

HIGH-PERFORMANCE COMPUTING + EFFICIENCY



OPERATIONS
RESEARCH
CENTER

IAP 15.S60 Session 8
Alex Schmid

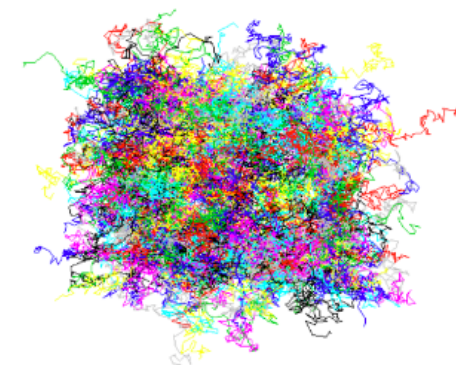
Today's Learning Objectives

- Submit interactive jobs, batch jobs, and job arrays on a computing cluster
- Recognize parallelizable code and implement a simple parallel job with shared memory
- Design a reproducible and efficient pipeline for scientific computing

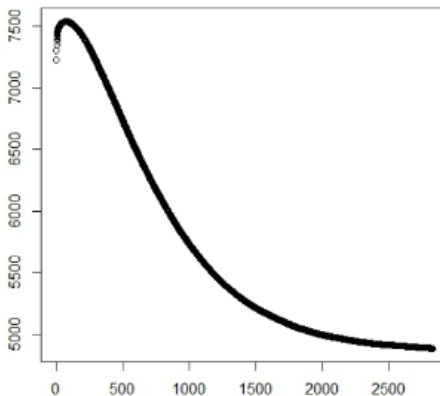
Clusters

Why do we use computing clusters?

In optimization and stats, we often need a lot of computational power:



Machine learning on a large
dataset



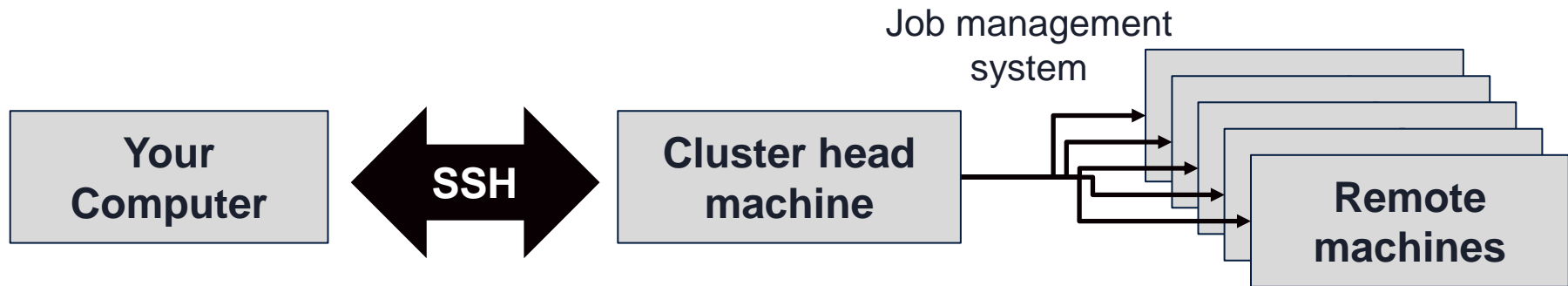
Simulations that require many
iterations to converge



