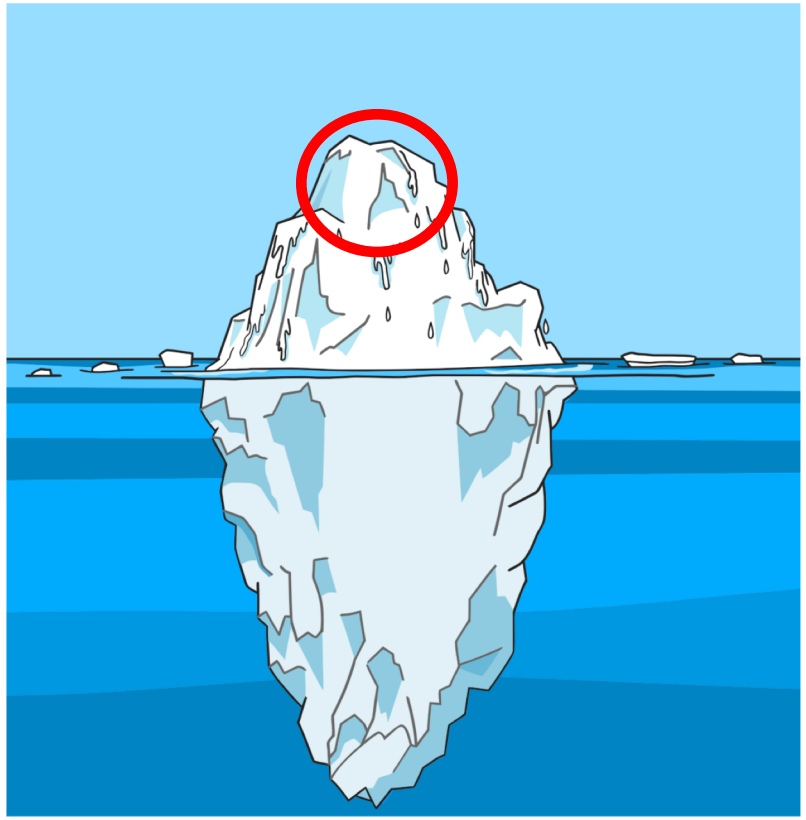
High Performance Computing (HPC) Crash Course

