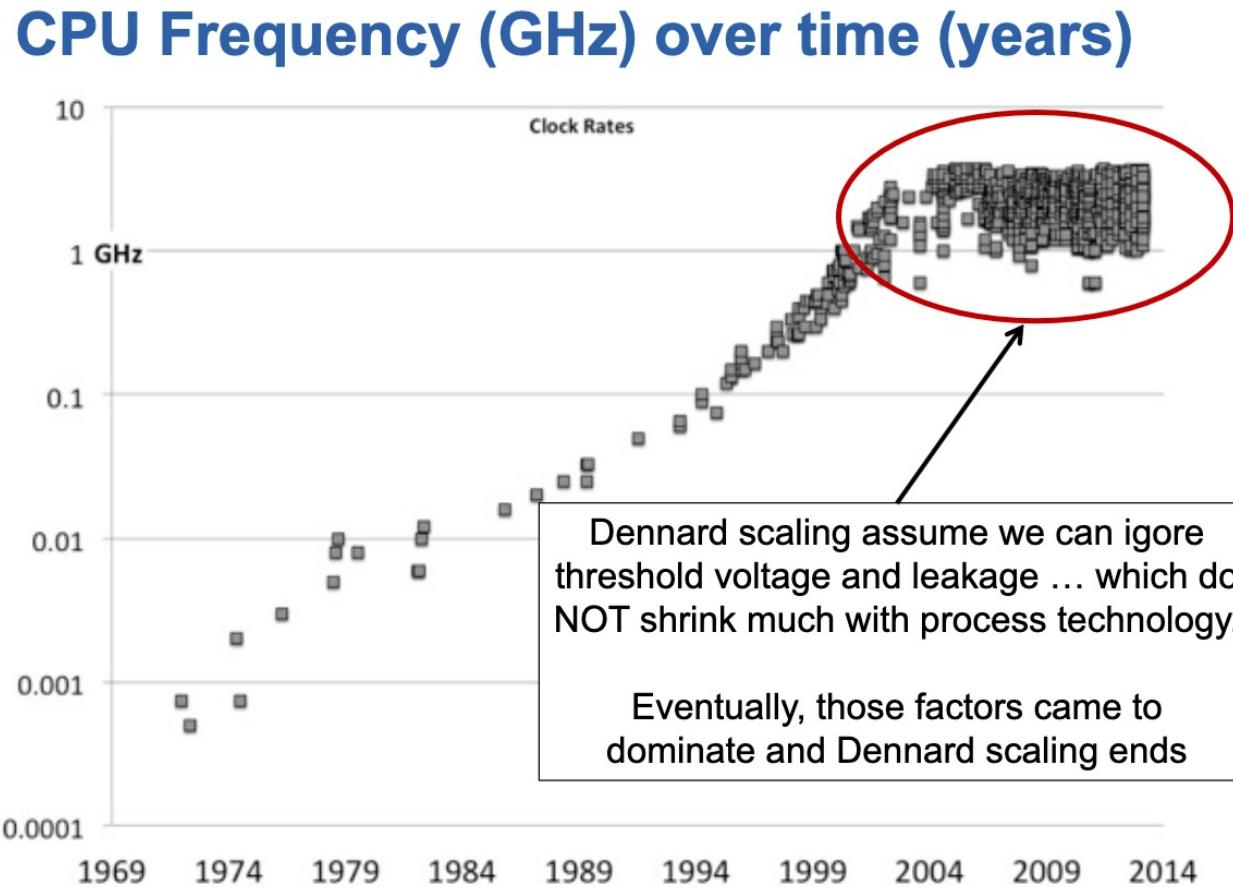
Reduce power
by adding
cores of small
frequency

2 cores at half
frequency f
can achieve
the same
throughput as a
single core of
frequency f



Dual Core

Moore's law plateau

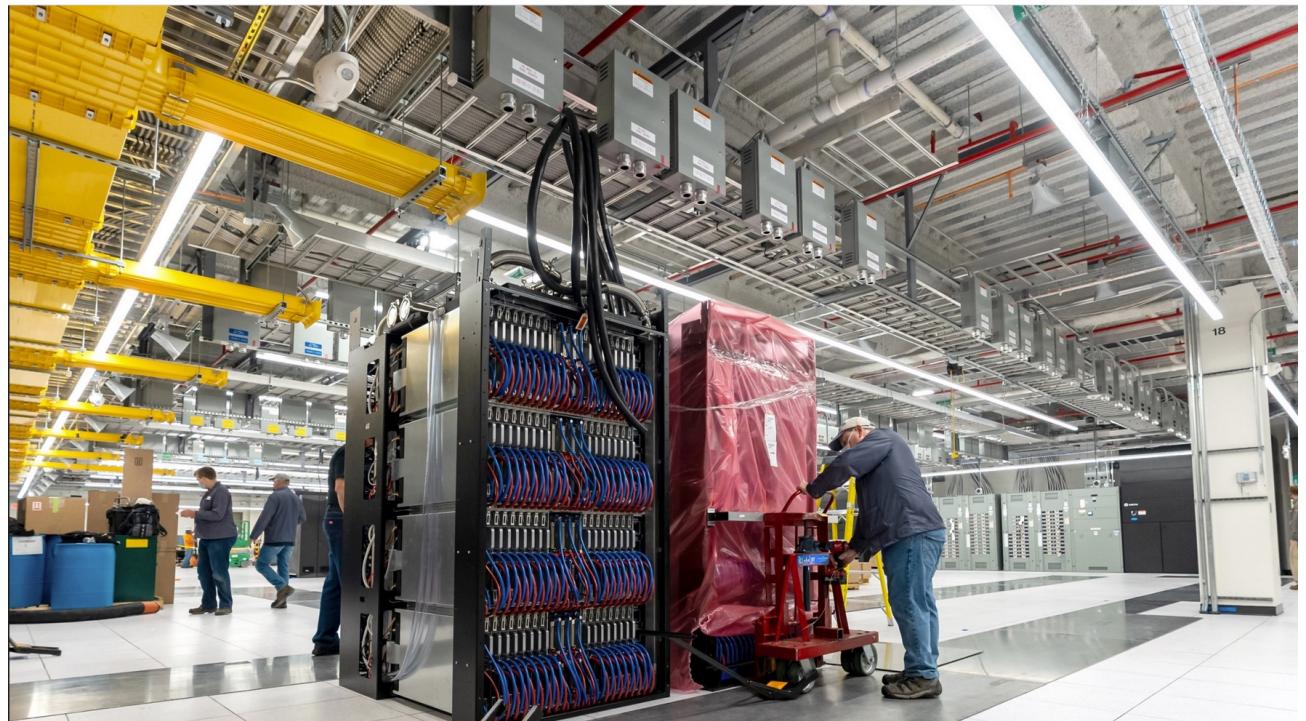
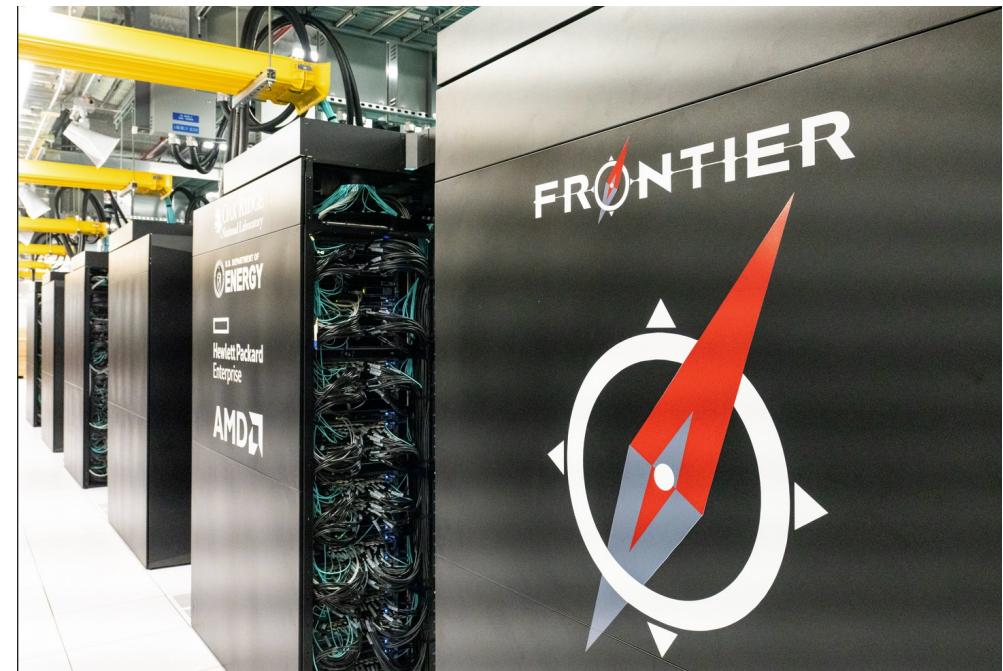


- Number of cycles per seconds plateaued in the 2010s
- Why?



A look at an HPC cluster(1)

- Frontier supercomputer OAK Ridge National lab.
[AMD Instinct MI250X GPU; 8,699,904 Cores]
- Most powerful at the moment benchmarking with Linpack (e.g. $Ax=b$)



A look at an HPC cluster(2)



June 2023: The TOP 10 Systems (52% of the Total Performance of Top500)

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	GFlops/Watt
1	DOE / OS Oak Ridge Nat Lab	Frontier, HPE Cray Ex235a, AMD 3 rd EPYC 64C, 2 GHz, AMD Instinct MI250X , Slingshot 10	USA	8,699,904	1,194	71	22.7	52.6
2	RIKEN Center for Computational Science	Fugaku, ARM A64FX (48C, 2.2 GHz), Tofu D Interconnect	Japan	7,299,072	442.	82	29.9	14.8
3	EuroHPC /CSC	LUMI, HPE Cray EX235a, AMD 3 rd EPYC 64C, 2 GHz, AMD Instinct MI250X , Slingshot 10	Finland	1,268,736	304.	72	2.94	52.3
4	EuroHPC/CINECA	BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 (108C) , Quad-rail NVIDIA HDR100	Italy	1,824,768	239.	78	7.4	32.2
5	DOE / OS Oak Ridge Nat Lab	Summit, IBM Power 9 (22C, 3.0 GHz), NVIDIA GV100 (80C) , Mellanox EDR	USA	2,397,824	149.	74	10.1	14.7
6	DOE / NNSA Livermore Nat Lab	Sierra, IBM Power 9 (22C, 3.1 GHz), NVIDIA GV100 (80C) , Mellanox EDR	USA	1,572,480	94.6	75	7.44	12.7
7	National Super Computer Center in Wuxi	Sunway TaihuLight, SW26010 (260C) , Custom Interconnect	China	10,649,000	93.0	74	15.4	6.05
8	DOE / OS NERSC - LBNL	Perlmutter HPE Cray EX235n, AMD EPYC 64C 2.45GHz, NVIDIA A100 , Slingshot 10	USA	706,304	64.6	71	2.59	27.4
9	NVIDIA Corporation	Selene NVIDIA DGX A100, AMD EPYC 7742 (64C, 2.25GHz), NVIDIA A100 (108C) , Mellanox HDR	USA	555,520	63.4	80	2.64	23.9
10	National Super Computer Center in Guangzhou	Tianhe-2A NUDT, Xeon (12C), MATRIX-2000 (128C) + Custom Interconnect	China	4,981,760	61.4	61	18.5	3.32

- Alpine has >18,000 cores!