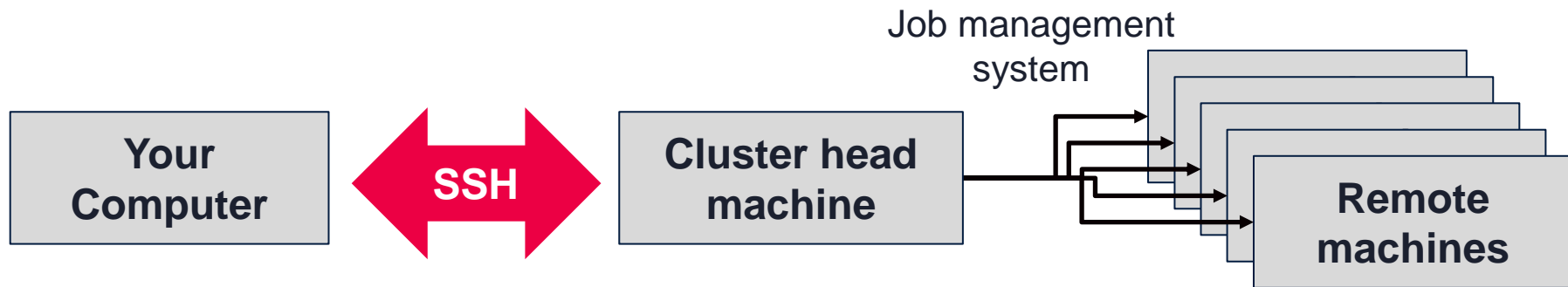
GUROBI
OPTIMIZATION

Hard optimization problems

Using a cluster



Using a Cluster

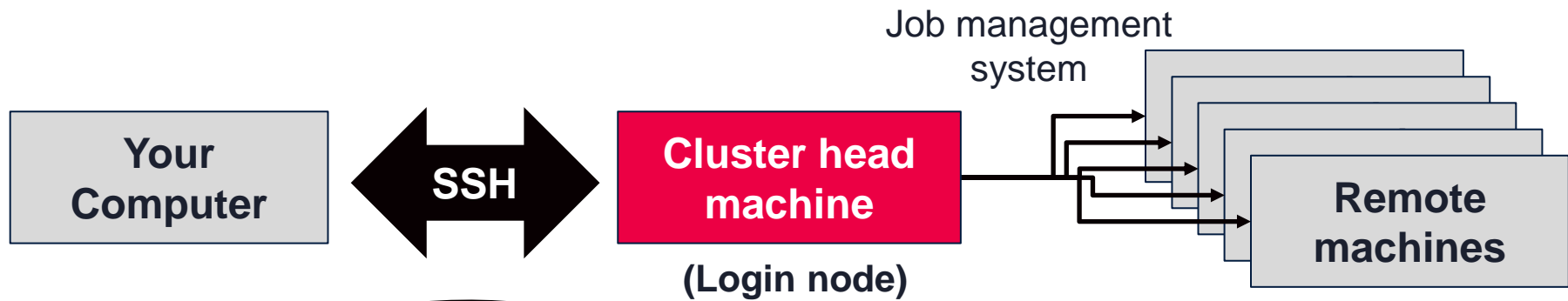


Hi, I'd like
to make some
crafts

Credit card and
ID, please



Using a Cluster

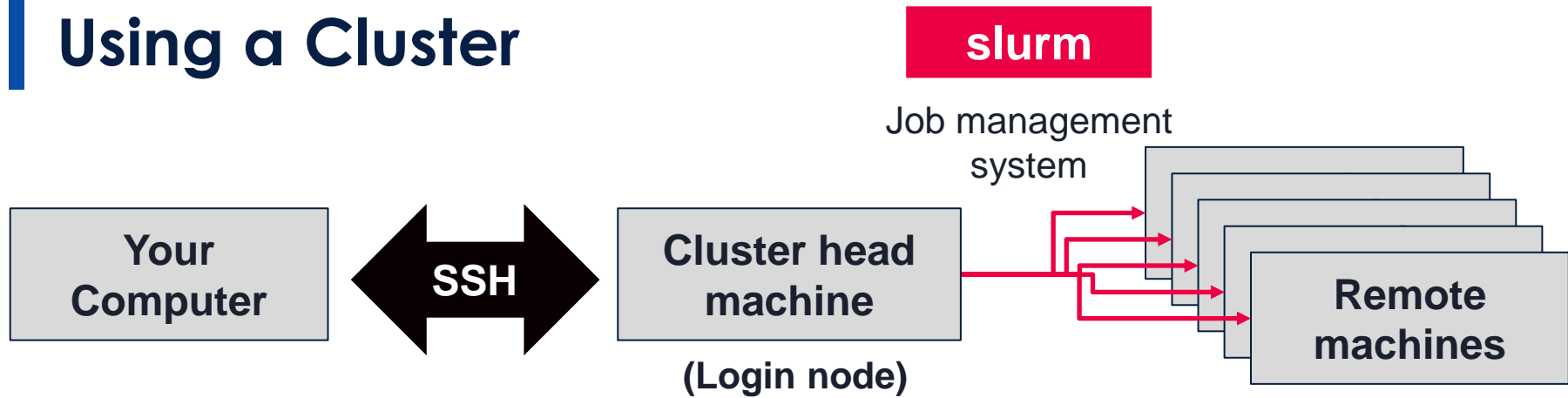


One hour studio
time, paint, and
a brush

What would
you like?



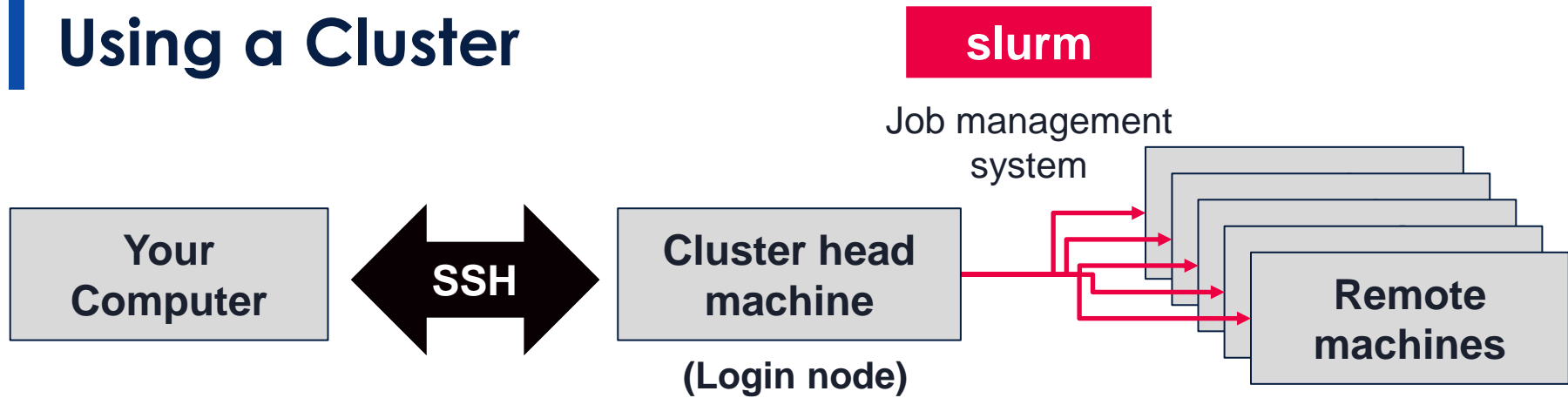
Using a Cluster



Sure thing!
Here you go...



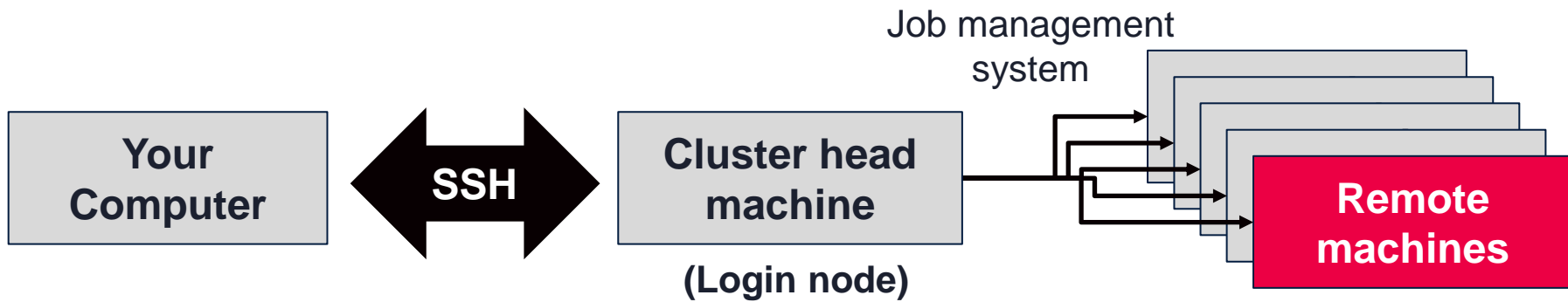
Using a Cluster



Sure thing!
Here you go...



Using a Cluster



Sure thing!
Here you go...