Brandon Reyes

For help email: rc-help@colorado.edu

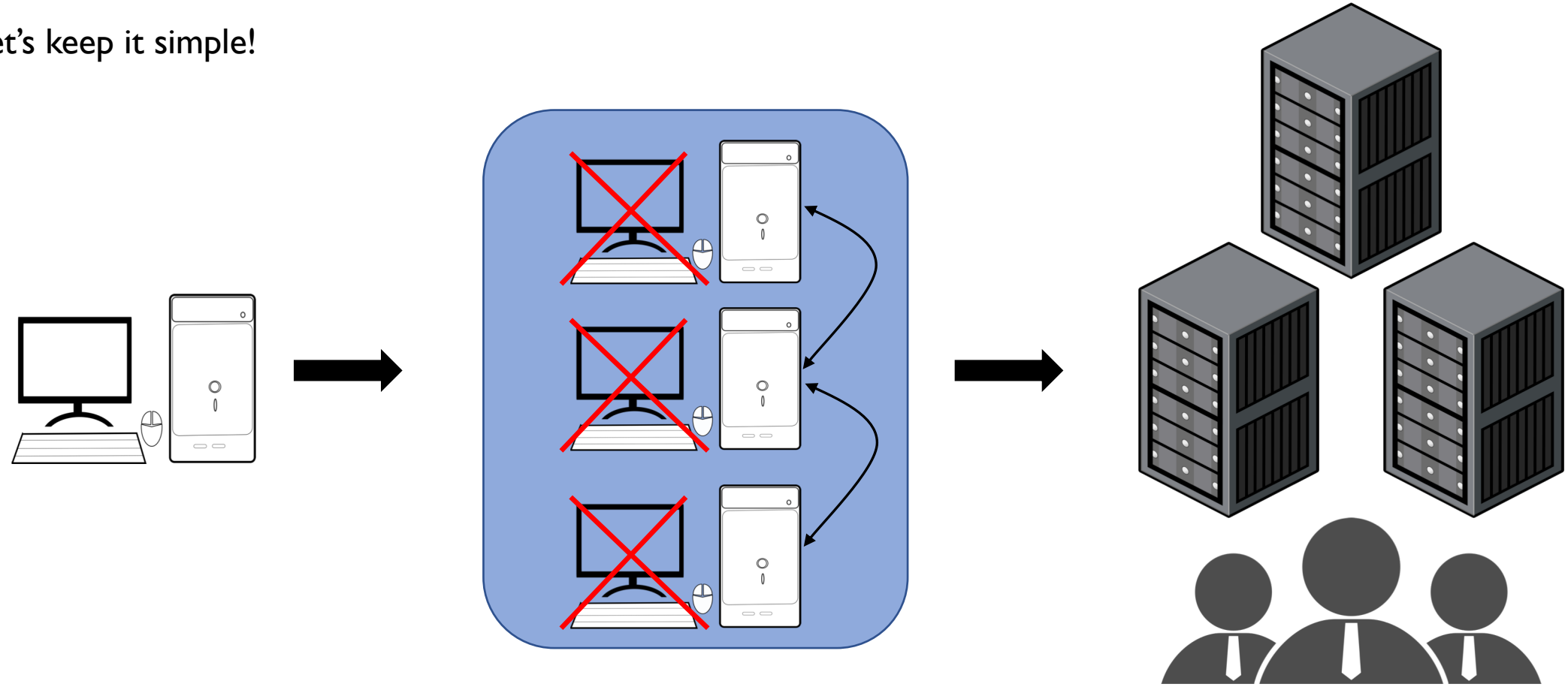
WHAT DO WE WANT TO ACHIEVE?

- **Goal:** Obtain an overview of a common workflow on an HPC system
- **Audience:** Individuals who are new to HPC systems



WHAT IS AN HPC SYSTEM?

Let's keep it simple!



rc-help@colorado.edu

SOME PERKS OF AN HPC SYSTEM

Perks	HPC System
Common software and programming languages are easily available (modules)	✓
Large storage, RAM, and number of CPUs or GPUs (✚ multiple nodes)	✓
Independent tasks or parallel applications can be completed significantly faster	✓

ALPINE STATS

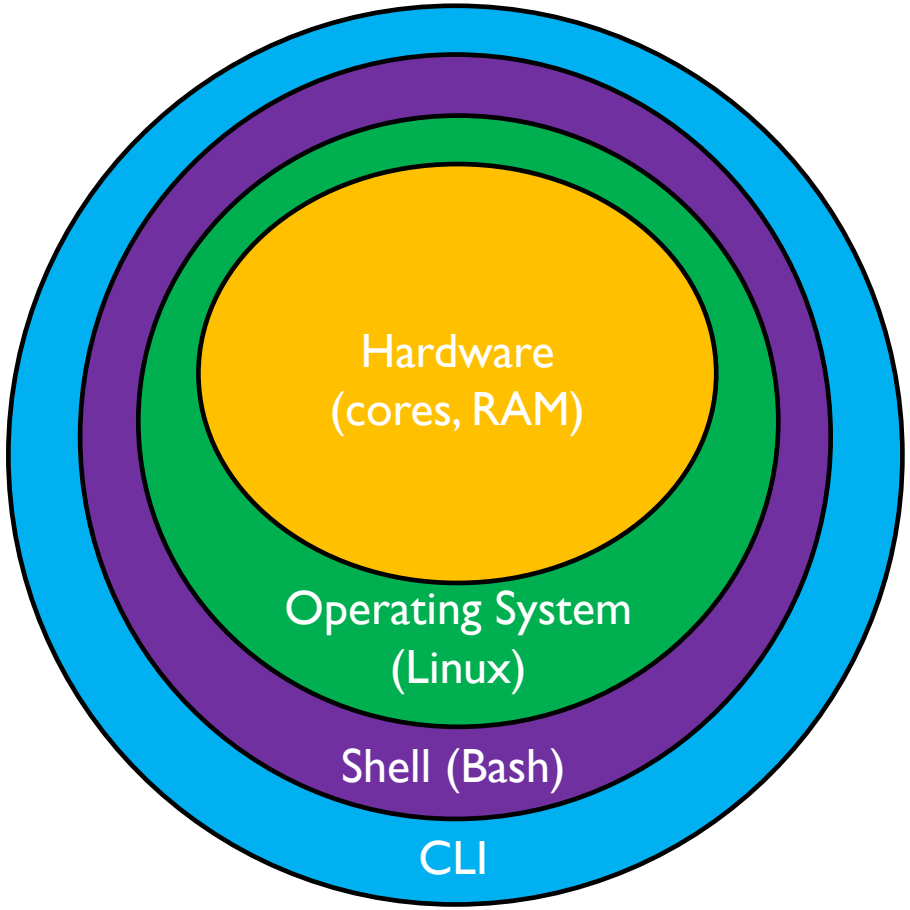
Partitions available on Alpine:

Partition	Description	# of nodes	cores/node	RAM/core (GB)
amilan	AMD Milan (default)	184	64	3.74
ami100	GPU-enabled (3x AMD MI100)	8	64	3.74
aa100	GPU-enabled (3x NVIDIA A100)	8	64	3.74
amem[1]	High-memory	4	48	21.486
csu	Nodes contributed by CSU	77	32 or 48	3.74
amc	Nodes contributed by AMC	20	64	3.74

~ 239 GB per node!

COMMAND-LINE INTERFACE (CLI)

- Text-based interaction with operation system
- A simple HPC workflow will use the CLI to:
  - Access HPC
  - Work with folders (directories) and files
    - View or edit files (text editor)
  - Submit our code to the HPC system
  - Help us monitor the HPC jobs



WHAT DOES A CLI LOOK LIKE?

Real world CLI

```
root@abe5c26dbf5d:/home# ls
HOMEWORK        data_analysis.py
Presentations   hello_world.cpp
README.txt      slurm_script.sh
```

Documentation CLI

```
$ ls
```

ACCESSING AN HPC SYSTEM

- Create a Research Computing account
- Set up Duo 2-factor Authentication
- Open a terminal

```
$ ssh username@login.rc.colorado.edu
```

- ssh is a secure way to access a computer over a network

```
$ module load slurm/alpine
```

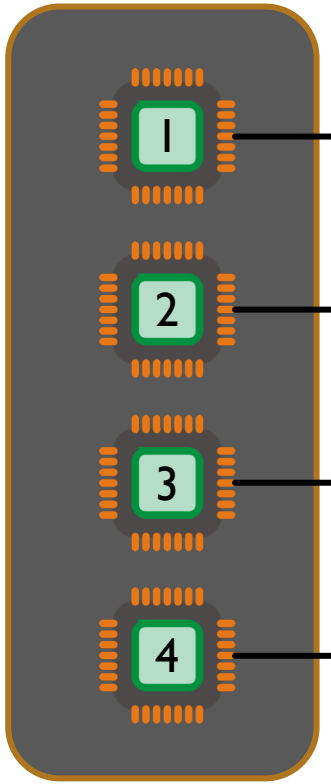
- Allows one to submit jobs to Alpine (more on this later)

PARALLEL APPLICATION

Parallel Application Pseudocode

```
1  # assign work to core
2  if core number is even:
3     print("Even")
4  else:
5     print("Odd")
```