Anatomy of supercomputing



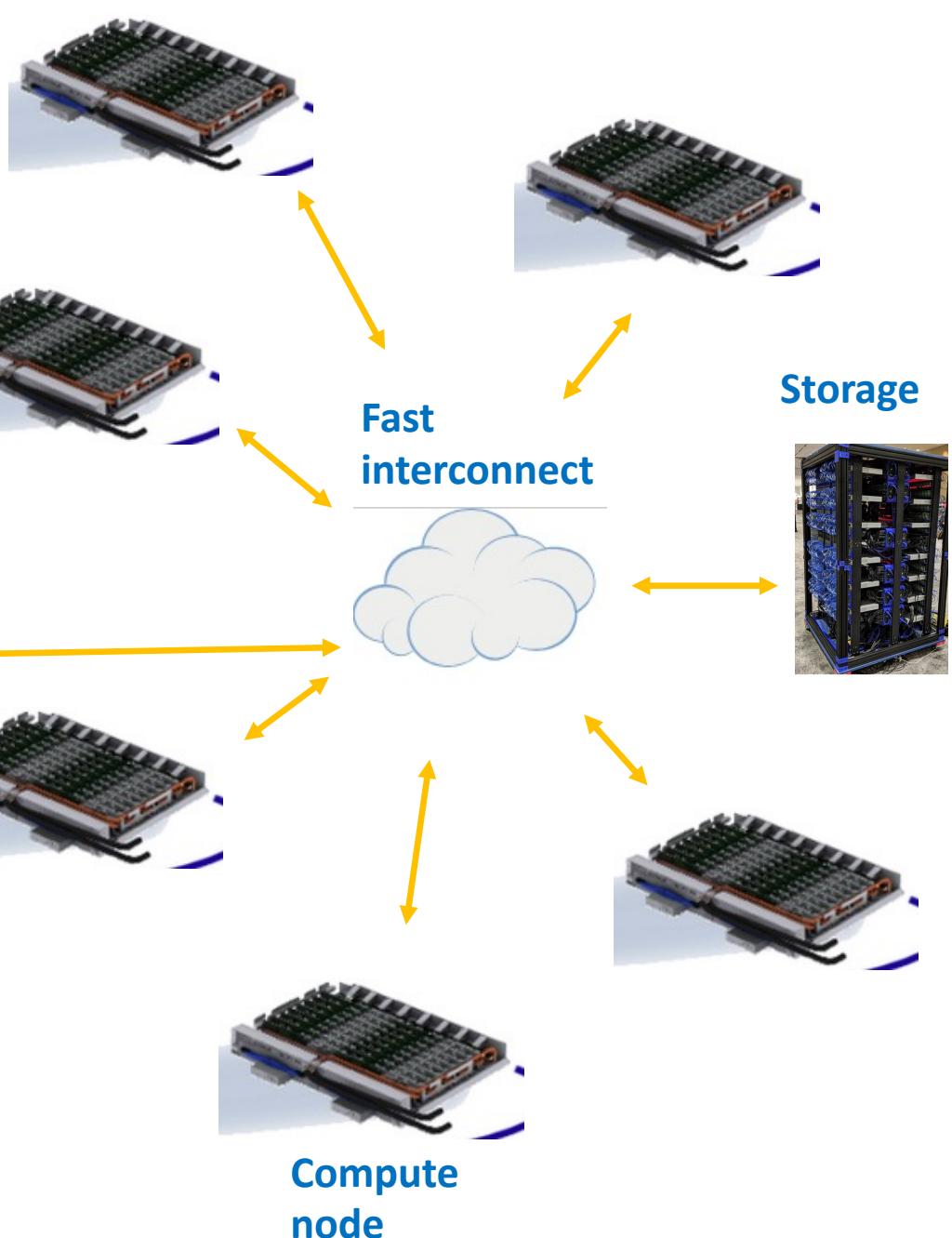
Login node



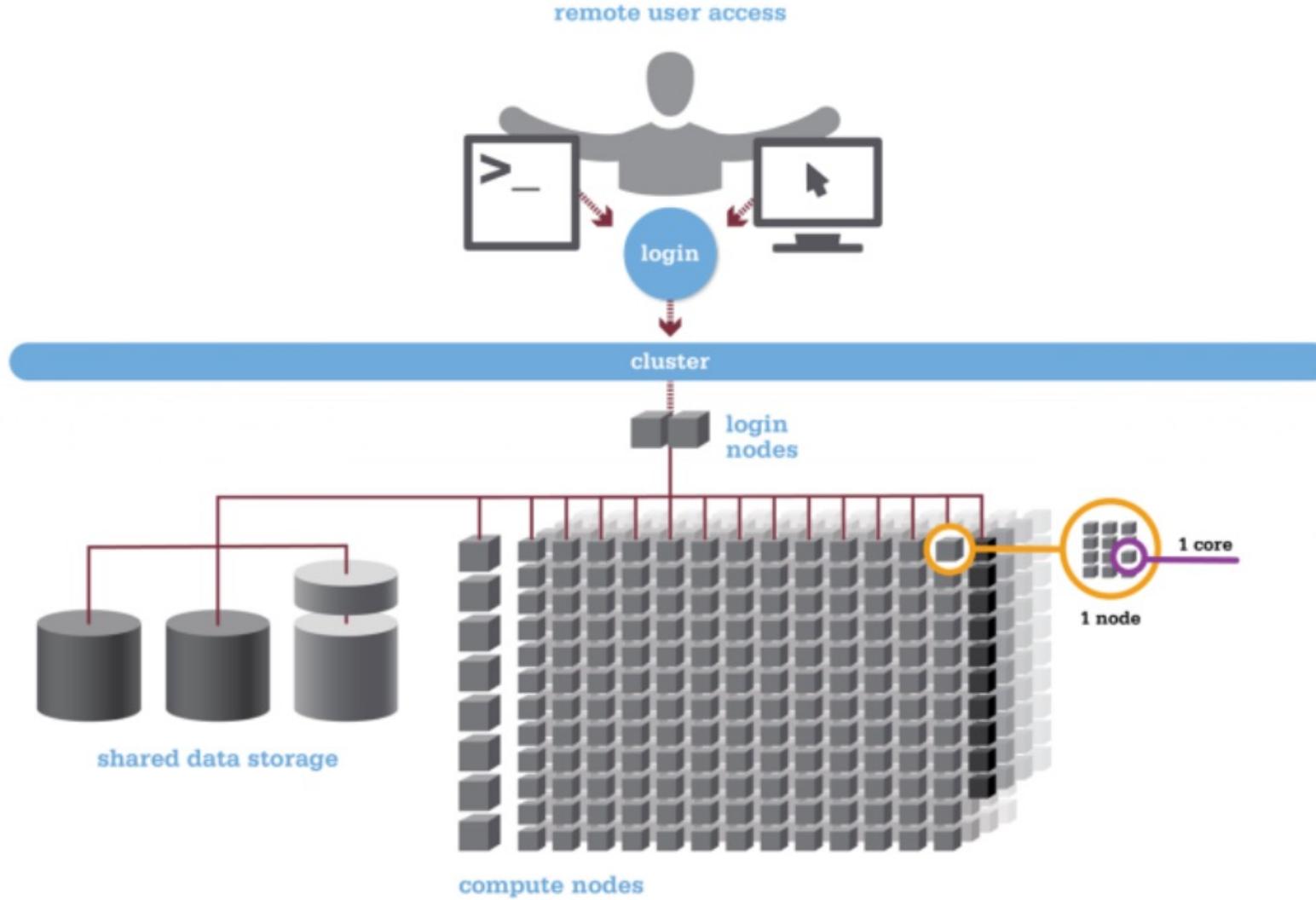
Slurm:
scheduler



Laptop or
PC



Architecture of a supercomputer

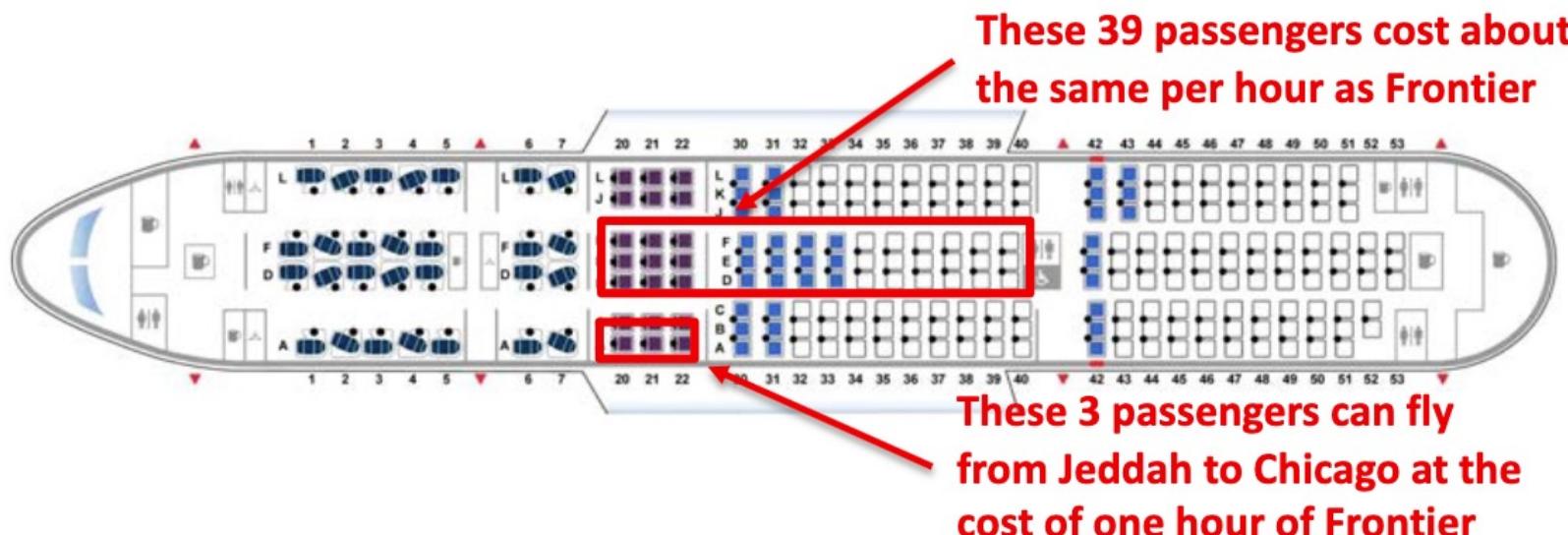


- Login nodes. To log into the system, cd into directories, look at files etc ...
- Compute node. Dedicated to do the computation
- The slurm scheduler controls access to the compute nodes to avoid a tragedy of the commons ...

Why Slurm?



- Allows for better tracking of resource used per user
- Better management of resource across user base
- More sustainable and less tragedy of the commons
- Widely used and supported by applications in case of bugs.



Why Slurm?

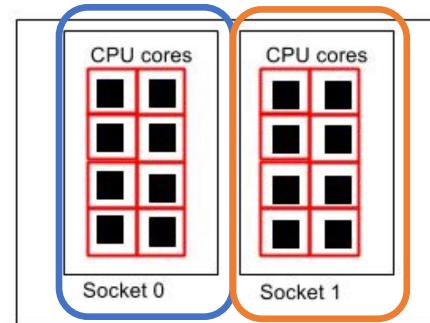
- Bulma request 8 cores from Node 1
- Dr. Brief requests 8 cores as well from Node 1
- Gohan requests 16 cores from Node 2
- No tragedy of the common!!!



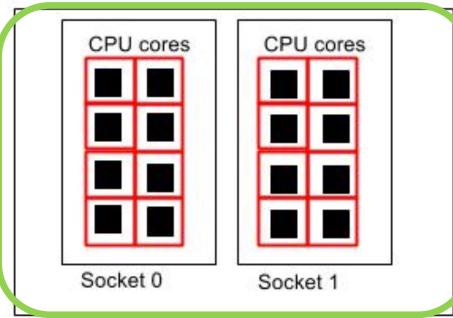
Slurm
scheduler



Node 1

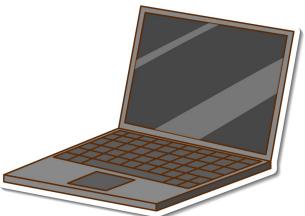


Node 2

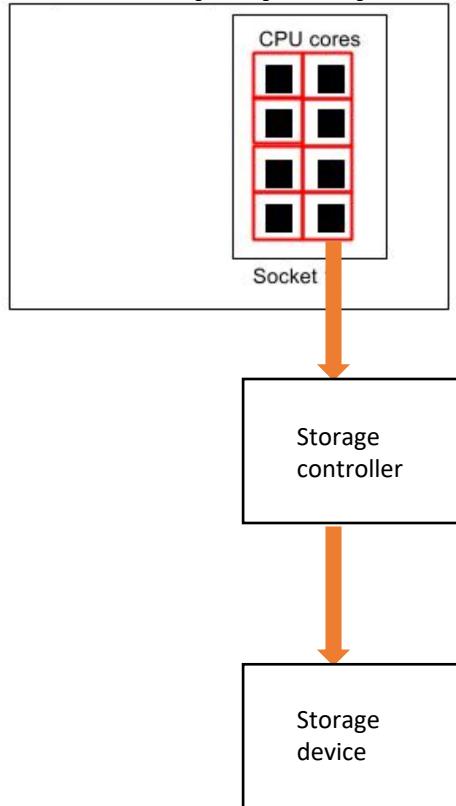


Overview of storage [PC]

Laptop

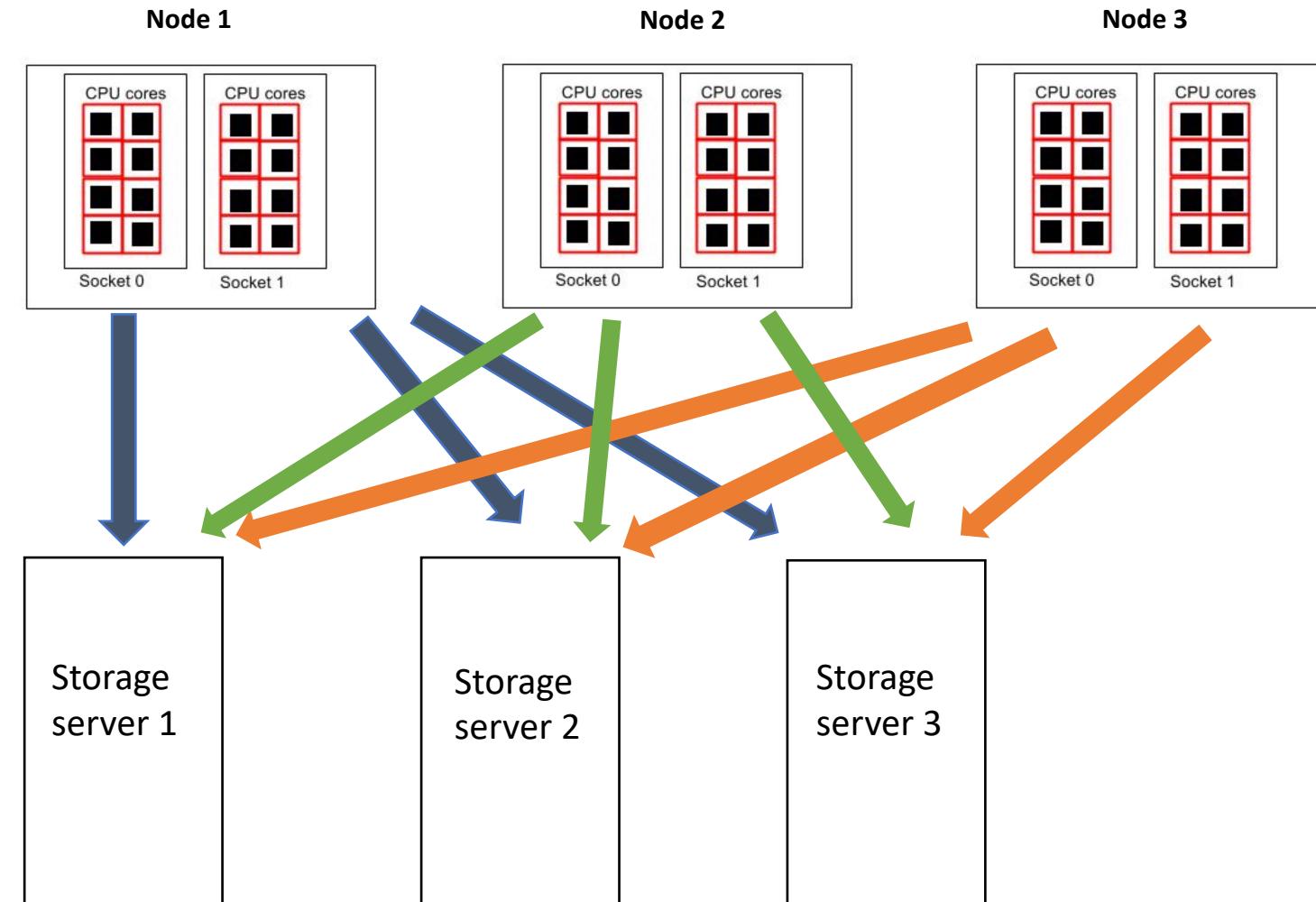


Laptop chip



- Compute core talks to the storage controller which talks to the storage device to get the data
- Path between software to storage is short.
- Low latency and low bandwidth.

Overview of storage [HPC]



- Files distributed across multiple servers.
- Much Higher bandwidth.