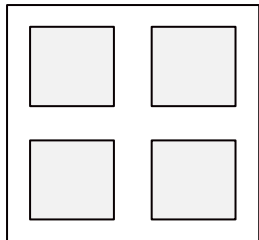


Partitions

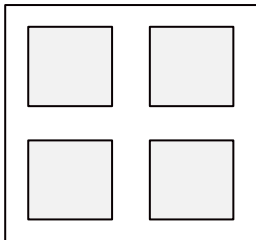
Clusters have many **nodes**, grouped into **partitions**

Engaging

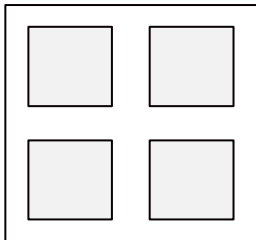
sched_any_quicktest



sched_any

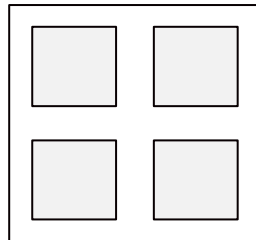


newnodes

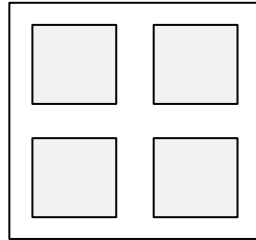


Sloan

sched_mit_sloan_
interactive

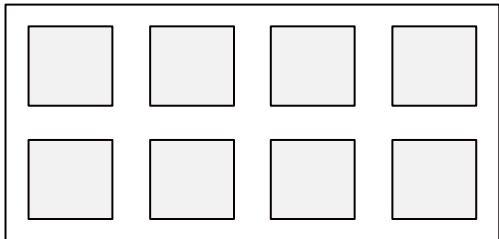


sched_mit_sloan_
batch

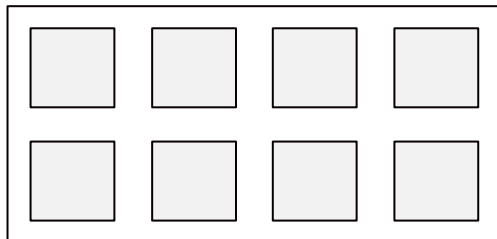


SuperCloud

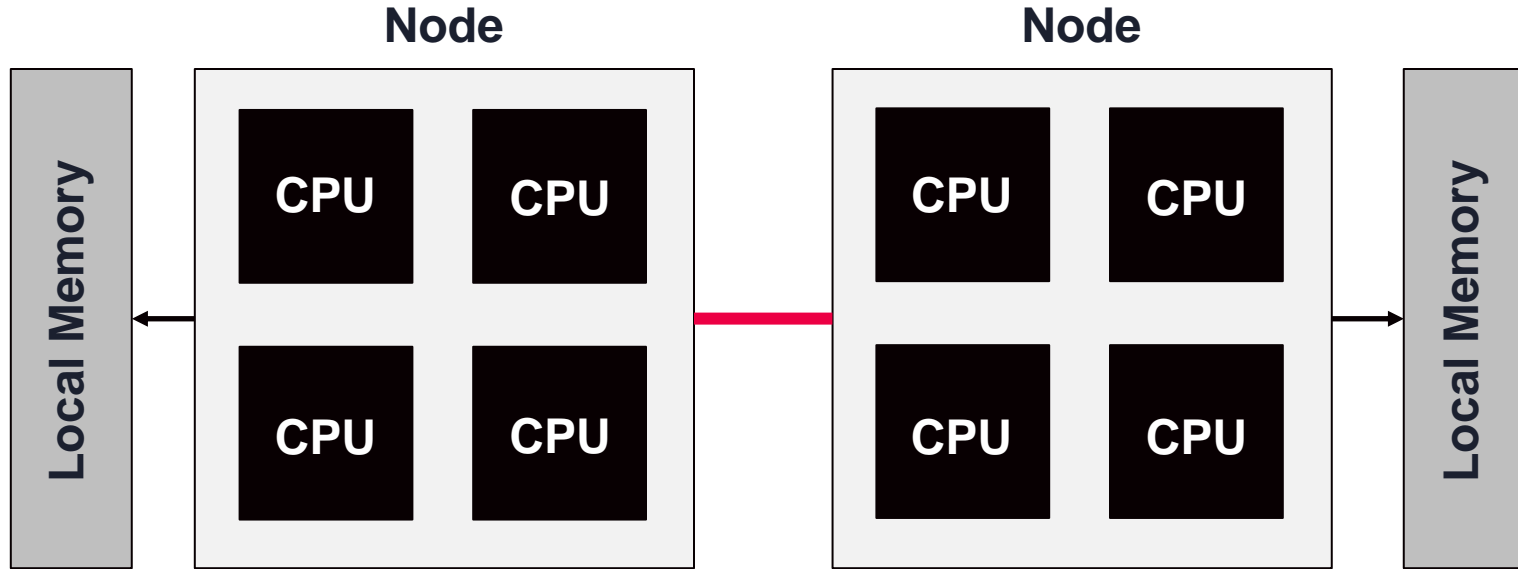
xeon-p8



xeon-p8-volta



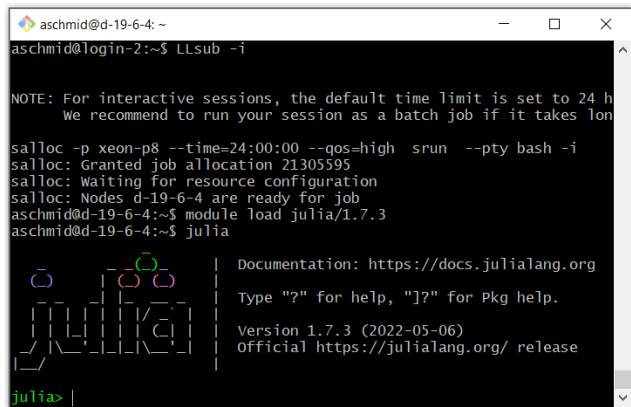
Nodes



Types of Jobs

Interactive jobs

- Use cluster resources to interact with your code as you would locally
- Prototyping, testing, long jobs



```
aschmid@d-19-6-4: ~  
aschmid@login-2:~$ LLsub -i  
  
NOTE: For interactive sessions, the default time limit is set to 24 h  
We recommend to run your session as a batch job if it takes lon  
  
salloc -p xeon-p8 --time=24:00:00 --qos=high srun --pty bash -i  
salloc: Granted job allocation 21305595  
salloc: Waiting for resource configuration  
salloc: Nodes d-19-6-4 are ready for job  
aschmid@d-19-6-4:~$ module load julia/1.7.3  
aschmid@d-19-6-4:~$ julia  
  
Documentation: https://docs.julialang.org  
Type "?" for help, "]"? for Pkg help.  
Version 1.7.3 (2022-05-06)  
Official https://julialang.org/ release  
  
julia> |
```

Batch jobs

- Set it and forget it! Request cluster resources to run scripts, then check back later for your results
- “Official” runs, running multiple scripts

```
1#!/bin/bash  
2#SBATCH -a 151-300  
3#SBATCH --cpus-per-task=2  
4#SBATCH --mem=32G  
5#SBATCH --partition=sched_mit_sloan_batch  
6#SBATCH --time=1-00:00  
7#SBATCH -o /home/aschmid/warehouse-task-assignment/outerr/train_both_\%a.out  
8#SBATCH -e /home/aschmid/warehouse-task-assignment/outerr/train_both_\%a.err  
9#SBATCH --mail-type=BEGIN,END,FAIL  
10#SBATCH --mail-user=aschmid@mit.edu  
11  
12module load julia/1.5.2  
13module load gurobi/8.1.1  
14  
15julia run_gettrainingdata_both.jl $SLURM_ARRAY_TASK_ID
```

Poll Question

pollev.com/alexandriaschmid171

What cluster are you using today?

