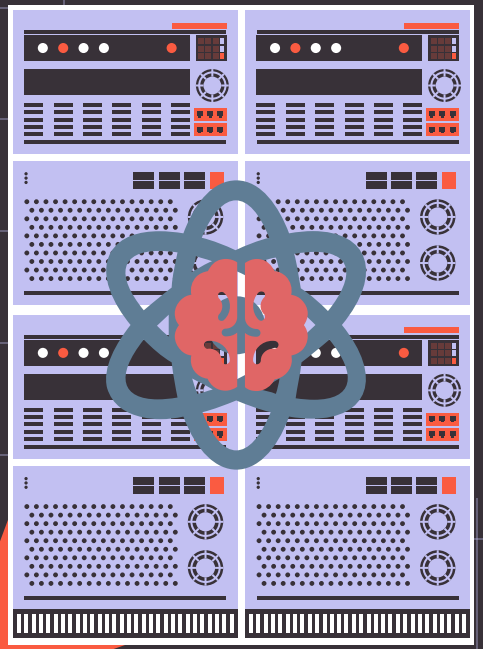
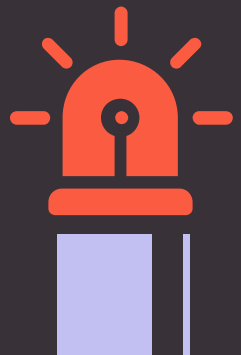


An Introduction to High Performance Computing for Neuroscience Research

Max Kramer, BrainBridge Lab

A quick note before we begin...



Overview

01

WHY
would I need/want to use an HPC for my research?

02

WHAT
is a High Performance Compute (HPC) cluster?

03

HOW
do I go about using an HPC? (Example Midway3 for SSD session)

04

OK...WHAT NEXT?
A quick primer on parallelization and optimization

WHY

would I want or need to use an HPC for my research?

01

Have you ever?

1. Written a program that takes a really long time to run?
2. Needed to use your own computer but couldn't because of a running program?
3. Worked with large data structures (MATLAB matrices, numPy arrays, fMRI/EEG data)?
4. Wanted to access a large dataset without carrying around physical storage?
5. Needed a GPU for computing but don't have $$$ to acquire one?

If so...you're in luck!

The [proper] use of a High Performance Computing (HPC) cluster can alleviate all of these issues...

...and we just so happen to have access to one here at UChicago. It's called Midway!

02

WHAT

Is a High Performance Computing (HPC) Cluster?

Like your computer...only better

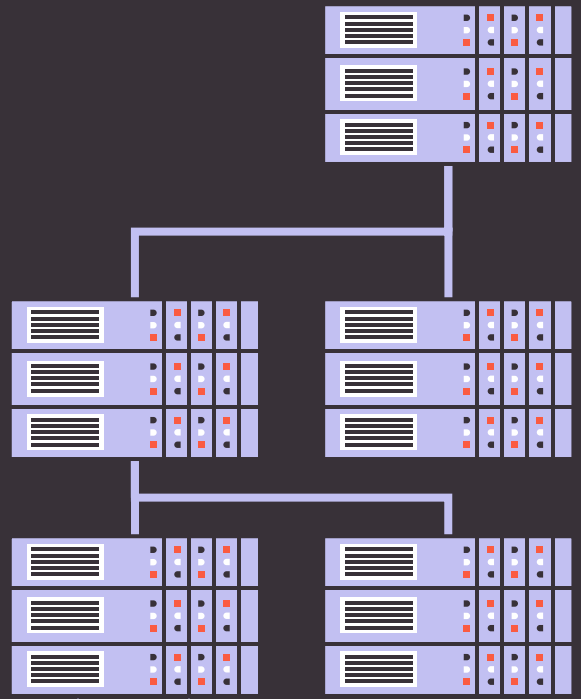
Your laptop may have:

- A quad core processor
- 8-16gb of RAM
- A gigabit ethernet port
- 256-1024gb of storage