- In general higher latency.

Filesystems on High Performance Computing

- Network filesystem (NFS): Allows users to access files and directories located on remote servers



GPFS High performance filesystem

- Optimized for heavy I/O pipelines
- Optimized for heavy parallelized pipelines



II- Part 2) Intro to Alpine



Hardware (1)



>317 compute nodes and >18,080 cores officially.



184 CPU nodes (HDR IB interconnect)



>12 high memory nodes (1TB)



8 NVIDIA A100 GPU and 12 AMD GPU MI100 nodes. (3 GPUs per node) + (2X25 Ethernet interconnect)



NVIDIA GPU tend to be more busy but AMD GPU are becoming popular.

Official Github pages:

- CU Anschutz HPC official Github page.

- Note: You can submit pull issues
Pull requests as well.

The screenshot shows a GitHub repository page for 'kf-cuanschutz / CU-Anschutz-HPC-documentation'. The repository is public and has 2 issues, 1 branch, and 0 tags. The commit history lists 63 commits from 'kf-cuanschutz' over the past 2 weeks, with various file updates like README.md, Globus-local-entry-point-files, and ALDEEx2-R-package-installation.md. The repository has 6 stars, 1 watching, and 0 forks. It includes sections for About (bioinformatics tutorial on the Alpine cluster), Releases (no releases published), and Packages (no packages published). The README file contains a CC-BY-NC-ND license logo and the text 'CU Anschutz-HPC-documentation'.