- A.) Engaging OnDemand
- B.) Engaging with Sloan resources
- C.) SuperCloud

Log in to the cluster with SSH

Open a terminal or Git Bash window

Engaging OnDemand:

```
ssh username@eofe7.mit.edu
```

```
ssh username@eofe8.mit.edu
```

Engaging via Sloan:

```
ssh username@eosloan.mit.edu
```

SuperCloud:

```
ssh username@txe1-login.mit.edu
```

Login Node

This will land you on the cluster login node

```
aschmid@login-2:~$
```

Don't run code on the login node! It doesn't have many compute resources and you may cause issues for others trying to log in.

- **Exception:** on SuperCloud, only the login node has internet access, so you must use it to install new software, add packages, clone repos, etc.

File system on the cluster

- Analogous to the file system on your local machine
 - We must move files and data to the cluster to run them
- **Engaging:** `/home/username`
- **SuperCloud:** `/home/gridsan/username`

Interacting with file system

Usual terminal commands

```
pwd, ls, cd
```

Move files from local to cluster

```
scp filename username@hostname:destination_file_path
```

e.g. `scp filename username@eofe7.mit.edu:/home/username/<destfolder>`

`scp filename username@eosloan.mit.edu:/home/username/<destfolder>`

`scp filename username@txel-login.mit.edu:/home/gridsan/username/<destfolder>`

Move **folder** from local to cluster

```
scp -r foldername/ username@hostname:destination_file_path
```

File Transfer – Try it out

Open a **second** terminal window to manipulate local files and run the following:

1. **Local window:** Create a new file on your local machine

```
touch newfile.txt
```

2. **Local window:** Move the file to your home directory on the cluster

```
scp newfile.txt user@eofe7.mit.edu:/home/user  
scp newfile.txt user@txel-login.mit.edu:/home/gridsan/user
```

3. **Cluster window:** Create a new folder on the cluster

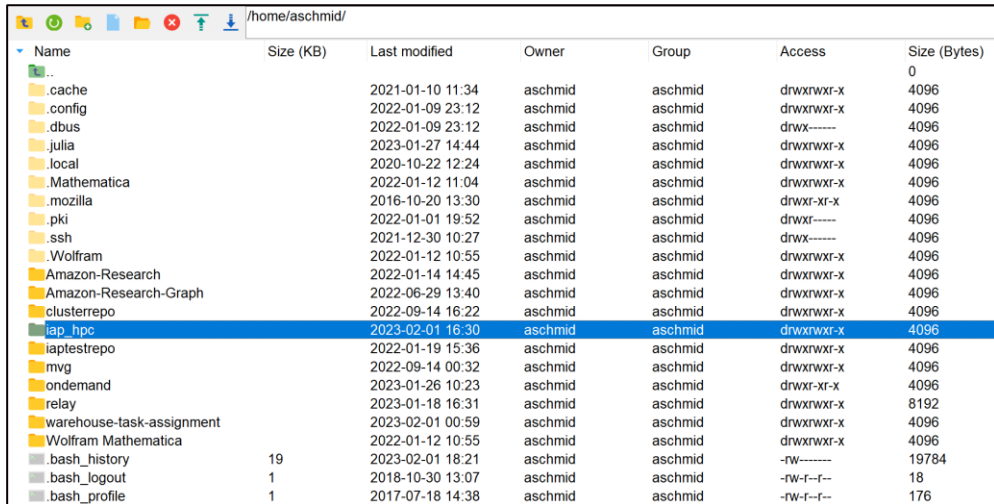
```
mkdir newfolder
```

4. **Local:** Move the new folder to your machine

```
scp -r user@eofe7.mit.edu:/home/user/newfolder/ ./  
scp -r aschmid@txel-login.mit.edu:/home/gridsan/user/newfolder/ ./
```

Aside: Graphical Interface for Files

- Engaging OnDemand has a [GUI](#) interface for viewing and manipulating your files
- Windows users can also download and use [MobaXterm](#), which has a nice GUI file system in addition to a shell for running slurm commands
- You can view the SuperCloud file system through the [web portal](#)



Name	Size (KB)	Last modified	Owner	Group	Access	Size (Bytes)
..						0
.cache		2021-01-10 11:34	aschmid	aschmid	drwxrwxr-x	4096
.config		2022-01-09 23:12	aschmid	aschmid	drwxrwxr-x	4096
.dbus		2022-01-09 23:12	aschmid	aschmid	drwx-----	4096
.julia		2023-01-27 14:44	aschmid	aschmid	drwxrwxr-x	4096
.local		2020-10-22 12:24	aschmid	aschmid	drwxrwxr-x	4096
.Mathematica		2022-01-12 11:04	aschmid	aschmid	drwxrwxr-x	4096
.mozilla		2016-10-20 13:30	aschmid	aschmid	drwxr-xr-x	4096
.pki		2022-01-01 19:52	aschmid	aschmid	drwxr-----	4096
.ssh		2021-12-30 10:27	aschmid	aschmid	drwx-----	4096
.Wolfram		2022-01-12 10:55	aschmid	aschmid	drwxrwxr-x	4096
Amazon-Research		2022-01-14 14:45	aschmid	aschmid	drwxrwxr-x	4096
Amazon-Research-Graph		2022-06-29 13:40	aschmid	aschmid	drwxrwxr-x	4096
clusterrepo		2022-09-14 16:22	aschmid	aschmid	drwxrwxr-x	4096
lap_hpc		2023-02-01 16:30	aschmid	aschmid	drwxrwxr-x	4096
iaptestrepo		2022-01-19 15:36	aschmid	aschmid	drwxrwxr-x	4096
mvq		2022-09-14 00:32	aschmid	aschmid	drwxrwxr-x	4096
ondemand		2023-01-26 10:23	aschmid	aschmid	drwxr-xr-x	4096
relay		2023-01-18 16:31	aschmid	aschmid	drwxrwxr-x	8192
warehouse-task-assignment		2023-02-01 00:59	aschmid	aschmid	drwxrwxr-x	4096
Wolfram Mathematica		2022-01-12 10:55	aschmid	aschmid	drwxrwxr-x	4096
.bash_history	19	2023-02-01 18:21	aschmid	aschmid	-rw-----	19784
.bash_logout	1	2018-10-30 13:07	aschmid	aschmid	-rw-r--r--	18
.bash_profile	1	2017-07-18 14:38	aschmid	aschmid	-rw-r--r--	176

We can also use Git and Github!