Node



Output

Odd

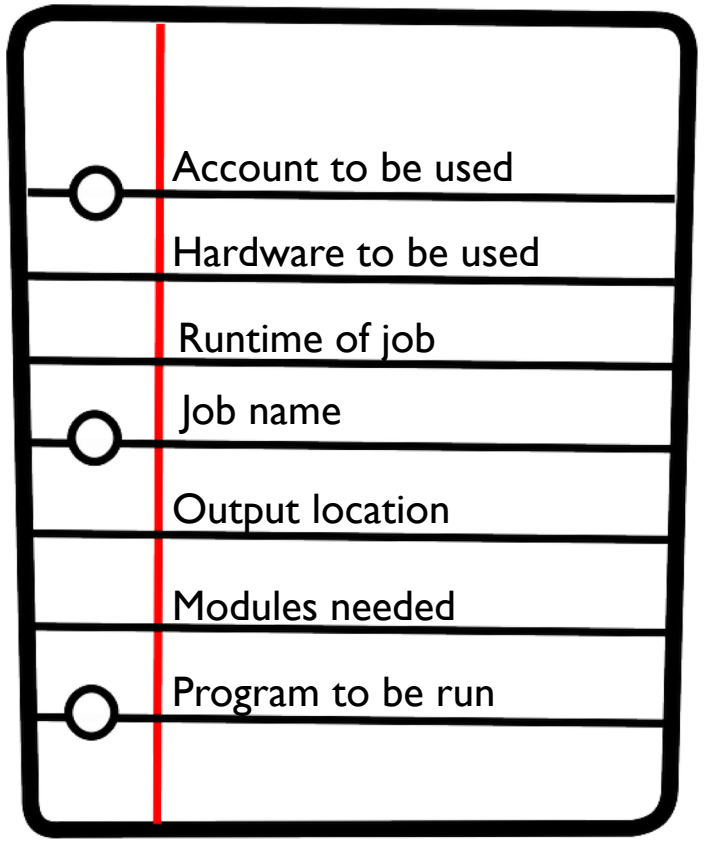
Even

Odd

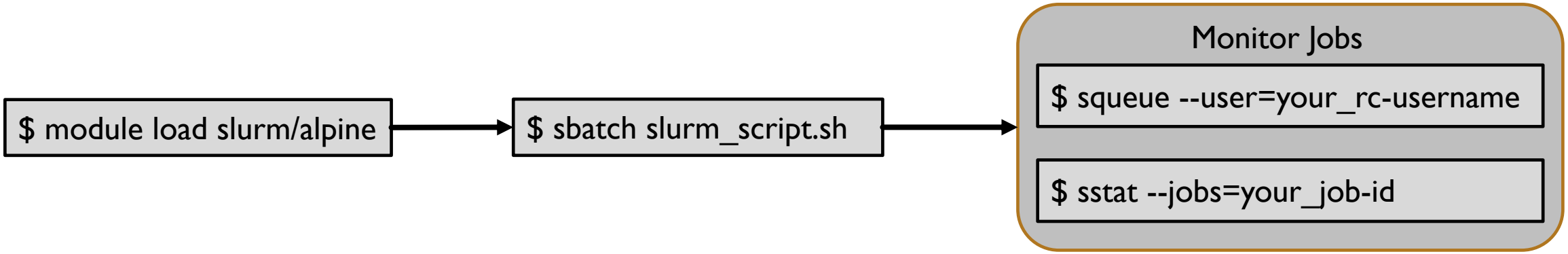
Even

SLURM JOB SCRIPTS

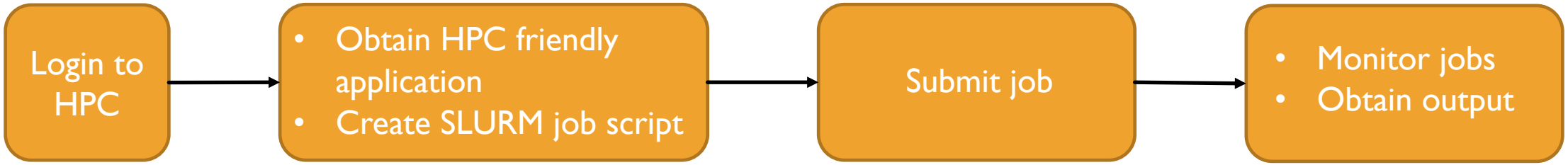
slurm_script.sh



SUBMITTING A SLURM SCRIPT

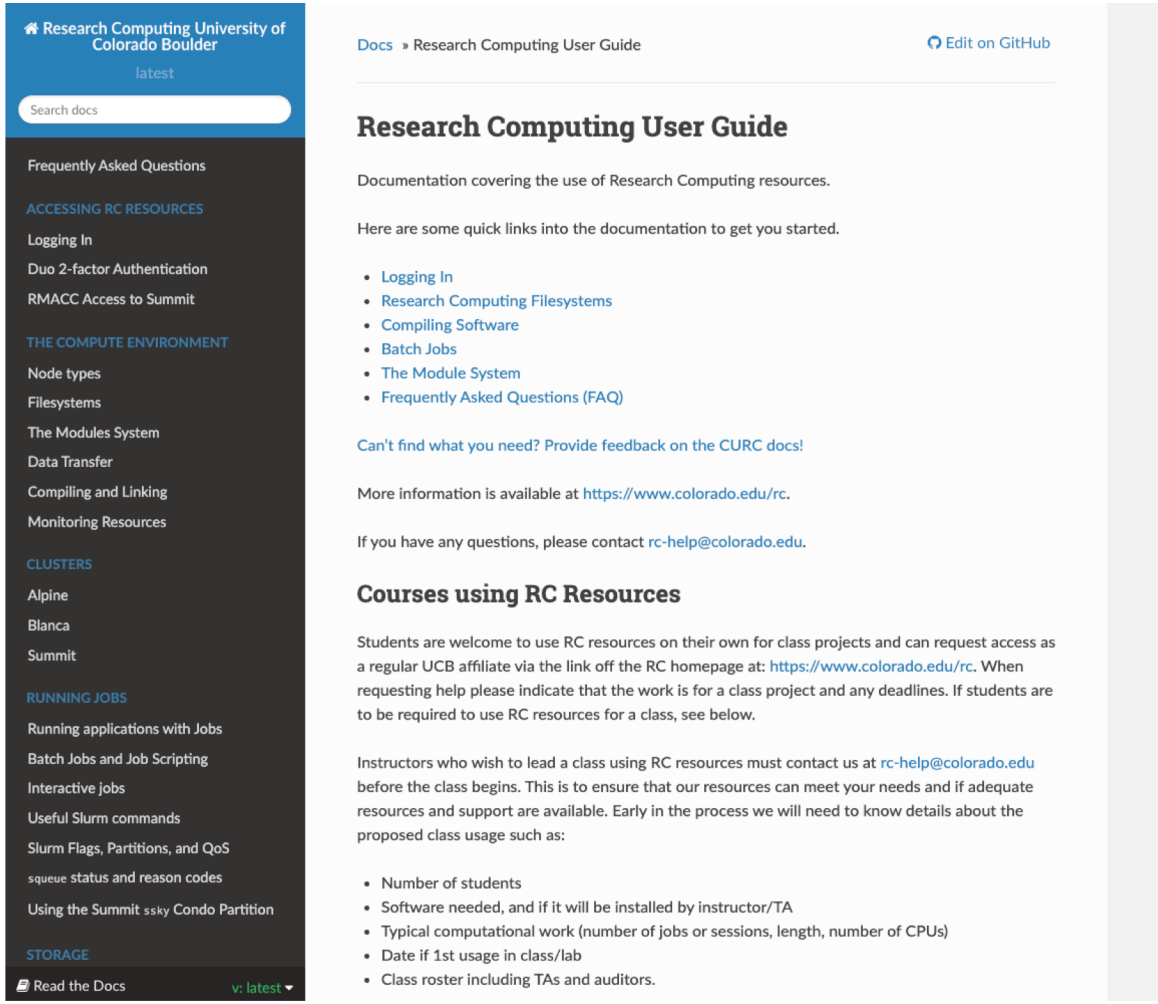


OVERVIEW OF HPC WORKFLOW



FURTHER HELP

- Documentation
  - curc.readthedocs.io
- rc-help@colorado.edu



Thank You!

WHEN SHOULD YOU USE AN HPC SYSTEM?

- Your calculation consumes more memory (RAM) than you have
- You need to run a large number of tasks that are independent
  - Your application can be run in parallel (more on this later)

WHAT DOES A CLI LOOK LIKE?