About
All documentation associated with bioinformatics tutorial on the Alpine cluster

Readme

Activity

6 stars

1 watching

0 forks

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

Code Issues 2 Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

kf-cuanschutz Update README.md ... a9741f6 2 weeks ago 63 commits

Globus-local-entry-point-files Delete Overview_of_Petalibrary.pdf 2 weeks ago

Office-hours-presentation-files Introduction to Petalibrary 2 weeks ago

ALDEEx2-R-package-installation.md Fixed the hyperlink that was pointing to globus 3 months ago

Alpine-cluster-maintenance.md replaced Preventative maintenance with FAQ 3 months ago

Alpine-pipeline-opt-FAQ.md Added Job arrays upper limit 2 months ago

EcholocateR.md small typo fix 3 months ago

MATLAB-kernel-on-Jupyterlab.md Create MATLAB-kernel-on-Jupyterlab.md 2 weeks ago

README.md Update README.md 2 weeks ago

cellRangerRkit.md Few changes added (e.g. indentation on Part 1) 2 months ago

README.md

CC BY NC ND

CU Anschutz-HPC-documentation

- CU Boulder curc doc:

<https://curc.readthedocs.io/en/latest/access/logging-in.html>

Logging into Alpine(1)

Logging into Alpine from the Shell App

Note

Make sure you already have your XSEDE/ACCESS user name and password set up before proceeding and Duo 2-factor authentication set up for your ACCESS/XSEDE account

1. Visit <https://ondemand-rmacc.rc.colorado.edu> You will be redirected to CILogon. From there, make sure you select the ACCESS CI (XSEDE) as your identity provider and then click the "Log On" button.

CILogon

The image shows two screenshots of the CILogon interface. The top screenshot is titled 'Consent to Attribute Release' and displays a list of requested information: 'Your CILogon user identifier', 'Your name', 'Your email address', and 'Your username and affiliation from your identity provider'. The bottom screenshot is titled 'Select an Identity Provider' and shows a dropdown menu with 'ACCESS CI (XSEDE)' selected. There is also a checkbox for 'Remember this selection' and a help icon.

Consent to Attribute Release

OOD RMACC requests access to the following information. If you do not approve this request, do not proceed.

- Your CILogon user identifier
- Your name
- Your email address
- Your username and affiliation from your identity provider

Select an Identity Provider

ACCESS CI (XSEDE) ?

Remember this selection ?

Logging into Alpine(2)

The screenshot shows the Research Computing OnDemand web interface. At the top, there is a dark navigation bar with several menu items: a logo, "Files", "Jobs", "Clusters", "Interactive Apps", and "My Interactive Sessions". The "Clusters" menu item is highlighted with a red circle, and a dropdown menu is open, listing two options: ">_Alpine Shell" and ">_Blanca Shell". Below the navigation bar is the University of Colorado Boulder logo, which consists of a gold "CU" monogram and the text "Research Computing UNIVERSITY OF COLORADO BOULDER". A main message states: "OnDemand provides an integrated, single access point for all of your HPC resources." Below this is a section titled "Message of the Day" with the text: "Welcome to the University of Colorado Research Computing." Underneath this, there is a "Quick Links" section containing several links: "CU Boulder RC Status", "Research Computing User Guide", "Research Computing at CU Boulder", "RMACC @ Ask.Cyberinfrastructure", and "Need help? Email (rc-help@colorado.edu)".

OnDemand provides an integrated, single access point for all of your HPC resources.

Message of the Day

Welcome to the University of Colorado Research Computing.

Quick Links

[CU Boulder RC Status](#)

[Research Computing User Guide](#)

[Research Computing at CU Boulder](#)

[RMACC @ Ask.Cyberinfrastructure](#)

Need help? Email (rc-help@colorado.edu)

Storage(1)

01

Home filesystem
(2G). Backed up +
for hosting config
files.

02

Project filesystem
(250G). Backed up
-> for package
installation

03

“cd .snapshot” to
access those back
ups

04

‘curc-quota’ or
‘du’ to check on
space

Storage(2)

	Used	Avail	Quota	Limit
/home/kfotso@xsede.org	289M	1.8G		2.0G
/projects/kfotso@xsede.org	144G	107G		250G
/scratch/alpine1	7786G	20825G		28611G



Storage(3)

- Scratch space (10 TB)
- GPFS filesystem
- Very suitable for parallel application + heavy I/O
- Purged every 90 days

Username Aliasing

- `/home/foo@xsede.org/ → /home/.xsede.org/foo/`
- `/projects/foo@xsede.org -> /projects/.xsede.org/foo/`
- `/scratch/alpine/foo@xsede.org -> /scratch/alpine/.xsede.org/foo/`

Anaconda config

- Edit the .condarc file located in \$HOME/.condarc with the editor of choice (e.g. nano, vim etc ...)
- Paste the following 4 lines and paste and exit.

pkgs_dirs:

- /projects/.xsede.org/foo/.conda_pkgs

envs_dirs:

- /projects/.xsede.org/foo/software/anaconda/envs

Scheduler Slurm



acompile --ntasks=1 --time=00:30:00 to build packages and do some testing. (max 12 hours)



sinteractive --ntasks-per-node=2 --nodes=2 --partition=atesting to test pipelines. (max 1 hour)

```
[kfotso@xsede.org@login-ci1 kfotso@xsede.org]$ sinfo --partition=aa100
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
aa100        up 1-00:00:00      1  resv c3gpu-a9-u33-1
aa100        up 1-00:00:00      7  mix  c3gpu-a9-u31-1,c3gpu-a9-u35-1,c3gpu-c2-u
aa100        up 1-00:00:00      4  alloc c3gpu-a9-u29-1,c3gpu-c2-u[7,13,15]
```

sinfo

Can be used to get
information about a node.

Package availability (1)

Some packages that have been built and accessible through lmod.

Adding new packages through lmod takes a lot of round of approval so it is recommended to build them locally.

Solutions: (cmake+make), Anaconda, pip, containers, spack etc ...

Submit a ticket at rc-help@Colorado.edu so that I can build it for you locally.

LMOD (CONTEXT)

- Modern environment module system for HPC
- Initially introduced by Robert McLay from TACC (UT Austin) in 2011.



LMOD (CONTEXT)

- Users have on HPC have different needs.
- Applications, compilers, libraries, versions being used might be different.



LMOD (benefits)

- Users do not need to know where software is installed
- Environment variable to interface packages can be set (e.g. picard).
- Very useful for software reproducibility



LMOD (benefits)

- Supports software hierarchy
- Users can set their environment modules at will.



List of officially available bio modules

- “acompile”
- “module avail”

<code>alphafold/2.2.0</code>	<code>bcftools/1.16</code>	<code>cellranger/7.1.0</code>
<code>alphafold/2.3.1 (D)</code>	<code>bedtools/2.29.1</code>	<code>cutadapt/4.2</code>
<code>bamtools/2.5.2</code>	<code>bowtie2/2.5.0</code>	<code>fastqc/0.11.9</code>
<code>bbtools/39.01</code>	<code>bwa/0.7.17</code>	<code>gatk/4.3.0.0</code>

<code>htslib/1.16</code>	<code>picard/2.27.5</code>	<code>sra-toolkit/3.0.0</code>
<code>multiqc/1.14</code>	<code>plink2/2.00a2.3</code>	<code>star/2.7.10b</code>
<code>nextflow/22.10.6</code>	<code>qiime2/2023.5</code>	<code>trimmmomatic/0.39</code>
<code>nextflow/23.04 (D)</code>	<code>samtools/1.16.1</code>	

Slurm example

```
#!/bin/bash ← Your shell is bash  
  
#SBATCH --partition=amilan ← Partition is amilan (CPU)  
#SBATCH --job-name=example-job ← Job name when it gets queued  
#SBATCH --output=example-job.%j.out ← Output file name  
#SBATCH --time=01:00:00 ← How long my job runs?  
#SBATCH --qos=normal  
#SBATCH --nodes=1  
#SBATCH --ntasks=4  
#SBATCH --mail-type=ALL  
#SBATCH --mail-user=youridentikey@colorado.edu
```

```
module purge  
module load anaconda  
conda activate custom-env
```

```
pythonmyscript.py
```

Slurm example

```
#!/bin/bash

#SBATCH --partition=amilan
#SBATCH --job-name=example-job
#SBATCH --output=example-job.%j.out
#SBATCH --time=01:00:00
#SBATCH --qos=normal
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --mail-type=ALL
#SBATCH --mail-user=youridentikey@colorado.edu

module purge
module load anaconda
conda activate custom-env

python myscript.py
```

Quality of service

How many nodes I want to run on?

Slurm example

```
#!/bin/bash

#SBATCH --partition=amilan
#SBATCH --job-name=example-job
#SBATCH --output=example-job.%j.out
#SBATCH --time=01:00:00
#SBATCH --qos=normal
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --mail-type=ALL
#SBATCH --mail-user=youridentikey@colorado.edu
```

```
module purge
module load anaconda
conda activate custom-env
```

```
python myscript.py
```

Total cores used

Email when job begins, ends and fails

Email address

Slurm example

```
#!/bin/bash

#SBATCH --partition=amilan
#SBATCH --job-name=example-job
#SBATCH --output=example-job.%j.out
#SBATCH --time=01:00:00
#SBATCH --qos=normal
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --mail-type=ALL
#SBATCH --mail-user=youridentikey@colorado.edu
```



Slurm cheatsheet (1)

Slurm script command	Description
<code>#!/bin/bash</code>	Sets the shell that the job will be executed on the compute node
<code>#SBATCH --ntasks=1</code> <code>#SBATCH --n1</code>	Requests for 1 processors on task, usually 1 cpu as 1 cpu per task is default.
<code>#SBATCH --time=0-05:00</code> <code>#SBATCH -t 0-05:00</code>	Sets the maximum runtime of 5 hours for your job
<code>#SBATCH --mail-user= <email></code>	Sets the email address for sending notifications about your job state.
<code>#SBATCH --mail-type=BEGIN</code> <code>#SBATCH --mail-type=END</code> <code>#SBATCH --mail-type=FAIL</code> <code>#SBATCH --mail-type=REQUEUE</code> <code>#SBATCH --mail-type=ALL</code>	Sets the scheduling system to send you email when the job enters the following states: BEGIN,END,FAIL,REQUEUE,ALL
<code>#SBATCH --job-name=my-named-job</code>	Sets the Jobs name

Slurm cheatsheet(2)

Slurm script command	Description
#SBATCH --ntasks=X	Requests for X tasks. When cpus-per-task=1 (and this is the default) this requests X cores. When not otherwise constraint these CPUs may be running on any node
#SBATCH --nodes=X	Request that a minimum of X nodes be allocated to this job
#SBATCH --nodes=X-Y	Request that a minimum of X nodes and a maximum of Y nodes be allocated to this job
#SBATCH --cpus-per-task=X	Request that a minimum of X CPUs per task be allocated to this job
#SBATCH --tasks-per-node=X	Requests minimum of X task be allocated per node

Slurm cheatsheet(3)

Slurm script commands	Description of effects
<code>#SBATCH --ntasks=1</code> <code>#SBATCH --cpus-per-task=1</code>	Requests 1 CPU (Serial) cpus-per-task is set to 1 by default and may be omitted.
<code>#SBATCH --cpus-per-task=X</code> <code>#SBATCH --ntasks=1</code> <code>#SBATCH --nodes=1</code>	Requests for X CPUs in 1 task on 1 node (OpenMP) Both ntasks and nodes are set to 1 by default and may be omitted
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --tasks-per-node=X</code> <code>#SBATCH --cpus-per-task=1</code>	Requests for X CPUs and tasks on 1 node cpus-per-task is set to 1 by default and may be omitted.
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --nodes=1</code> <code>#SBATCH --cpus-per-task=1</code>	Requests for X CPUs and tasks on 1 node cpus-per-task is set to 1 by default and may be omitted.

