Cluster Computing with ICC

James Joseph Balamuta

Departments of Informatics and Statistics
University of Illinois at Urbana-Champaign

December 07, 2018

On the Agenda

1. Cluster Computing
   - Motivation
   - Background
2. Getting Started with ICC
   - Background
   - Request Access
   - Connecting to ICC
3. ICC Structure
   - Overview
   - Storage
   - Software Modules on ICC
4. Appendix
   - SSH Tricks (Keys & Aliases)
   - Acknowledgements

Goals of the Talks

Talk 1: Overview of Cluster Computing

- Become Knowledgeable about ICC
- Connect into ICC
- Use Software

Talk 2: Overview of Scheduling Jobs

- Downloading and Uploading Data
- Writing a PBS Files
- Submitting and Checking Jobs

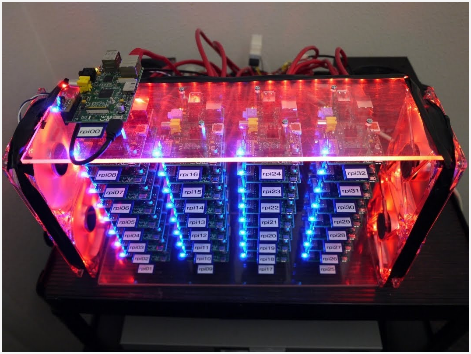
Overall Goal

Figure 1: Yubai taking a break from working with ICC

What is Cluster Computing?

Definition: Cluster

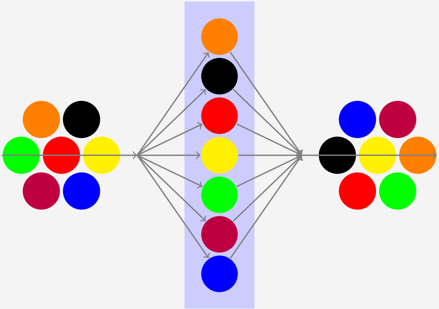
A **cluster** is a *set of computers* that are connected together and share resources as if they were one gigantic computer.



How Does Cluster Computing WorK?

Definition: Parallel Processing

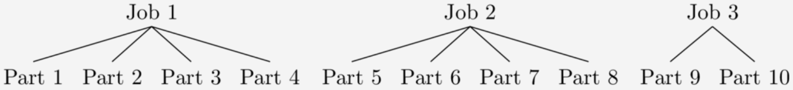
- **Parallel Processing** is the act of carrying out multiple tasks simultaneously to solve a problem.
- This is accomplished by dividing the problem into independent subparts, which are then solved concurrently.



Parallelization Realized

Definition: Jobs

- **Jobs** denote the independent subparts.



Why Should we use Cluster Computing?

Pros

- Speeds up simulations by allowing iterations to be run simultaneously.
- Provides more resources for computations.
  - e.g. CPU Cores, RAM, Hard Drive Space, and Graphics Cards (GPUs).
- Nightly snapshots/backups of files.
- Extends the lifespan of your computer.

Cons

- Simulations are **not** instantly run.
  - Need to “queue” for resources.
- Higher barrier of entry due to knowledge requirements.
- Poorly handles opening and closing data sets.
- Adding or updating software is complex.

Overview of Resources

Clusters at UIUC

- **Illinois Campus Cluster (ICC)**
  - Follows a time share model with a majority of departments buying in and is also usable for classes. *Try to use this option first*.
- **Keeling** (formerly **manabe**)
  - LAS machine for faculty & graduate students. Provides a stepping stone environment to ICC usage.
- **Biocluster**
  - Open to a majority of departments with preference to biology fields under a Research Computing as a Service (RCaaS) paradigm.
- **BlueWaters**
  - Expensive, but grants can be had if faculty are affiliated with NCSA. Requires two-factor authentication.

What is ICC?

Fast Facts

- Illinois Campus Cluster (ICC) is the public facing name to the underlying node arrangement called: Golub (deployed 2013).
- The cluster has over **300+ computing nodes** available for use.
- These nodes are managed by Torque Resource Manager, a form of OpenPBS, with the Moab Workload Manager.

Time-Share Model

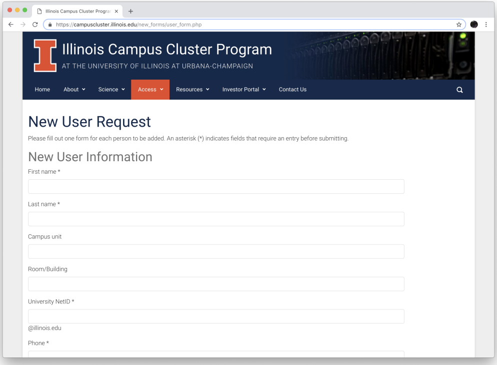
Management of nodes relate to two forms of queues for job submission:

- **Primary:** Settings specific to the investor.
- **Secondary:** Shared resource queue that allows access to any idle nodes in the cluster under specific limits (see queue slide).

