Parallel and Distributed Computing for Optimization and Statistics

Chris McCord[1]

MIT

February 1, 2018

[1]Special thanks to Sebastien Martin for providing many of the slides!

Heavy Computations

In Optimization and Statistics, we often need a lot of computational power:

- Machine Learning on large datasets
- Simulations that require many iterations to converge
- Hard optimization problems, mixed integer programming

Heavy Computations

In Optimization and Statistics, we often need a lot of computational power:

- Machine Learning on large datasets
- Simulations that require many iterations to converge
- Hard optimization problems, mixed integer programming

Or repetitive computations:

- Parameter tuning
- Benchmarking

Limitations of a Personal Computer

Using your personal computer may seem simple, but there are serious limitations:

- Limited memory (Big Data, large matrices...)

Limitations of a Personal Computer

Using your personal computer may seem simple, but there are serious limitations:

- Limited memory (Big Data, large matrices...)
- Limited computational power.

Limitations of a Personal Computer

Using your personal computer may seem simple, but there are serious limitations:

- Limited memory (Big Data, large matrices...)
- Limited computational power.
- Limited number of cores.

Limitations of a Personal Computer

Using your personal computer may seem simple, but there are serious limitations:

- Limited memory (Big Data, large matrices...)
- Limited computational power.
- Limited number of cores.
- Limited time available. (You, presumably, have better things to do than sitting and watching your program run.)

Resources

There are several ways to have access to a remote computer or a cluster.

- Another personal computer you own.
- Athena at MIT.
- Resources of your department/lab (Engaging at Sloan).
- Cloud computing as a service (Amazon AWS, Google Cloud Computing...).

Distributed Computing, Parallel Computing, Remote Computing, etc.

Many terms are used to described this type of computing, and the distinctions between them aren't always clear.

- Remote computing - execute commands on a remote machine

Distributed Computing, Parallel Computing, Remote Computing, etc.

Many terms are used to described this type of computing, and the distinctions between them aren't always clear.

- Remote computing - execute commands on a remote machine
- Parallel computing - execute multiple commands simultaneously

Distributed Computing, Parallel Computing, Remote Computing, etc.

Many terms are used to described this type of computing, and the distinctions between them aren't always clear.

- Remote computing - execute commands on a remote machine
- Parallel computing - execute multiple commands simultaneously
- Distributed computing - parallel computing without shared memory

Why Should I Use This?

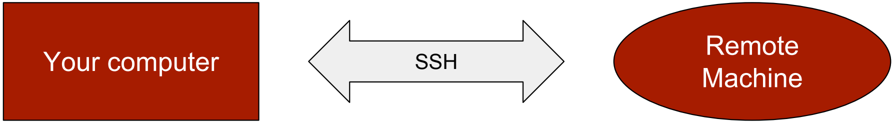
For research:

- Tackle bigger data sets in Stats and Optimization.
- Parallelize your computations.
- Longer computational times: run overnight or for a whole week!
- Can be very simple to use with interactive sessions (RStudio...)

In general:

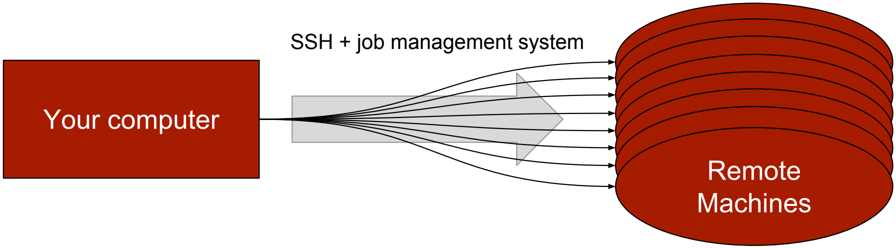
- Valuable skill, used everywhere in industry.

How does it work: using a remote computer



- We use SSH (see lecture 1) to run commands on a remote machine through our computer.
- We can use the shell to do almost anything on the remote computer: create files, run a program, use Julia...
- It is also possible to use GUI applications like R-Studio or Matlab.
- It works with any computer, including your own machines.

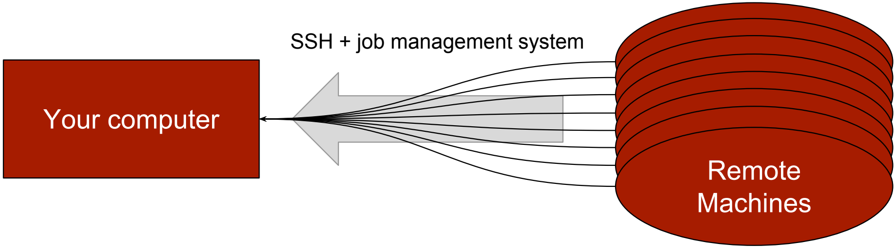
How does it work: using a cluster



When a computing cluster is available, we can run multiple "jobs" thanks to a job management system.