Get information about jobs

```
[kfotso@xsede.org@login-ci1 ~]$ squeue -l --me
```

JOBID	PARTITION	NAME	USER	STATE	TIME	TIME_LIMI	NODES	NODELIST(REASON)
2158225	acompile	acompile	kfotso@x	RUNNING	0:16	3:00	1	c3cpu-c11-u21-2

Monitor resources

```
[kfotso@xsede.org@login-ci1 ~]$ module load slurmtools
[kfotso@xsede.org@login-ci1 ~]$ jobstats $USER 2
job stats for user kfotso@xsede.org over past 2 days
jobid      jobname  partition   qos       account    cpus state start-date-time elapsed    wait
-----
2064187    sinterac atesting_+ testing    amc-gener+  48  TIMEOUT 2023-06-20T23:32:52 01:00:04  0 hrs
2071952    vep_loft aa100        normal    amc-gener+  64  COMPLETE 2023-06-21T13:47:55 00:26:42  3 hrs
```

- Allows to get information about a past jobs

```
[kfotso@xsede.org@login-ci1 ~]$ seff 1451164
Job ID: 1451164
Cluster: alpine
User/Group: kfotso@xsede.org/kfotsopgrp@xsede.org
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 48
CPU Utilized: 26-03:21:39
CPU Efficiency: 94.06% of 27-19:00:48 core-walltime
Job Wall-clock time: 13:53:46
Memory Utilized: 412.77 GB
Memory Efficiency: 41.28% of 999.98 GB
```

- To get more computational information about the job efficiency

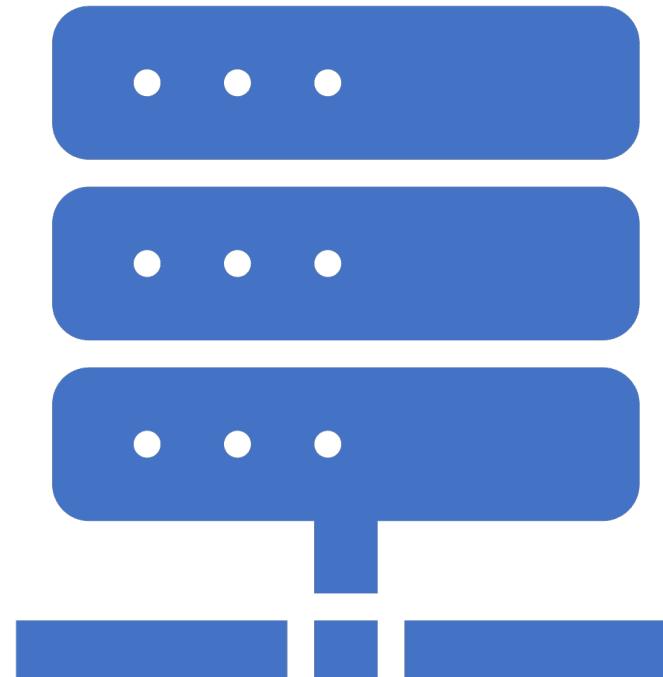
Slurm Quality of service (qos)

- Used to modify or constrain characteristics that a job can have.
- **--qos=normal** corresponds to a walltime of 24 hours and is the default.
- **--qos=long** corresponds to a walltime of up to 7 days
- **--qos=mem** corresponds to high memory jobs only (up to 1TB)



Fairshare overview

- Difference between the portion of computing resource that has been promised and the amount of resources that has been consumed.
- Level fairshare of 1 indicates average priority compared to other users in that account (amc-general)
- **module load slurmtools; levelfs \$USER**



Job priority calculation formula

```
Job_priority =  
    site_factor +  
    (PriorityWeightAge) * (age_factor) +  
    (PriorityWeightAssoc) * (assoc_factor) +  
    (PriorityWeightFairshare) * (fair-share_factor) +  
    (PriorityWeightJobSize) * (job_size_factor) +  
    (PriorityWeightPartition) * (partition_factor) +  
    (PriorityWeightQOS) * (QOS_factor) +  
    SUM(TRES_weight_cpu * TRES_factor_cpu,  
        TRES_weight_<type> * TRES_factor_<type>,  
        ...)  
    - nice_factor
```

Check fairshare

```
Host: login-ci1.rc.int.colorado.edu
[kfotso@xsede.org@login-ci1 ~]$ levelfs $USER
LevelFS for user kfotso@xsede.org and institution amc:
Account          LevelFS_User      LevelFS_Inst
-----
amc-general      0.194275        4.750220
[kfotso@xsede.org@login-ci1 ~]$ █
```

- 0.19 means that my priority will be low
- On the other hand 4.75 means that priority for the institution is high

Service Units (SU)

- It is the number of core hours used.

```
[kfotso@xsede.org@login-ci1 ~]$ suuser $USER 10
SU used by user kfotso@xsede.org in the last 10 days:
Cluster|Account|Login|Proper Name|TRES Name|Used|
alpine|amc-general|kfotso@xsede.org|Kevin Fotso|billing|8393|
```

- suacct to get the number of core hours used by institution

```
Host: login-ci1.rc.int.colorado.edu
[kfotso@xsede.org@login-ci1 ~]$ suacct amc-general 180
SU used by account (allocation) amc-general in the last 180 days:
Cluster|Account|Login|Proper Name|TRES Name|Used
alpine|amc-general|||billing|1806360
alpine| amc-general|acozart@xsede.org|Abigail Cozart|billing|573
alpine| amc-general|agillen@xsede.org|Austin Gillen|billing|40320
alpine| amc-general|agray@xsede.org|Alyx Gray|billing|22
```

Best practices on Alpine

- /home or /tmp have a very small size. Please always include the following in your slurm script.

```
export TMP=/scratch/alpine/$USER
```

```
export TEMP=/scratch/alpine/$USER
```

```
export TMPDIR=/scratch/alpine/$USER
```

```
export TEMPDIR=/scratch/alpine/$USER
```

- In general, always make sure to check your cache in \$HOME to make sure you are not running out of space.

Brief mention of parallelism models

- OpenMP (multithreading).
- MPI (Parallel distribution between multiple nodes)
- GNU parallel (embarrassingly parallel)
- Job arrays (max 1000 on Alpine)
- Load balancing (developed by CU Boulder)
- GPU computing



Slurm example to see the number of cores(1)

- Here, we are requesting 4 cores for a 2 minutes



```
[kfotso@login-ci2 ~]$ acompile --ntasks=4 --time=00:02:00
acompile: submitting job... salloc --nodes=1 --partition=acompile --ntasks=4 --time=00:02:00 --qos=compile --job-name=acompile --bell
--oversubscribe srun --pty /bin/bash
salloc: Granted job allocation 2998293
salloc: Nodes c3cpu-a5-u32-1 are ready for job
```



- We are getting node c3cpu-a5-u32-1 and our jobID is 2998293

Slurm example to see the number of cores(2)

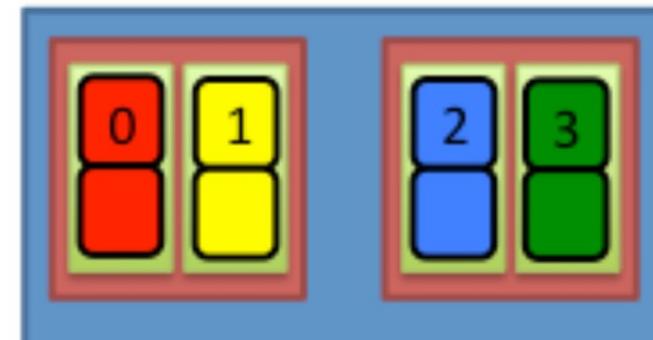
- We want load compiler gcc because openmpi depends on it



```
[kfotso@xsede.org@c3cpu-a5-u32-1 ~]$ module load gcc  
[kfotso@xsede.org@c3cpu-a5-u32-1 ~]$ module load openmpi
```



- We want to use openmpi because we want to demonstrate parallelism on the 4 cores we have reserved



Slurm example to see the number of cores(3)

- hostname is just a program that prints the name of the node I am working with



```
[kfotso@xsede.org@c3cpu-a5-u32-1 ~]$ mpirun -n 4 hostname  
c3cpu-a5-u32-1.rc.int.colorado.edu  
c3cpu-a5-u32-1.rc.int.colorado.edu  
c3cpu-a5-u32-1.rc.int.colorado.edu  
c3cpu-a5-u32-1.rc.int.colorado.edu
```