Requesting an Account for ICC

Fill out the access request form.

`https://campuscluster.illinois.edu/new_forms/user_form.php`



Illinois Campus Cluster Program
https://campuscluster.illinois.edu/new_forms/user_form.php
Illinois Campus Cluster Program
AT THE UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Home
About
Science
Access
Resources
Investor Portal
Contact Us

New User Request

Please fill out one form for each person to be added. An asterisk (*) indicates fields that require an entry before submitting.

New User Information

First name *

Last name *

Campus unit

Room/Building

University NetID *

@illinois.edu

Phone *

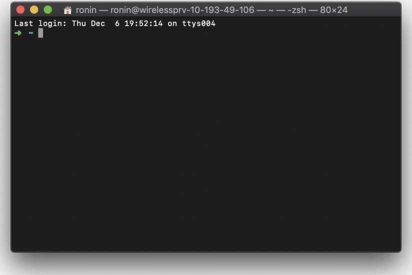
Talking to a Cluster

Definition: CLI

Command Line Interfaces (CLI) encourages interactions with a computer via user issued text-commands.

Terminal

- macOS and Linux have built in "shells" or "terminal" applications.

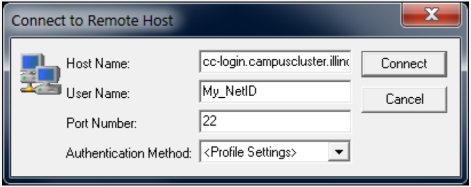


```
Last login: Thu Dec  6 19:52:14 on ttys004
➜ ~ 
```

Windows-Specific options

PuTTY

- If on Windows, then **download and install PuTTY**
- Setup a connection portal with:



Connect to Remote Host

Host Name: cc-login.campuscluster.illinc

User Name: My_NetID

Port Number: 22

Authentication Method: <Profile Settings>

Windows Subsystem Linux (WSL)

- Run a Linux terminal directly under a Windows 10 PC.
- Follow the **Installation guide**

Accessing ICC via CLI

Example login

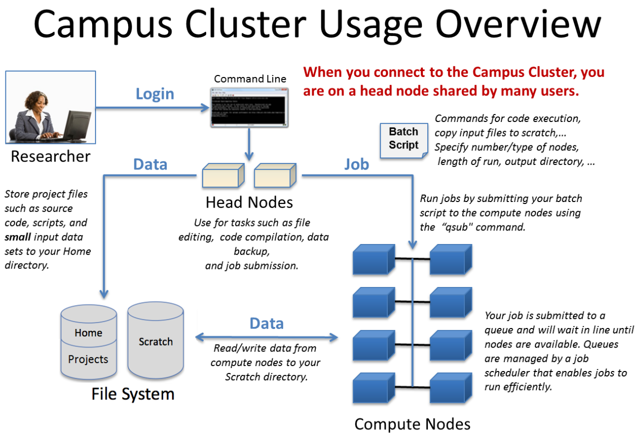
We first need to establish a connection to ICC to work on it. We can do this using **Secure Shell**, more commonly known as: `ssh`

```
ssh netid@cc-login.campuscluster.illinois.edu
# Enter password
```

How I would login. . .

```
ssh balamut2@cc-login.campuscluster.illinois.edu
# nottelling
```

Structure of ICC[1]



Campus Cluster Usage Overview

When you connect to the Campus Cluster, you are on a head node shared by many users.

Commands for code execution, copy input files to scratch,… Specify number/type of nodes, length of run, output directory, …

*Store project files such as source code, scripts, and **small** input data sets to your Home directory.*

Use for tasks such as file editing, code compilation, data backup, and job submission.

Run jobs by submitting your batch script to the compute nodes using the "qsub" command.

Read/write data from compute nodes to your Scratch directory.

Your job is submitted to a queue and will wait in line until nodes are available. Queues are managed by a job scheduler that enables jobs to run efficiently.

[1] Mirrors the academic model of professors distributing ideas to graduate students, waiting for them to solve the ideas, then aggregating the results into a paper. c.f. The Simpson S18E6 - Moe'N'A Lisa.

Queue Details[2]

Queue	Max Walltime	Max # of Nodes
test	5 minutes	2
secondary	4 hours	208
stat	336 hours	2

The `stat` queue has **14 nodes (~368 cores available)** structured as:

- 4: each with 128GB of memory & 16 cores (oldest)
- 4: each with 256GB of memory & 24 cores (older).
- 2: each with 256GB of memory, 24 cores, & 2 NVIDIA Tesla K80 GPUs (newer).
- 2: each with 384GB of memory & 40 cores (newest)
- 2: each with 384GB of memory, 40 cores, & 2 NVIDIA Tesla P100 16GB GPUs (newest)

[2] The *newest* 4 nodes are on order and should arrive soon (tm).

Storing Data & Code

Possible locations. . .