- Different job management systems exist, we will use the example of Slurm, which the cluster Engaging uses.
- More complex to use, but far more powerful.

How does it work: using a cluster



When a computing cluster is available, we can run multiple “jobs” thanks to a job management system.

- Different job management systems exist, we will use the example of Slurm, which the cluster Engaging uses.
- More complex to use, but far more powerful.

Engaging

Engaging is a powerful computing cluster for MIT Sloan affiliates, request an account at `stshelp@mit.edu`. It uses the job management system Slurm, that we will use in this presentation.

Connecting to Engaging

Hostname	`eosloan.mit.edu`
Username	MIT Kerberos username
Password	MIT Kerberos password

`wikis.mit.edu/confluence/display/sloanrc/Engaging+Platform`

Athena

Any MIT affiliate can access the Athena distributed computing environment, you can use it to follow if you do not have access to Engaging.

Connecting to Athena

Hostname	`athena.dialup.mit.edu`
Username	MIT Kerberos username
Password	MIT Kerberos password

`http://web.mit.edu/dialup/www/ssh.html`

SSH

As seen in Lecture 1, we can connect to the remote machine through SSH:

```
ssh <user_name>@eosloan.mit.edu
```

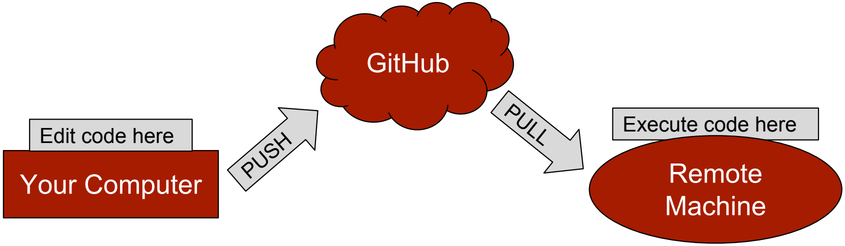
File Transfer

- Downloading from the Internet (useful to install software, or download datasets): wget
- Transferring files between the local and remote machines: Secure Copy scp

```
scp file_name.csv <user_name>@eosloan.mit.edu:
```

Using Git/GitHub to Synchronize Code

If you need to run code on the remote machine, an elegant way is to use git and GitHub:



For Advanced Users

Feel free to read up if you're interested!

- sshfs allows you to access the files of the remote machine as if they were on your own (magic)!
- screen allows you to keep your session active after you disconnect.
- tmux (terminal multiplexer) is like screen with lots of additional functionalities (panes...)

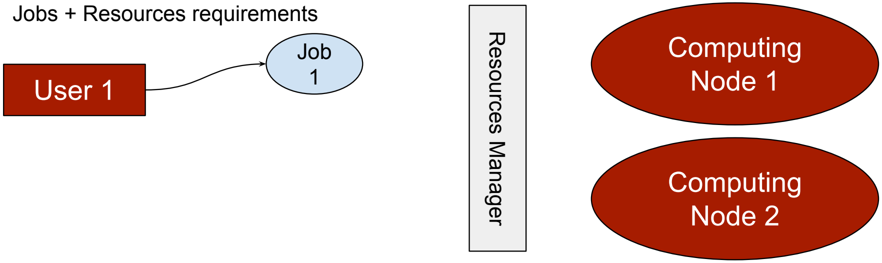
Running Jobs on a Computing Cluster

Important!

Do not run jobs/processes on the login node!

- When we login to Engaging, we are actually logging into a special login node.
- The login nodes are not designed for performing computations, but rather just for submitting jobs to the job management system.
- The actual work will be done by the computing nodes.

Running Jobs on a Computing Cluster

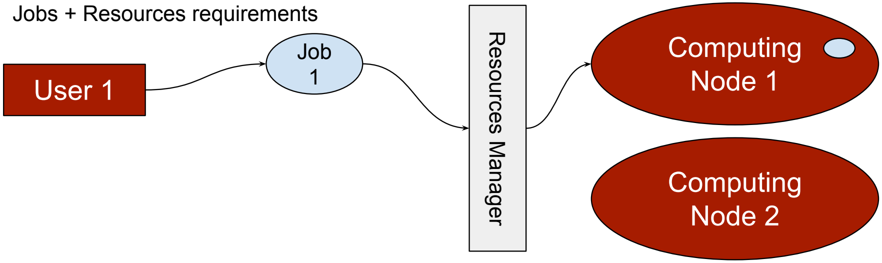


Job
: A program to be executed. (running a R/Julia/Python source file, a bash script, etc..)