In the terminal logged in to your cluster, clone today's repo into your home directory:

```
git clone https://github.com/alexschmid3/iap_hpc.git
```

Let's check it out! Four folders for today's four examples:

- 1_interactive (_sc for SuperCloud or _eng for Engaging)
- 2_batch
- 3_array
- 4_parallel

Interactive Jobs

Starting an interactive job

Launch an interactive job with default resources

Engaging

```
srun --pty --partition=sched_any_quicktest bash
```

SuperCloud

```
LLsub -i
```

You can also specify your resources

Engaging

```
srun --pty --partition=sched_any_quicktest  
--cpus-per-task=1 --mem=2G bash
```

SuperCloud

```
LLsub -i -s 20 -g volta:1
```

Loading Software

A variety of software is installed on the cluster, including many versions of Python, Julia, R, etc. To use them, we must load the appropriate module.

Load a known module

```
module load julia/1.7.3
```

See all modules

```
module avail
```

See specific modules, e.g. Julia

```
module avail julia
```


Start an Interactive Job

1. Start your interactive job

```
srun --pty --partition=sched_any_quicktest bash
```

```
LLsub -i
```

2. Load Julia

```
module load julia/1.7.3
```

3. Navigate to the first directory for today

```
cd 1_interactive_sc or cd 1_interactive_eng
```

4. Run our test script

```
julia testscript.jl
```

Passing Arguments

Let's run some of the scripts in `1_interactive` which find the shortest path between two nodes in the network given in the CSV file

1. First, let's check out the network details

```
julia networkdetails.jl
```

2. Let's find the shortest path between node 1 and node 20

```
julia shortestpath_noargs.jl
```

3. Now, let's pass two arguments to find the shortest path between nodes 6 and 7

```
julia shortestpath_args.jl 6 7
```

Batch Jobs

Batch Jobs

Instead of interacting directly, we'll tell the cluster to run a set of commands on its own time!

- Run scripts with long computational time or that require lots of resources
- Running many scripts at once (e.g. testing an algorithm on 100 different datasets or instances)

Shell scripts

We tell the cluster what commands to execute with a shell script:

```
#!/bin/bash
```

“shebang”

```
#Set up computing environment
```

```
#SBATCH --cpus-per-task=2
```

```
#SBATCH --mem=16G
```

```
#SBATCH --partition=sched_mit_sloan_batch
```

```
#SBATCH --time=1-00:00
```

**request resources
(optional)**

```
#Load software
```

```
module load julia/1.7.3
```

```
module load gurobi/9.0.3
```

load software

```
#Run the script as usual
```

```
julia myscript.jl
```

**run script, pass
arguments**

Kicking off a batch job

Write your batch shell script!

```
touch mybatchfile.sh  
nano mybatchfile.sh
```

Kickoff job

```
sbatch mybatchfile.sh
```

Monitor your jobs

Engaging:

```
eo-show-myjobs
```

SuperCloud:

```
LLstat
```

Batch Job – Try it out

1. Navigate to the folder `2_batch_sc` or `2_batch_eng`

2. Take a look at the batch file

```
cat batchjob.sh
```

3. Kick it off, check it out, then look at the results!

```
sbatch batchjob.sh
```

```
eo-show-myjobs or LLstat
```

```
cat output.txt
```

Resources

Task	Syntax
CPUs	<code>--cpus-per-task=1</code>
Memory	<code>--mem=4G</code>
Time	<code>--time=1-00:00</code>

Engaging

Check partition resources with:

```
eo-show-partition
```

SuperCloud

Check available resources with:

```
LLfree
```

When the cluster is busy, your job will be queued until the resources are available.
The job will fail if you use more than the requested memory.

Takeaways

- Interactive jobs let you code as if the cluster were your computer, while batch jobs run without your intervention
- Use interactive jobs when prototyping and testing, use batch jobs to kick off your final runs
- **Never run your scripts on the login node!** Use `sbatch` to start a batch job, or `srun / LLsub -i` to start an interactive job before running your scripts

Break

Resume in 10 minutes

Job Arrays

Submitting many batch jobs at once

We have a script and we want to run it for several different parameters.

e.g. testing a new optimization model or algorithm on many instances

Let's check out an example in this folder:

`3_array_eng/`

or

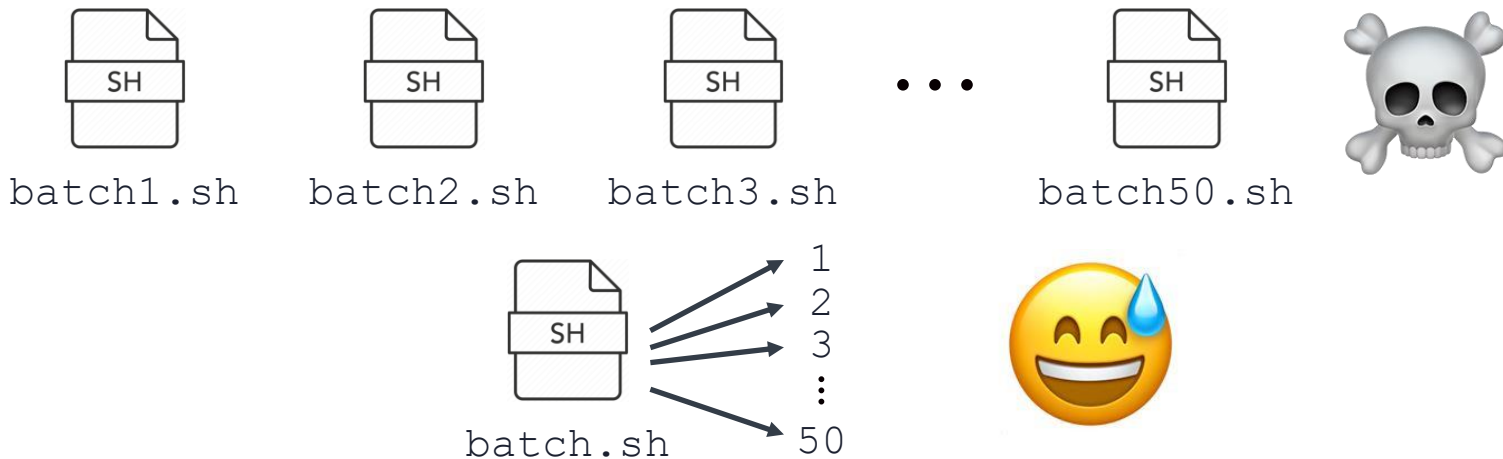
`3_array_sc/`

How would you approach this?