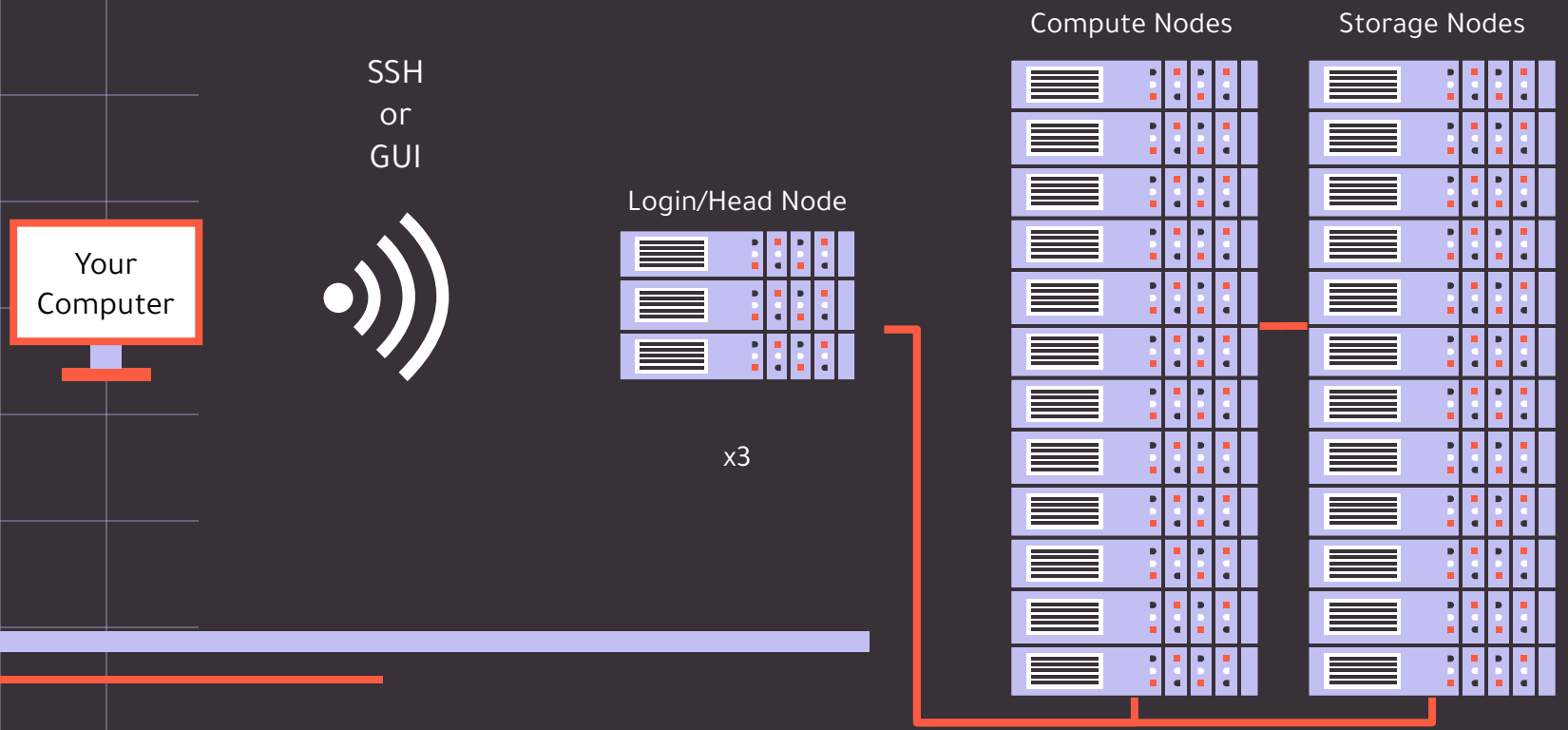
A midwaySSD node has:

- x2 20-core processors
- 192gb of RAM
- 100 gigabit interlink
- 960gb of SSD storage
- NVIDIA V100/A100/RTX 6000 GPU [COMING SOON]

..and there are 21 nodes in the system.



Basic structure of an HPC [Midway-SSD]



So what can I do with it?

Using Midway, we can:

1. Store large amounts of data securely and access it remotely

2. Build coding environments that can be shared across researchers / labs

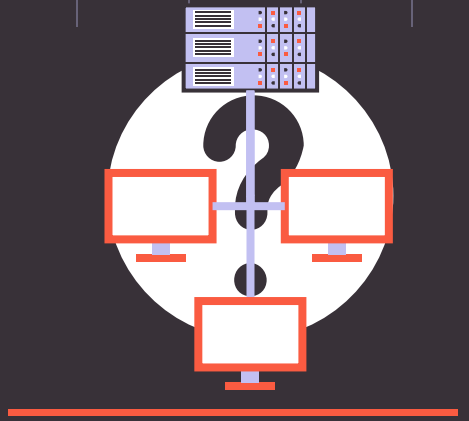
3. Run tasks independent of local machines, freeing up time for other work

4. Run tasks that require more computational power than local machines can provide

And more!

HOW
do I go about using Midway-SSD?