Real world CLI

```
root@abe5c26dbf5d:/home# ls
HOMEWORK        data_analysis.py
Presentations   hello_world.cpp
README.txt      slurm_script.sh
```

Documentation CLI

```
/home $ ls
HOMEWORK        data_analysis.py
Presentations   hello_world.cpp
README.txt      slurm_script.sh
```

Access to a CLI:

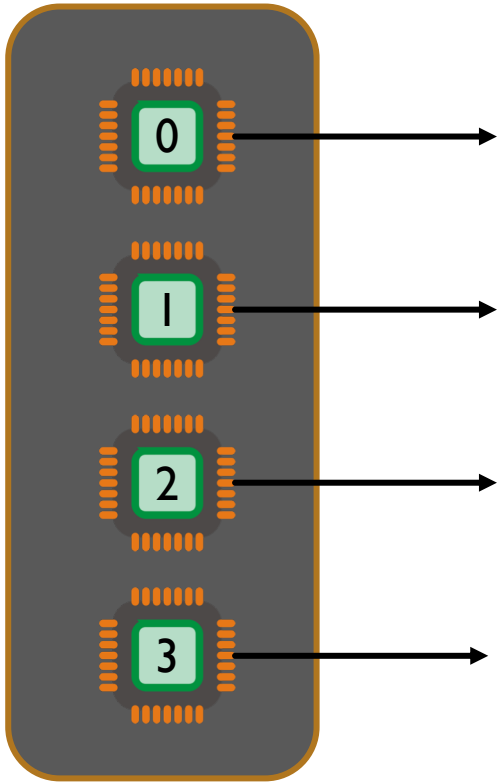
- macOS and Ubuntu (Linux)
  - Start up a terminal
- Windows
  - PuTTY (terminal emulator)

PARALLEL APPLICATION

Parallel Application Pseudocode

```
1  # get parallel library
2  Load_MPI(…)
3
4  # get core number
5  core = MPI_get_core_number(…)
6
7  # print core number
8  print("core number: " core)
```