Script: `shortestpath_one.jl`

Instance list: `data/`

How can we use the cluster to accomplish this?



Job Array Batch Script - Engaging

```
#!/bin/bash
```

```
#SBATCH -a 1-50
```

```
#SBATCH --cpus-per-task=1
```

```
#SBATCH --mem=2G
```

```
#SBATCH --partition=sched_mit_sloan_batch
```

```
#SBATCH --time=0-00:10
```

```
#SBATCH -o /home/aschmid/iap_hpc/run_\%a.out
```

```
#SBATCH -e /home/aschmid/iap_hpc/run_\%a.err
```

```
#Load software
```

```
module load julia/1.7.3
```

```
#Run the script as usual
```

```
julia shortestpath_one.jl $SLURM_ARRAY_TASK_ID
```

Job Array Batch Script - [SuperCloud](#)

```
#!/bin/bash
```

```
#SBATCH -o array.sh.log-%a
```

```
#SBATCH -a 1-3
```

```
#Load software
```

```
module load julia/1.7.3
```

```
julia shortestpath_many.jl $SLURM_ARRAY_TASK_ID  
$SLURM_ARRAY_TASK_COUNT
```

Poll Question

pollev.com/alexandriaschmid171

What would happen if we kicked off `shortestpath_one.jl` using the batch script we just looked?

- A.) The cluster runs experiments 1-50
- B.) The cluster runs experiment 1 fifty times

Modifying our Julia script

We need to tell Julia that we'll be passing arguments and specify how she should handle them. For example,

```
runid = parse{Int, ARGS[1]}

networkfile = string("data/network", runid, ".csv")
outputfile = string("outputs/network", runid, ".csv")
```

Kickoff Job Array – Try it out

1. Kick off the batch job

```
sbatch array.sh
```

2. Once it completes, check out the `outputs` folder

```
ls outputs/
```

3. Run the script `combineoutputfiles.jl` to gather the outputs from the experiments into one file (use an interactive job to avoid login node!)

```
srun... or LLsub -i  
module load julia/1.7.3  
julia combineoutputfiles.jl
```

4. Check out the combined file

```
cat outputs/combined.csv
```

Poll Question

pollev.com/alexandriaschmid171

What if instead of having each of the 50 runs write to its own output file, we had them all write to one file?

- A.) All 50 would write to the output file, but the rows may be in a random order
- B.) Some of the 50 may overwrite each other, leaving an incomplete output file
- C.) An error, as all jobs are trying to access the same file at the same time

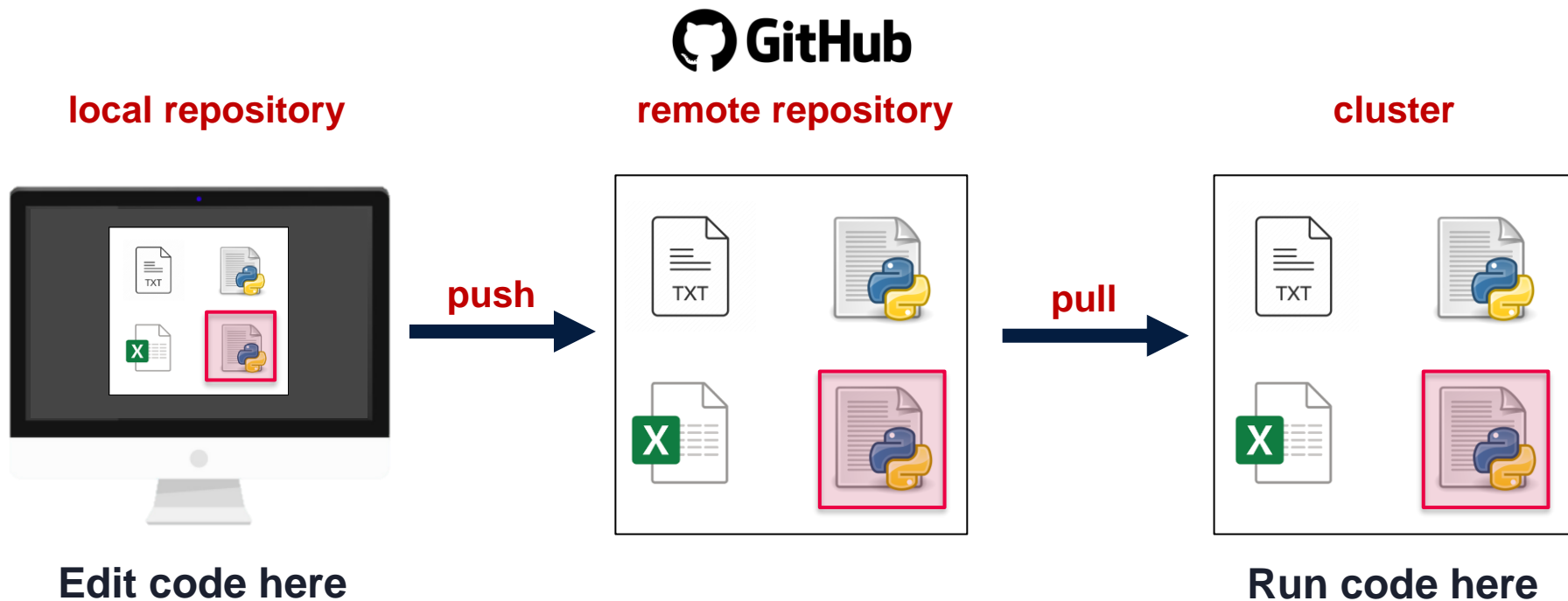
Best Practices

- Before kicking off n jobs (where n is large), test code locally, then perhaps interactively, then kick-off 1-3 batch runs to avoid having all n fail
- To automate the last step (gathering files), check out [LLMapReduce](#) if you're using SuperCloud

Collaboration Tips

Collaborating with the Cluster

We know we can use Git/Github on the cluster. We can then:



Connecting Engaging to Github

Follow the link below for instructions on how to generate an SSH key for your Github account, add your key to the SSH agent, then add the key to your Github account <https://docs.github.com/en/get-started/quickstart/set-up-git>

Aside: Data Storage on the Cluster

- If your data is small, you can just store it on Github
- Otherwise, SuperCloud has a nice collaboration features that lets you create a shared group directory, which will be stored in `/home/gridsan/groups`
 - To request a group directory, send email to `supercloud@mit.edu` ([details](#))