03



Step 1: Getting access to MidwaySSD

You will need:

1. A Midway user account (either PI or student)

   If you don't have one, you can request one: https://midwayssd.rcc.uchicago.edu/getting-started/

Either:

   2a. A connection to the Uchicago VPN

   2b. A wired internet connection to your lab

Step 2: Connecting to MidwaySSD

MAC OS

1. Open Terminal

WINDOWS

1. Open Powershell or Command Prompt

2. Run the following command:

```
ssh <CNET ID>@ssd.rcc.uchicago.edu
```

3. When prompted, enter your CNET password and then enter '1' to send a 2FA push to your phone

Step 3: Transferring Data to MidwaySSD

There are several ways to go about this!

Secure Copy (SCP) from your local machine

Open either Powershell (Windows) / Terminal (Mac)

```
scp -r BLRB_3_15 <CNET ID>@ssd.rcc.uchicago.edu:
```

Secure File Transfer Protocol (SFTP) Client

Windows: WinSCP (https://winscp.net/eng/index.php)
Mac OS: CyberDuck (https://cyberduck.io/)

SAMBA: “Mount” your Midway storage on your local machine
Globus: Online service for moving large files to/from MidwaySSD

Step 4: Navigating around MidwaySSD

CLI

Common bash commands:

`pwd`: print current working directory

`ls`: view files and directories in current directory

`cd`: change directory

GUI

1. Connect to the Uchicago VPN
2. In a web browser, type:

   ssd.rcc.uchicago.edu

3. Login with your CNET & Password
4. Navigate around the desktop in a graphical manner

Step 5: Set up for analyses on MidwaySSD

Step 1: From home directory

```
unzip BLRB_3_15.zip
```

Step 2: Navigate to BLRB_3_15

```
cd BLRB_3_15
```

MATLAB

1. `module load matlab`
2. `matlab`

PYTHON

1. `module load python`
2. `conda env create -f environment.yml`
3. `conda activate BLRB_environment`
4. `python`

Running your code on MidwaySSD

There are two ways to run your code on MidwaySSD

1. An interactive session

   Launch using GUI: automatically starts interactive session on login node

   Launch using CLI: sinteractive starts interactive session on compute node

2. A “batch job”

   A request for a set of resources for a given time

   Managed using SLURM - resource management software

   Created using an .sbatch script

Breaking down an sbatch script

```
#!/bin/bash
#SBATCH --job-name=MATLAB_ex    # Specify name for job [do not use spaces]
#SBATCH --time=00:05:00         # Specify maximum time [Program will terminate after max time if not complete, default is 36:00:00]
#SBATCH --partition=ssd         # Specify partition [Should always be ssd]
#SBATCH --account=ssd           # Account [should always be set to ssd]
#SBATCH --nodes=1               # Number of nodes [Number of physical machines to request, determines other parameters]
#SBATCH --ntasks-per-node=4     # Number of tasks [Number of processing cores per node requested, 1=single threaded, 2+=multithreaded]
#SBATCH --cpus-per-task=1       # Number of threads per task [Should be one unless using MPI]
#SBATCH --mem=1gb               # Amount of RAM [May need to run interactive first to find correct number, DO NOT set too high]
#SBATCH --output=MAT_ex.out     # Name of output file [Can change jobname to suit whatever job you run]
#SBATCH --error=MAT_ex.err      # Name of error file [Can change jobname to suit whatever job you run]
```

TO RUN: `sbatch matlab_example.sbatch`

04

OK...What Next?

What more can I do with MidwaySSD?

Software on MidwaySSD

Programming Languages

Python, R, Matlab, Stata, Julia...

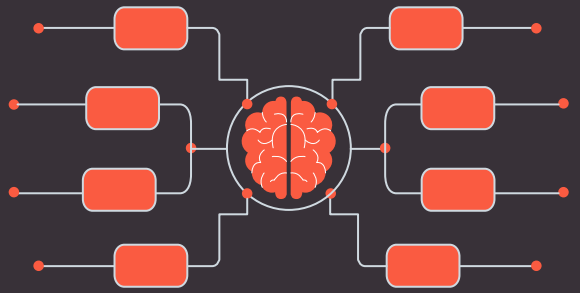
Neuroimaging software [coming soon]

AFNI, SPM, FSL

Deep Learning

PyTorch

Tensorflow



Parallelism and Optimization

CPU Parallelism

Python: Message Passing Interface (MPI4py)

MATLAB: Parallel Computing Toolbox (parfor)

GPU Parallelism

Python: pyOpenCL

MATLAB: gpuArray()

Questions?

Thank You!