Node



Output

core number: 0

core number : 1

core number : 2

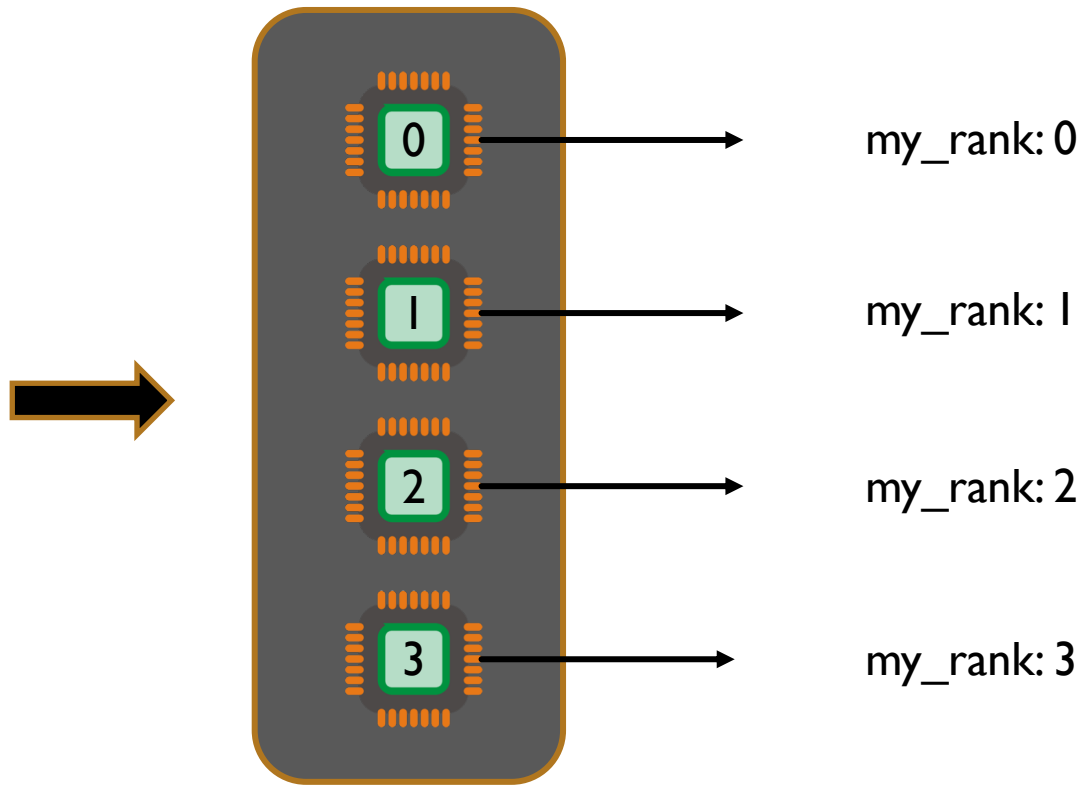
core number : 3

PARALLEL CONCEPT

Parallel Application

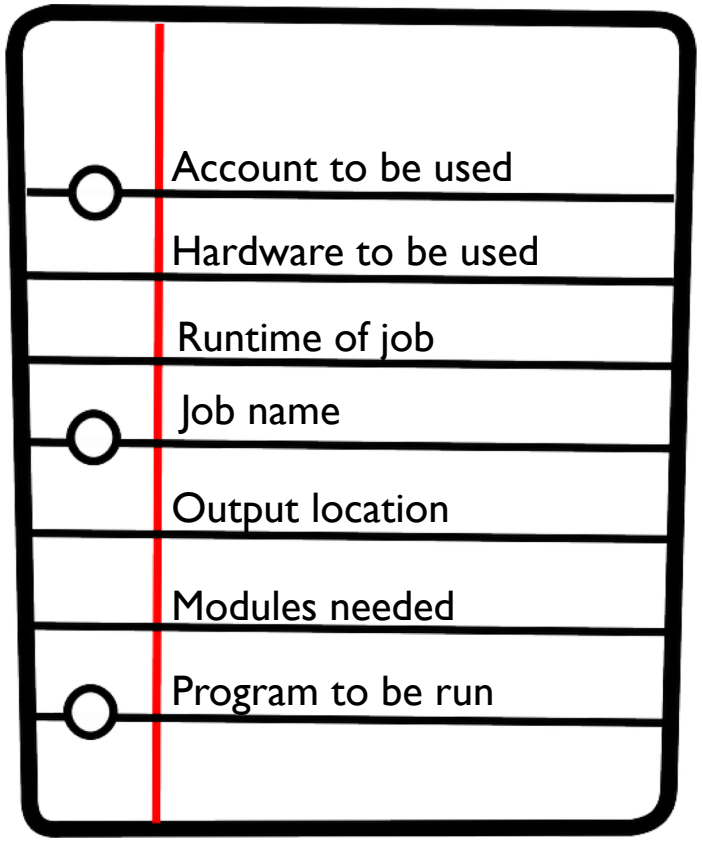
MPI Hello World

```cpp
#include <iostream>
#include <mpi.h>

int main(int argc, char **argv)
{
    // initialize MPI
    MPI_Init(&argc , &argv);

    // Get the rank of the running process
    int my_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);

    // print the rank
    std::cout << " my_rank: " << my_rank << std::endl;

    //finalize MPI
    MPI_Finalize();

    return 0;
}
```



rc-help@colorado.edu

SLURM JOB SCRIPTS



slurm_script.sh

```
1  #!/bin/bash
2
3  #SBATCH --account=...
4
5  #SBATCH --partition=amilan
6  #SBATCH --nodes=1
7  #SBATCH --ntasks=4
8
9  #SBATCH --time=01:00:00
10
11 #SBATCH --job-name=hello-world
12
13 #SBATCH --output=hello-world.%j.out
14
15 module purge
16
17 module load gcc
18 module load openmpi
19
20 mpirun -np 4 ./hello_world.exe
```

SUBMITTING A SLURM SCRIPT