- Home Directory `~/`
  - Up to ~**2GB** (Soft cap[a]) / ~**4GB** (Hard cap[b]) with **nightly backups**.
  - Storage is **private**.
- Project Spaces `/projects/stat/shared/$USER`
  - ~**21TB** of shared space with **nightly backups**.
  - Storage is **shared** among `stat` members.
- Temporary Networked Storage `/scratch`
  - ~**10TB** of space purged after **30 days** with **no backup**.
  - Storage is **shared** among **all** ICC community members.

[a] Soft caps gently warn the user to lower their storage size.

[b] Hard caps prevent the user from adding new files.

Backups

Backup Info

- **Daily** night time backups.
- **30 days** of backups exist.
- **No off-site backups for disaster recovery.**

Location of Backups

- Home Directory `~/`

```
/gpfs/iccp/home/.snapshots/home_YYYYMMDD*/$USER
```

- Project Directory `/projects/stat/shared/$USER`

```
/gpfs/iccp/projects/stat/.snapshots/statistics_YYYYMMDD*
```

Software Modules

Module Files

Unlike a traditional desktop, you must load the different software that you wish to use into the environment via `modulefiles`. The list of supported software can be found on Software List or by typing:

```
module avail
```

Viewing, Retrieving, and Disabling Module Software

```
module list                # See active software modules
module load <software>     # Enable software
module unload <software>   # Disable software
module purge               # Removes all active modules
```

Working with Software

Latest Version of *R*

As of **November 2018**, the latest version of *R* on ICC is *R* **3.5.1**. *R* can be accessed by using[a]:

```
module load R/3.5.1 # Load software
```

[a] If the version is not specified during the load, e.g. `module load R`, then the oldest version of *R* will be used.

Ask for Help

ICC's help desk (via `help@campuscluster.illinois.edu`) can help install software on ICC. Please send them an e-mail and *CC* your advisor.

Writing a Custom Module

It is possible to compile and create your own modules. For details, see the tutorial A Modulefile Approach to Compiling *R* on a Cluster.

Setup R Package library[3] and Temporarily use *R* on ICC[4]

```
# Create a directory for your R packages
mkdir ~/project-stat/rlibs

# Load R version 3.5.1
module load R/3.5.1

# Set the R library environment variable (R_LIBS) to
# include your R package directory
export R_LIBS=~/project-stat/rlibs

# See the path
echo $R_LIBS
```

[3] This takes advantage of the `stat` project space (~21TB) instead of the home directory (~2gb) limit.

[4] Always load *R* via `module load`. Otherwise, *R* will **not** be available.

Permanently setup *R* home library

To ensure that the R_LIBS variable remains set even after logging out run the following command to permanently add it to the environment.[5]

```
cat <<EOF >> ~/.bashrc
  if [ -n $R_LIBS ]; then
      export R_LIBS=~/project-stat/rlibs:$R_LIBS
  else
      export R_LIBS=~/project-stat/rlibs
  fi
EOF
```

[5] The routine modifies the .bashrc file, which is loaded on startup.

Install *R* packages into home library

```
# Use the install.packages function to install your R package
$ Rscript -e "install.packages('devtools',
              '~/Rlibs', 'http://ftp.ussg.iu.edu/CRAN/')"

# Use devtools to install package
$ Rscript -e "devtools::install_github('coatless/visualize')"

# Devtools install from secret repo
$ Rscript -e "devtools::install_github('stat385/netid',
                                       subdir='secretpkg',
                                       auth_token = 'abc')"
```

- Watch the use of `'` and `"`!
- For `auth_token` obtain a **GitHub Personal Access Token**

Thanks! Robin's up next. . .

Speeding Up Access

Repetitively typing out login credentials is tedious:

```
ssh netid@cc-login.campuscluster.illinois.edu
# password
```

There are two tricks that void this and also make locally launched script jobs possible.

- Public/Private keys
    - Passwordless login
- SSH Config
    - Alias connection names

Public/Private Keys

Authentication with Keys

Instead of entering a password, the local computer can submit a private key to be verified by a server. This is a bit more secure and avoids the hassle of constantly typing passwords.

Generating an SSH Key

```
## Run:
ssh-keygen -t rsa -C "netid@illinois.edu"

## Respond to:
# Enter file in which to save the key (/home/demo/.ssh/id_rsa): # [Press enter]
# Enter passphrase (empty for no passphrase): # Write short password
```

Copy SSH Key to Server

```
## Run:
ssh-copy-id netid@cc-login.campuscluster.illinois.edu
```

SSH Config[6]

Setting up a Configuration

Add the following to `~/.ssh/config`[a]

```
Host icc
    HostName cc-login.campuscluster.illinois.edu
    User netid
```

[a] **Replace** `netid` with your netid.

[6] **Note:** This assumes a default location is used for the SSH key. If there is a custom SSH key location add `IdentityFile ~/.ssh/sshkeyname.key` after the `User` line.

Acknowledgements

- Special thanks to the ICC team for putting together a great user guide.