Resources
: Resources required to run the job (memory, cpus, time, software requirements...)

The Job/Resource manager allocates the available resources of the computing nodes of the cluster to the different users.

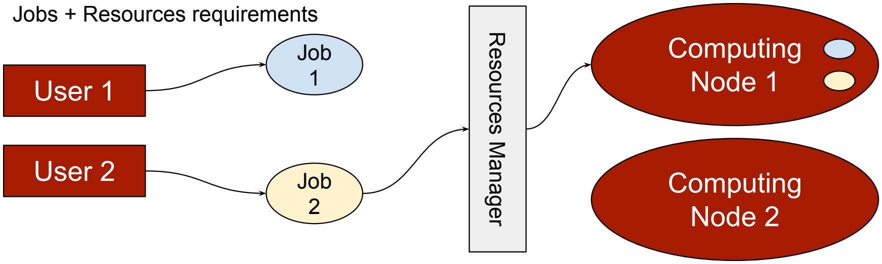
Running Jobs on a Computing Cluster



Job	A program to be executed. (running a R/Julia/Python source file, a bash script, etc..)
Resources	Resources required to run the job (memory, cpus, time, software requirements...)

The Job/Resource manager allocates the available resources of the computing nodes of the cluster to the different users.

Running Jobs on a Computing Cluster

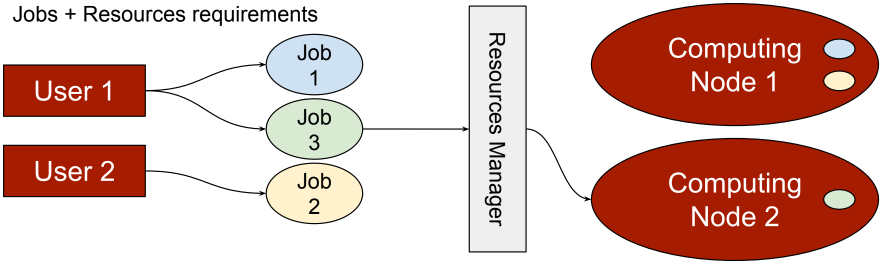


Job A program to be executed. (running a R/Julia/Python source file, a bash script, etc..)

Resources Resources required to run the job (memory, cpus, time, software requirements...)

The Job/Resource manager allocates the available resources of the computing nodes of the cluster to the different users.

Running Jobs on a Computing Cluster



Job A program to be executed. (running a R/Julia/Python source file, a bash script, etc..)

Resources Resources required to run the job (memory, cpus, time, software requirements...)

The Job/Resource manager allocates the available resources of the computing nodes of the cluster to the different users.

Module Loading

Many packages are installed on the cluster. To use them, we use the following special commands.

- Show all modules: `module avail`
- Search for modules: `eo-module-find julia`
- Load a module: `module load engaging/julia/0.6.1`

Running an Interactive Job on Engaging - `srun`

To ask SLURM, we use the command `srun`.

- The main resource parameters are:

Memory	`--mem-per-cpu=1G`
CPUs	`--cpus-per-task=1`
Time	`--time=1-12:00` (for 1 day and 12 hours)
Partition	`--partition=sched_mit_sloan_interactive`

- The partition is the name of the cluster you want to use. The two main partitions you should use are `sched_mit_sloan_interactive` and `sched_mit_sloan_batch`.
- When running interactive jobs, we should use the `--pty` flag to start the session in the current window.

Monitoring Jobs

- You can check if your job is running or in the queue: `eo-show-myjobs`
- Cancel a job: `scancel <JOB_ID>`

Running a Batch Job on Engaging - `sbatch`

- A nice way to run jobs is to save them and all their parameters in a special bash file, and use `sbatch` to run them.
- The job can be run using “`sbatch myjobname.sh`”
- The output of your jobs (what would normally appear in the console) will be saved in a `slurm-<yourjob>.out` log file where you ran `sbatch`.

Types of Parallel Computing Jobs

- “Embarassingly parallel”
  - Multiple instances of the same program with different inputs
  - Each instance can run independently, no need for shared memory
  - eg. grid search for hyperparameter tuning
- Parallel computing with shared memory
  - Commands execute simultaneously on separate processors, but work on shared memory
  - Computation cannot (easily) be split across separate nodes because communication between processes is necessary
  - eg. matrix multiplication, parallel SGD
- Advanced topics
  - Distributed computing with a message passing interface (MPI)
  - GPU computing

Other Things to Explore

There are a lot of possibilities with Engaging and other systems! Read the Engaging and SLURM documentation, use Google, etc.. For example:

- Running interactive jobs (ie with a console that you control)
- Running graphical interfaces (RStudio, SAS, Matlab...)
- Receive an email when your job completes
- Use Jupyter Notebooks on the cluster...