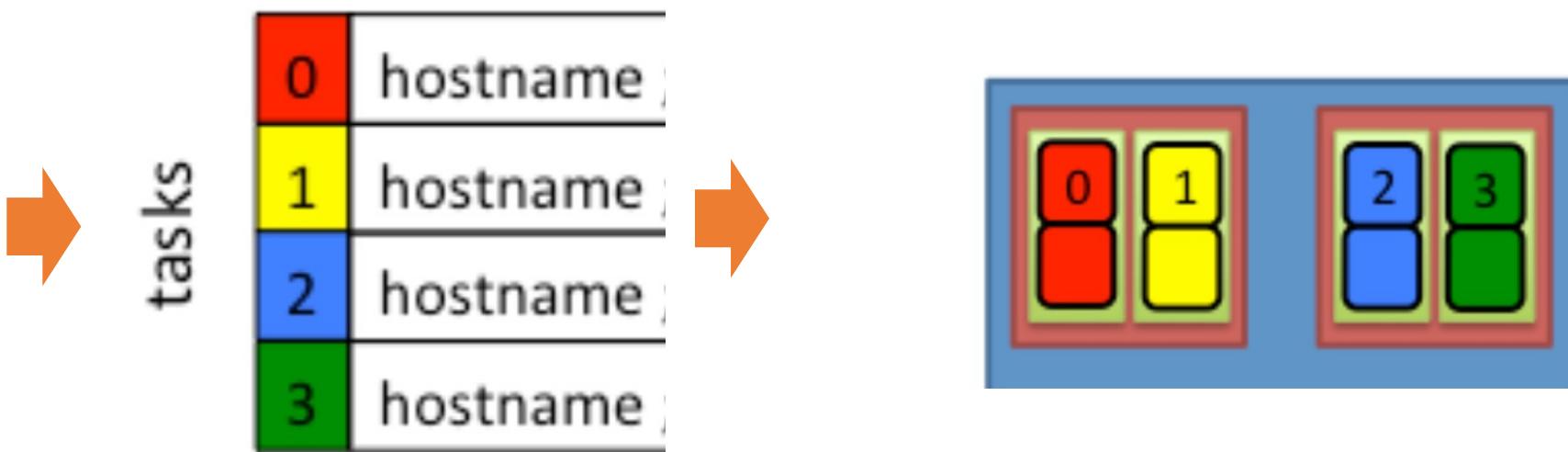


- With mpirun, we schedule 4 tasks on the 4 cores, thus the name of the name of the node gets printed 4 times !!!

Slurm example to see the number of cores(4)

Recap of what we did

`mpirun -n 4 hostname`



Slurm example to see the number of cores(5)

- Congratulations!!! You just wrote your first HPC parallel code program!



Build the corresponding slurm script (1)

- Create the empty slurm script and modify it with your editor of choice (nano, vim, emac, Ondemand etc ...)



```
[kfotso@xsede.org@login-ci2 openmpi_hostname]$ touch hello_mpi.sh  
[kfotso@xsede.org@login-ci2 openmpi_hostname]$ █
```

Build the corresponding slurm script (2)

```
#!/bin/bash

#SBATCH --partition=amilan
#SBATCH --job-name=first_mpirun
#SBATCH --output=first_mpirun-job.%j.out
#SBATCH --error=first_mpirun-job.%j.err
#SBATCH --account=amc-general
#SBATCH --qos=normal
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --mail-type=ALL
#SBATCH --mail-user=kfotsotagne@unm.edu
#SBATCH --time=00:00:01

module load gcc
module load openmpi

mpirun -n $SLURM_NTASKS hostname
```

Account name is
amc-general

Job runs for 1
second!!

Build the corresponding slurm script (3)

```
#!/bin/bash

#SBATCH --partition=amilan
#SBATCH --job-name=first_mpirun
#SBATCH --output=first_mpirun-job.%j.out
#SBATCH --error=first_mpirun-job.%j.err
#SBATCH --account=amc-general
#SBATCH --qos=normal
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --mail-type=ALL
#SBATCH --mail-user=kfotsotagne@unm.edu
#SBATCH --time=00:00:01

module load gcc
module load openmpi

mpirun -n $SLURM_NTASKS hostname
```

\$SLURM_NTASKS is the slurm env
variable for the 4 cores

Build the corresponding slurm script (4)

We use sbatch to submit the script



```
[kfotso@xsede.org@login-ci2 openmpi_hostname]$ sbatch hello_mpi.sh  
Submitted batch job 2998325
```

Build the corresponding slurm script (5)

The .out file shows the name of the nodes
4X. .err file is empty so no error.



```
[kfotso@xsede.org@login-ci2 openmpi_hostname]$ ls -latr
total 184
drwxrws---. 44 kfotso@xsede.org kfotsogrp@xsede.org 1672 Sep 15 01:45 ..
-rw-r--r--. 1 kfotso@xsede.org kfotsogrp@xsede.org 400 Sep 15 01:52 hello_mpi.sh
-rw-r--r--. 1 kfotso@xsede.org kfotsogrp@xsede.org 0 Sep 15 01:52 first_mpirun-job.2998325.err
drwxr-sr-x. 2 kfotso@xsede.org kfotsogrp@xsede.org 122 Sep 15 01:52 .
-rw-r--r--. 1 kfotso@xsede.org kfotsogrp@xsede.org 140 Sep 15 01:52 first_mpirun-job.2998325.out
[kfotso@xsede.org@login-ci2 openmpi_hostname]$ cat first_mpirun-job.2998325.out
c3cpu-c15-u1-2.rc.int.colorado.edu
c3cpu-c15-u1-2.rc.int.colorado.edu
c3cpu-c15-u1-2.rc.int.colorado.edu
c3cpu-c15-u1-2.rc.int.colorado.edu
[kfotso@xsede.org@login-ci2 openmpi_hostname]$
```

Ondemand MATLAB

RC will conduct its monthly planned maintenance on Wednesday, September 6. Please visit curc.statuspage.io for details.

Notice: Users will be limited to a maximum of 8 cores per Core Desktop session through mid-September due to ongoing maintenance.

[Home](#) / [My Interactive Sessions](#) / [MATLAB \(Presets\)](#)

Interactive Apps

- Desktops
- Core Desktop (Presets)
- GUIs
- MATLAB (Presets)

Servers

- Jupyter Session (Custom)
- Jupyter Session (Presets)
- RStudio Server (Custom)
- RStudio Server (Presets)
- VS Code-Server (Custom)
- VS Code-Server (Presets)

MATLAB (Presets)

This app will launch a MATLAB GUI on a CURC node. You will be able to interact with MATLAB through a VNC session. GPU based options are not meant for computationally intensive workflows. Additionally, please keep in mind that these GPU based options are a shared resource amongst all users. Thus, significant computation by one user can affect other users of this service.

MATLAB version

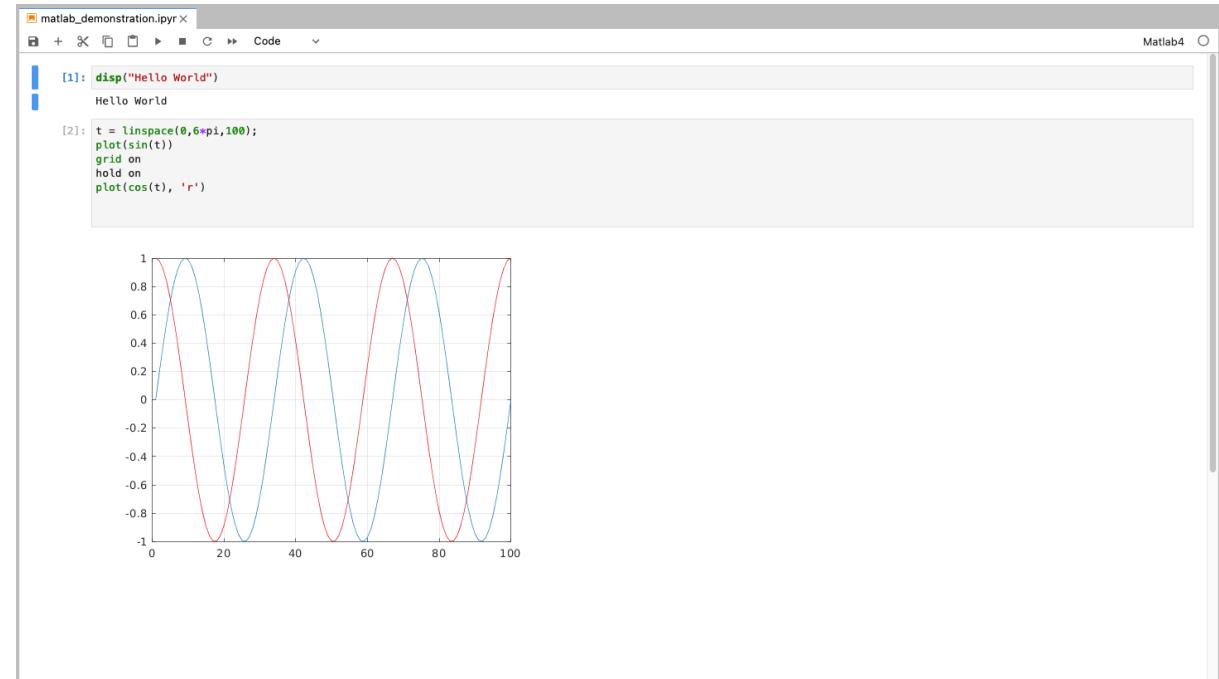
R2021b

Configuration

- ✓ 2 cores, 1 hour, K80 GPU
- 2 cores, 1 hour, RTX8000 GPU
- 2 cores, 12 hours, K80 GPU
- 2 cores, 12 hours, RTX8000 GPU
- 4 cores, 4 hours, K80 GPU
- 4 cores, 4 hours, RTX8000 GPU

Ondemand Jupyterlab with MATLAB kernel

- Instructions here:
<https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/MATLAB-kernel-on-Jupyterlab.md>
- Possibility to use MATLAB with ssh (if registered)
- MATLAB Ondemand coming soon.



Rstudio with Ondemand

The screenshot shows the XSEDE OnDemand interface for launching an RStudio Server (Custom) session. The top navigation bar includes links for Files, Jobs, Clusters, Interactive Apps (selected), My Interactive Sessions, Help, and Logout. A dropdown menu under 'Interactive Apps' lists several options: JupyterHub 1 (Presets), JupyterHub 2 (Custom), RStudio Server (Custom), RStudio Server (Presets), JupyterHub 1 (Presets), JupyterHub 2 (Custom), RStudio Server (Custom) (selected), and RStudio Server (Presets). The main content area displays the 'RStudio Server (Custom)' application details, stating it will launch RStudio Server, an IDE for R on Alpine. It includes a note about reading the RStudio section of the CURC documentation and running custom interactive applications. Configuration fields include RStudio Version (Rstudio 2023.03.0, R 4.2.2), Cluster (Alpine), Account (amc-general), Partition (ahub), Number of cores (4), Memory [GiB] (4), and QoS Name (empty field).