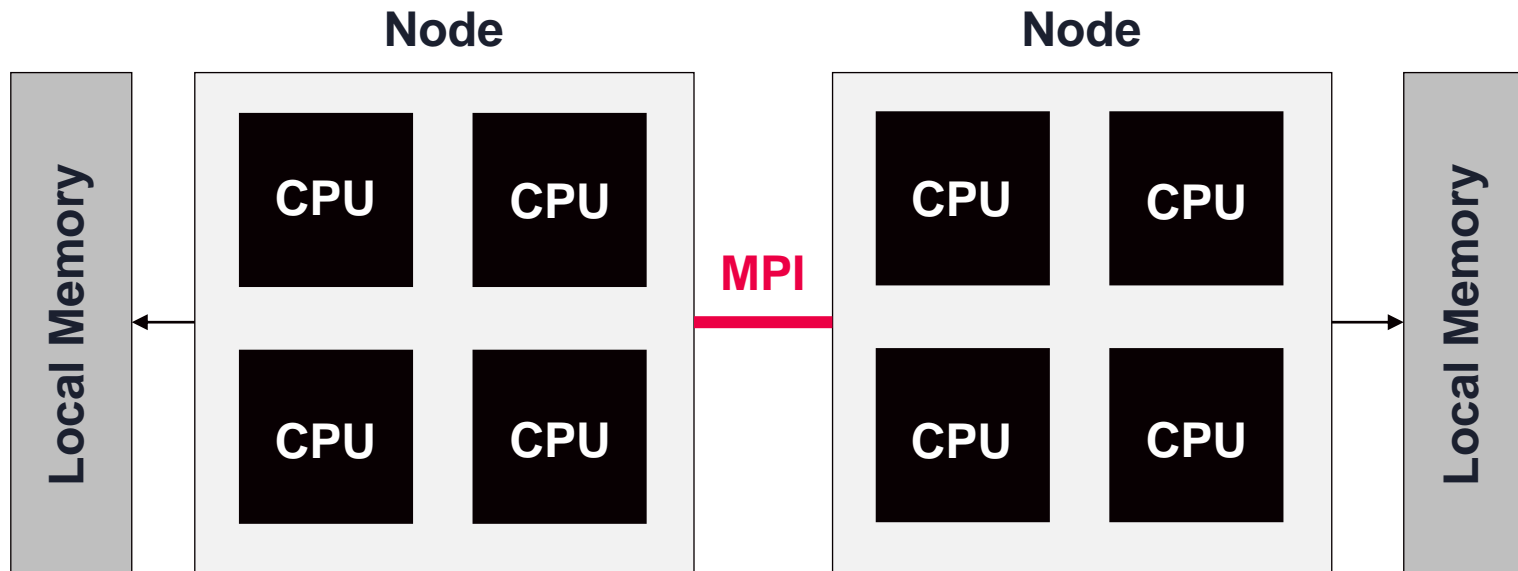
Storage space:

- **Engaging:** `/home/username` (100GB per user)
`/pool001/username` (1TB per user)
- **SuperCloud:** `/home/gridsan/username` (no limit)

Parallel Computing on SuperCloud

Adapted from Lauren Milechin

Distributed computing



Parallelization

We can use **MPI (Message Passing Interface)** to parallelize our code!

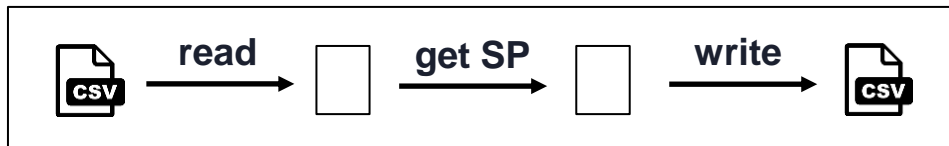
Unlike the commands we've run so far today, integrating MPI requires language-specific code

→ We will use **MPI.jl** to integrate parallelization into our Julia code

```
1 using MPI
2
3 # Initialize MPI environment
4 MPI.Init()
5
6 # Get MPI process rank id
7 rank = MPI.Comm_rank(MPI.COMM_WORLD)
8
9 # Get number of MPI processes in this communicator
10 nproc = MPI.Comm_size(MPI.COMM_WORLD)
11
12 # Print hello world message
13 print("Hello world, I am rank $(rank) of $(nproc)
14 processors\n")
15
16
```

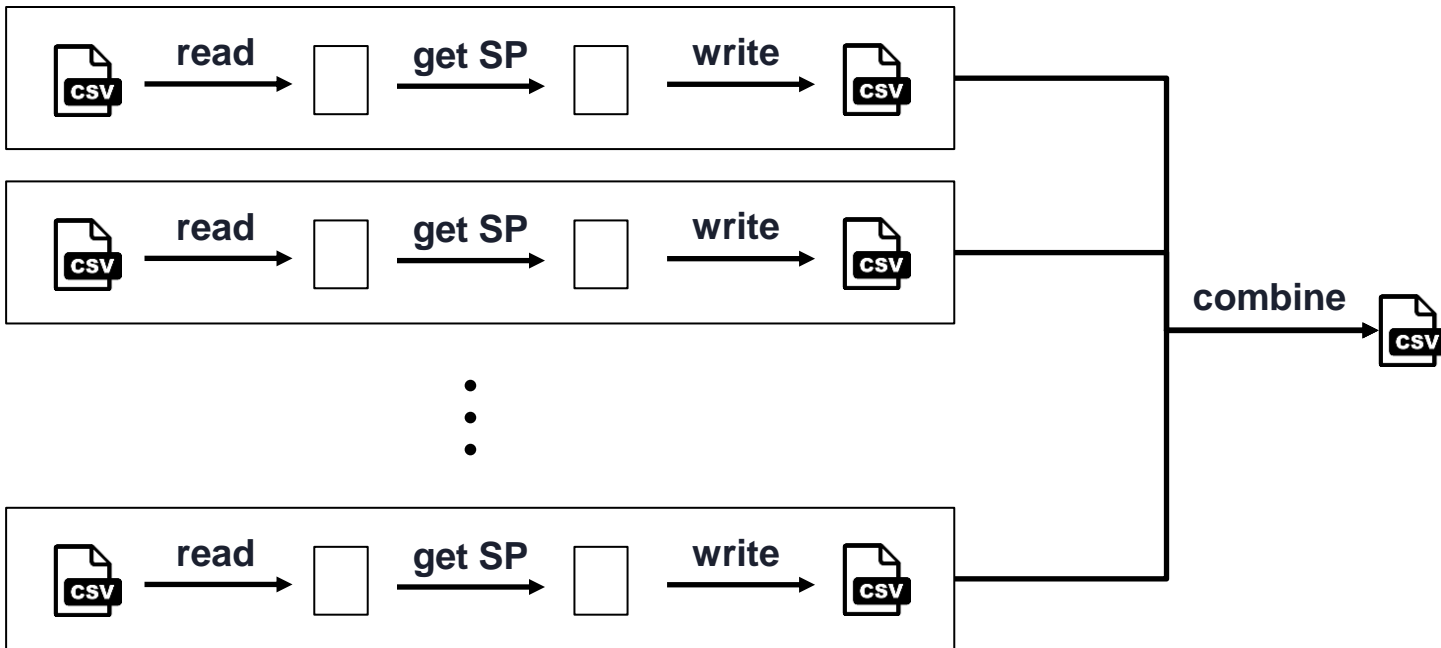
Option 1: One big for loop

```
for runid in 1:50
```