Interactive Apps / My Interactive Sessions

Help Logged in as kfots@xsede.org Logout

Servers

- JupyterHub 1 (Presets)
- JupyterHub 2 (Custom)
- RStudio Server (Custom)
- RStudio Server (Presets)
- JupyterHub 1 (Presets)
- JupyterHub 2 (Custom)
- RStudio Server (Custom) **Selected**
- RStudio Server (Presets)

RStudio Server (Custom)

This app will launch [RStudio Server](#), an IDE for R on Alpine.

Before utilizing this application, please see the [RStudio section of the CURC documentation](#). This documentation includes important information regarding quitting an RStudio session. For more information on possible settings for this application, see [Running Custom Interactive applications](#) in our documentation.

RStudio Version

Rstudio 2023.03.0, R 4.2.2

Cluster

Alpine

Account

amc-general

Partition

ahub

Number of cores

4

Memory [GiB]

4

QoS Name

VS Code on Ondemand-rmacc

Home / My Interactive Sessions / VS Code-Server (Custom)

Interactive Apps

- Desktops
- Core Desktop (Presets)
- GUIs
- MATLAB (Presets)

Servers

- Jupyter Session (Custom)
- Jupyter Session (Presets)
- RStudio Server (Custom)
- RStudio Server (Presets)
- VS Code-Server (Custom)**
- VS Code-Server (Presets)

VS Code-Server (Custom)

This app will launch a [VS Code](#) server using [Code-Server](#). For more information on possible settings for this application, see [Running Custom Interactive applications](#) in our documentation. Additionally, for more information on installing VS Code extensions, please see our [Installing VS Code-Server Extensions](#) section of the documentation.

Cluster

Alpine

Code-Server version

✓ 4.16.1
4.14.1

Account

amc-general

Partition

ahub

QoS Name

interactive

Time

4

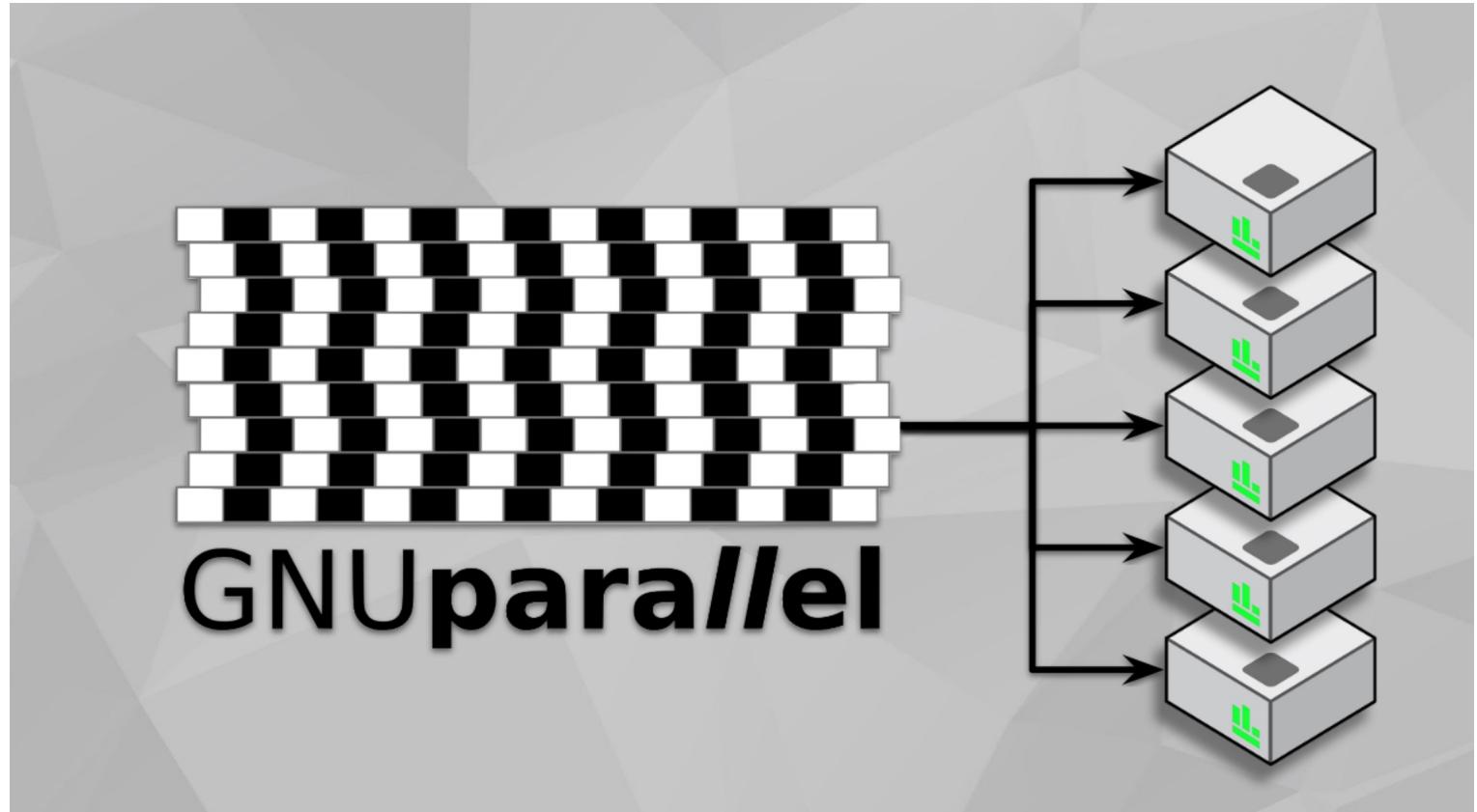
Number of cores

1

Launch

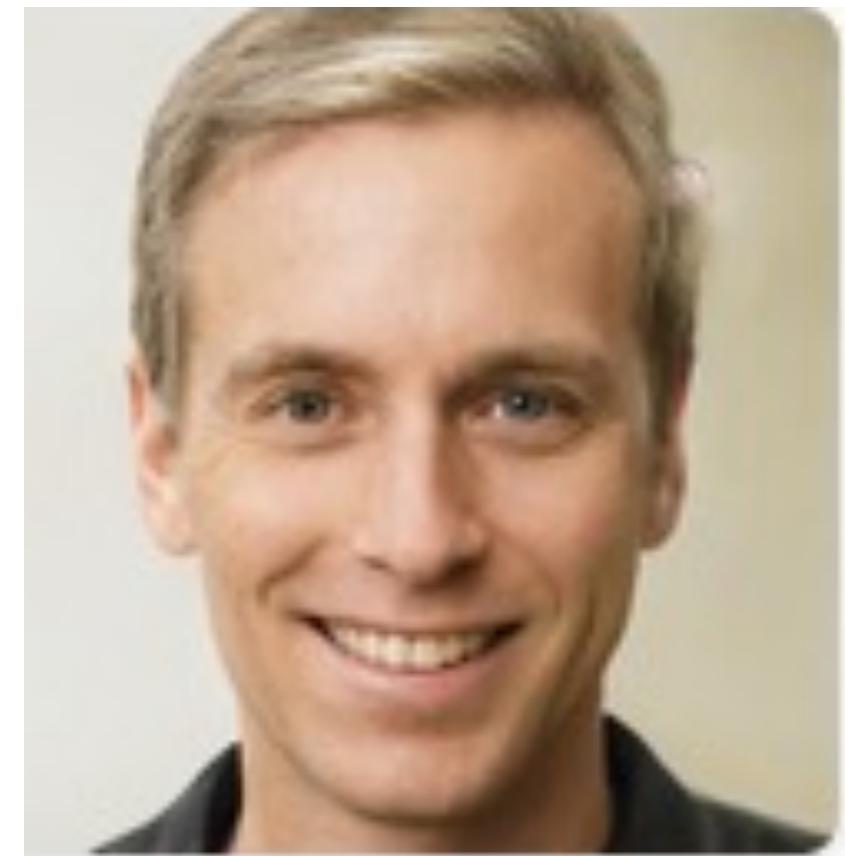
* The VS Code-Server (Custom) session data for this session can be accessed under the [data root directory](#).

III- Part 3) Intro to Gnu parallel



What is GNU parallel? (1)

- It is a shell tool made for executing jobs in an embarrassingly parallel way in 1 or more nodes
- Was created by Ole Tange from the University of Copenhagen



Why GNU parallel?

- I have a code and I want to run multiple instances of the same code which instance running on very similar files.
- There is little synchronization required. (The files do not need to know about one another).

Why GNU parallel?

- I have a code and I want to run multiple instances of the same code which instance running on very similar files.
- There is little synchronization required. (The files do not need to know about one another).

Program to
process
multiple .fa
files

Why GNU parallel?

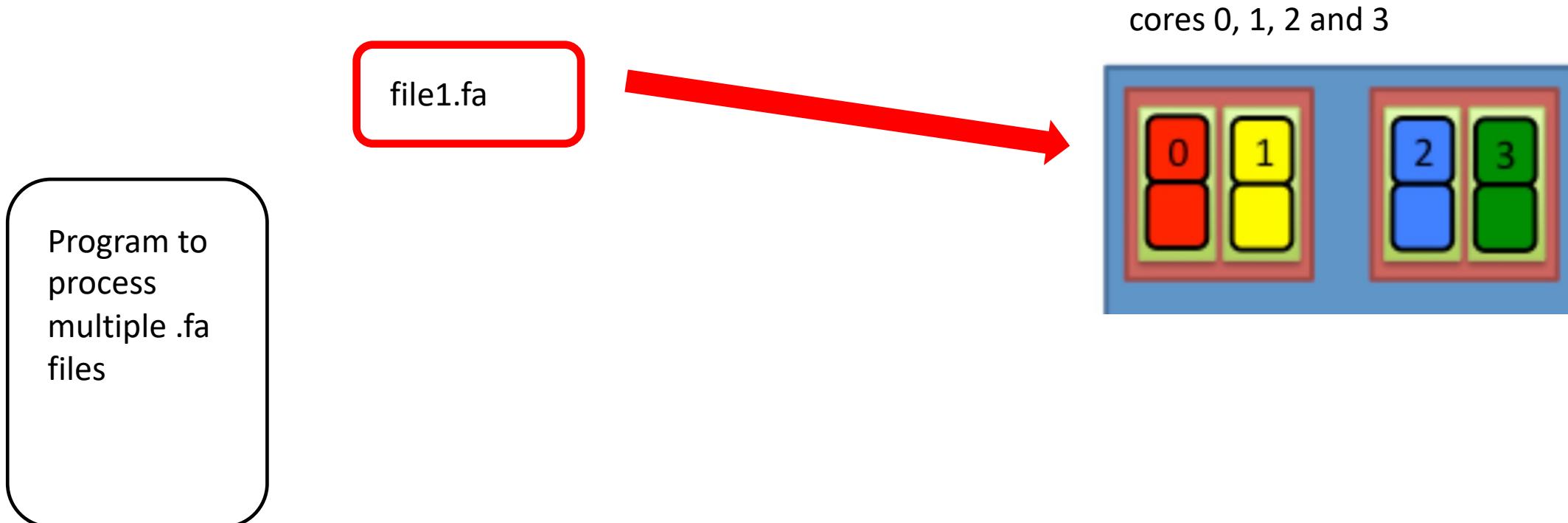
- I have a code and I want to run multiple instances of the same code which instance running on very similar files.
- There is little synchronization required. (The files do not need to know about one another).

file1.fa

Program to
process
multiple .fa
files

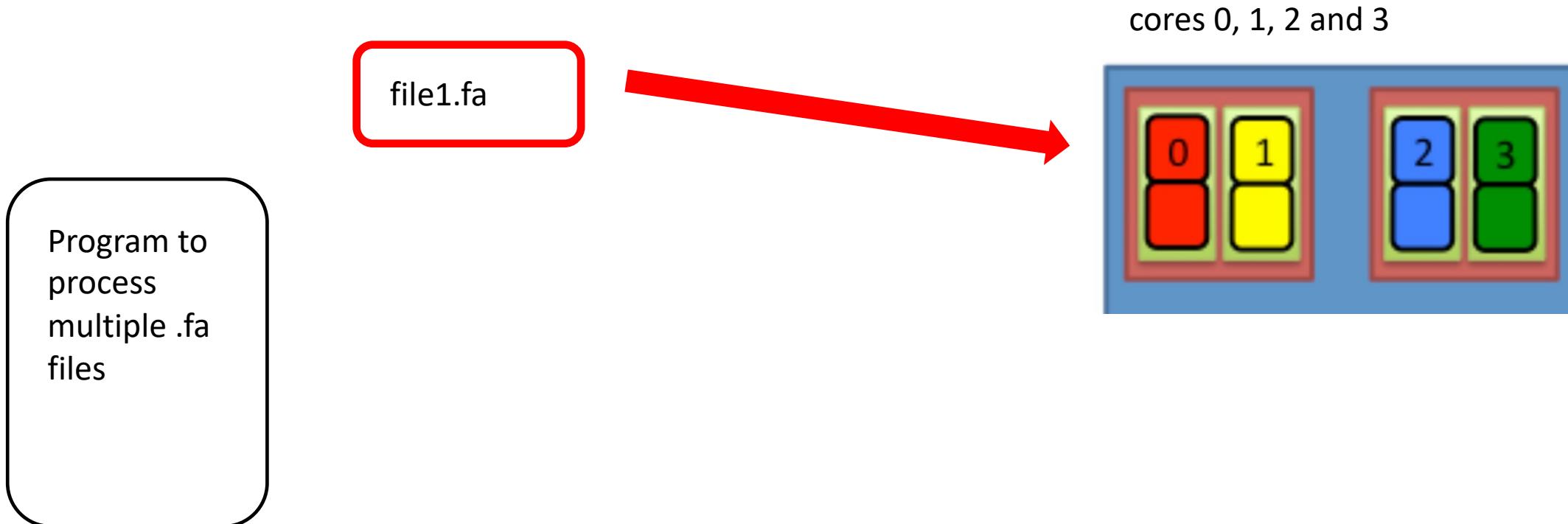
Why GNU parallel?

- I have a code and I want to run multiple instances of the same code which instance running on very similar files.
- There is little synchronization required. (The files do not need to know about one another).



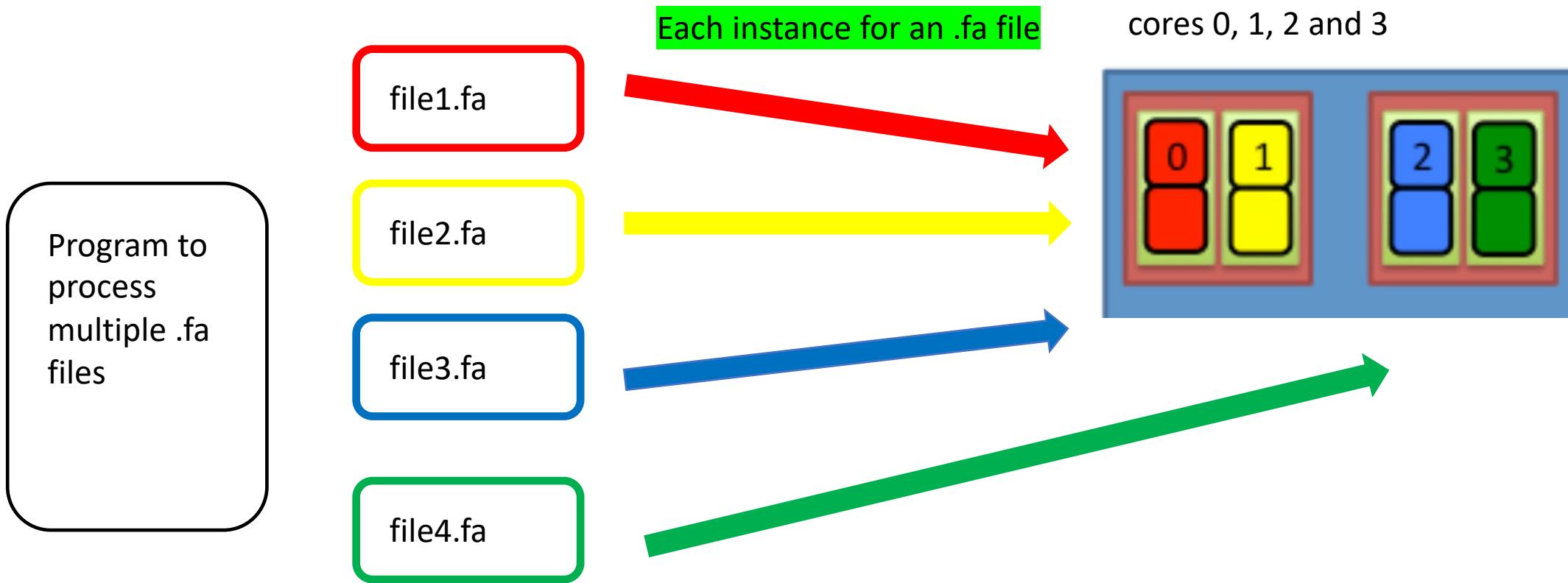
Why GNU parallel?

- I have a code and I want to run multiple instances of the same code which instance running on very similar files.
- There is little synchronization required. (The files do not need to know about one another).



Why GNU parallel?

- I have a code and I want to run multiple instances of the same code which instance running on very similar files.
- There is little synchronization required. (The files do not need to know about one another).



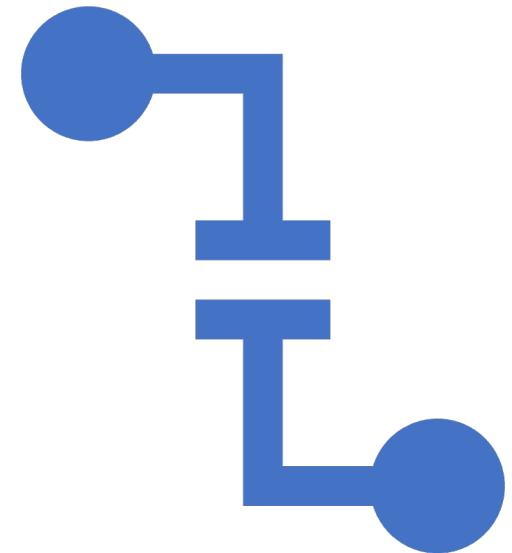


Strength of GNU parallel

- Easily scales to very large number of tasks.
- Easily scales to multiple nodes.
- Efficient use of Slurm resources.

Cons of gnu parallel

- Does not easily balance nodes across multiple nodes
- Careful optimization of input/output files is required
- Scaling up requires careful consideration of I/O performance.
- Bash scripting familiarity recommended



Basic example

```
[kfotso@xsede.org@c3cpu-c11-u17-2 openmpi_hostname]$ parallel -j 4 echo :::: A B C D  
Academic tradition requires you to cite works you base your article on.  
If you use programs that use GNU Parallel to process data for an article in a  
scientific publication, please cite:
```

Tange, O. (2021, March 22). GNU Parallel 20210322 ('2002-01-06').
Zenodo. <https://doi.org/10.5281/zenodo.4628277>

This helps funding further development; AND IT WON'T COST YOU A CENT.
If you pay 10000 EUR you should feel free to use GNU Parallel without citing.

More about funding GNU Parallel and the citation notice:
https://www.gnu.org/software/parallel/parallel_design.html#Citation-notice

To silence this citation notice: run 'parallel --citation' once.

B
A
C
D

Number of instances
For arguments

Additional param

Joblog for
debug



```
[kfotso@xsede.org@c3cpu-c11-u17-2 openmpi_hostname]$ parallel --joblog job.log -j 4 echo :::: A B C D
Academic tradition requires you to cite works you base your article on.
If you use programs that use GNU Parallel to process data for an article in a
scientific publication, please cite:
```

Tange, O. (2021, March 22). GNU Parallel 20210322 ('2002-01-06').
Zenodo. <https://doi.org/10.5281/zenodo.4628277>

This helps funding further development; AND IT WON'T COST YOU A CENT.
If you pay 10000 EUR you should feel free to use GNU Parallel without citing.

More about funding GNU Parallel and the citation notice:
https://www.gnu.org/software/parallel/parallel_design.html#Citation-notice

To silence this citation notice: run 'parallel --citation' once.

A
C
B
D

Navigation icons: back, forward, search, etc.

Joblog