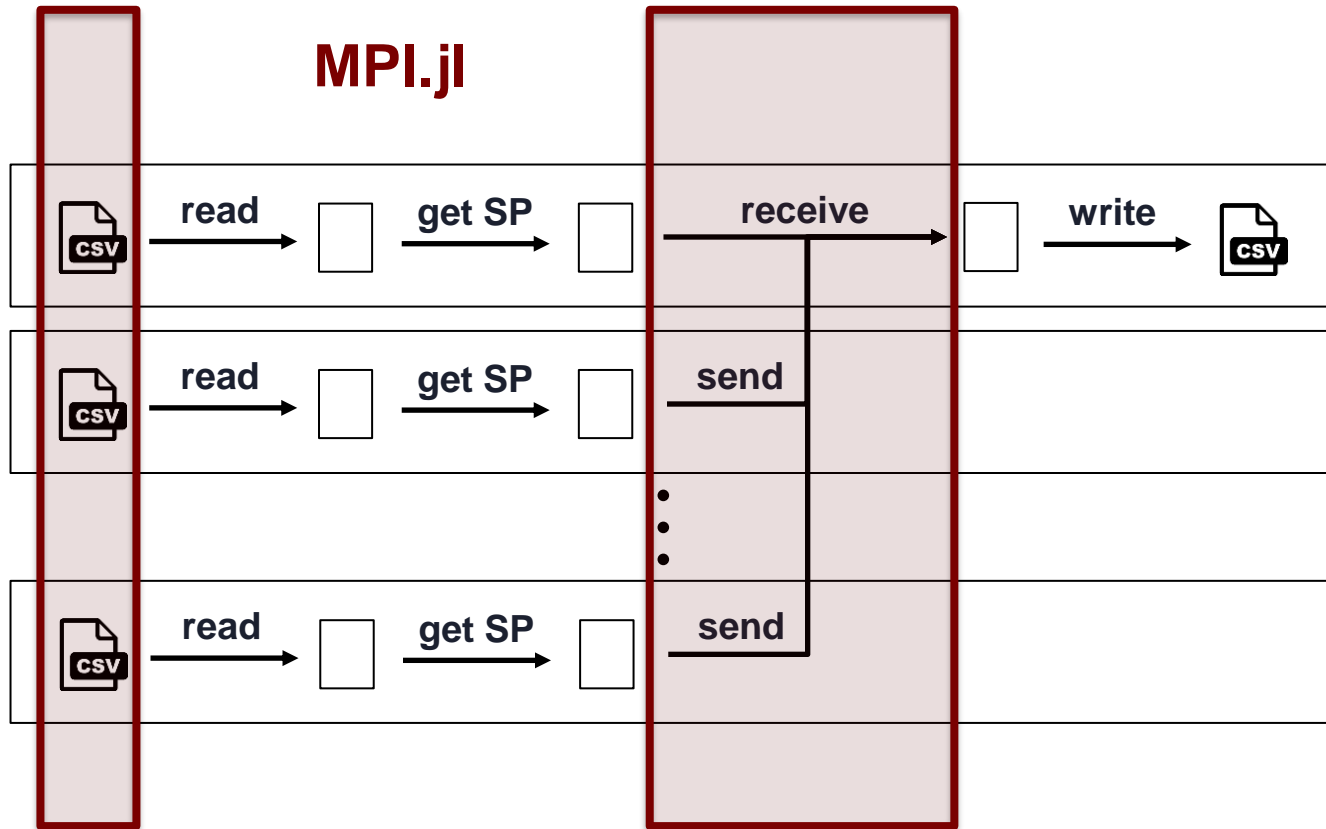


```
end
```

Option 2: Job array / MapReduce



Option 3: Parallelize across cores



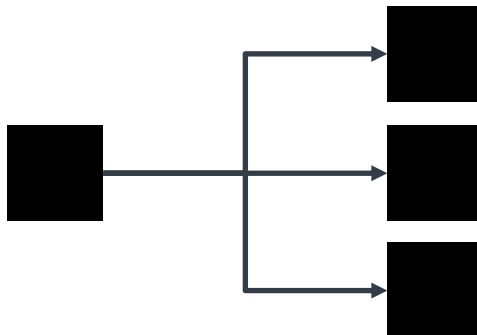
MPI – Basic Messaging

Point-to-point

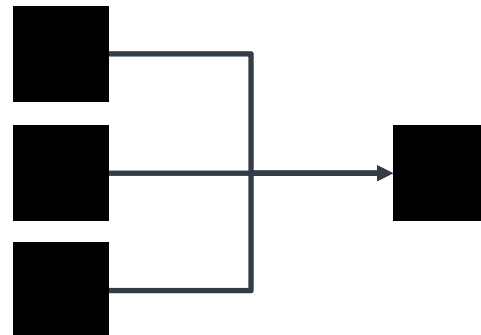
(we'll stick to this one)



Broadcast



Gather



MPI Commands

Initialize MPI environment

```
MPI.Init()
```

MPI communicator

```
MPI.COMM_WORLD
```

The number of MPI processes

```
MPI.Comm_size
```

The rank / ID of a given process

```
MPI.Comm_rank
```

Send message

```
MPI.send(message, recvr_rank, my_id, comm)
```

Receive message

```
MPI.send(sender_rank, my_id, comm)
```


Running parallelized code with MPI

1. Navigate to `4_parallel_sc`

2. Load `mpi`, add `MPI.jl`, and build `MPI.jl`

```
module load mpi
julia
using Pkg
Pkg.add("MPI")
Pkg.build("MPI")
```

3. Kickoff `hello.jl` with `mpi` and four cores

```
mpirun -n 4 julia hello.jl
```

Poll Question

pollev.com/alexandriaschmid171

What do you expect to happen when I run `mpi run -n 4 hello.jl`?

A.) Hello, World!

B.)
Hello, World!
Hello, World!
Hello, World!
Hello, World!

Questions?

Final assignment in 5_assignment

Due: Monday, Feb. 6 at 11:59pm (hard deadline)

I will be sticking around until noon. I encourage you to work on the assignment now and ask any questions, that way you don't have to worry over the weekend.

Best of luck in the upcoming semester 😊

Thank you!