Introduction to Interactive HPC

Evan Bollig and Michael Milligan
August 1, 2017
What is Interactive Computing?

- Software GUIs
- Prototyping workflows
  - Design your workflow for a single node (multi-core) or small set of nodes
  - Discover and test new tools/concepts
  - Profile, optimize and debug
- Data Visualization
Tutorial Outline

• Batch vs. Interactive Computing @ MSI

• Getting Started with Interactive Resources
  o Connecting to MSI
  o Calming your Fears
  o Prototyping Workflows
  o Conscientious Computing

• Hands-on
  o MATLAB Parallel Jobs
  o IPython Notebooks

Watch out for the “Don’t Be a Jerk” clause!
# Batch vs. Interactive Computing

<table>
<thead>
<tr>
<th></th>
<th>Batch (qsub)</th>
<th>Interactive SSH (qsub -I)</th>
<th>Interactive Desktop (Linux)</th>
<th>Interactive Desktop (Windows)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall clock limit</td>
<td>696 hours</td>
<td>696 hours$^{1,2}$</td>
<td>24 hours</td>
<td>48 hours</td>
</tr>
<tr>
<td>Requires SUs</td>
<td>✔</td>
<td>✔$^3$</td>
<td>✗$^4$</td>
<td>✗</td>
</tr>
<tr>
<td>Memory limit</td>
<td>1 TB</td>
<td>1 TB</td>
<td>16 GB</td>
<td>61 GB</td>
</tr>
<tr>
<td>Core Limit</td>
<td>8640</td>
<td>8640</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Software Modules</td>
<td>400+</td>
<td>400+</td>
<td>400+</td>
<td>50+</td>
</tr>
<tr>
<td>GPUs</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>GUIs</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

1. Don’t be a jerk    2. Larger requests receive lower priority    3. Resource dependent    4. Subject to change
Resource Allocations

ALL groups will have access to these basic resources:

- Service Units (SUs) for HPC usage
  - 70,000 (~100,000 CPU-hours) or more if requested
- High Performance Storage
  - 150 GB home directory quota or more if requested
- Access to interactive gateways
- Access to MSI labs
  - contact us to activate your Ucard for physical access
Batch
Batch Jobs

• When should you use Batch Jobs?
  o *Whenever possible!* This is the traditional way to work in HPC
  o “Don’t Be a Jerk”; share resources and be considerate of other researchers

• What are the benefits of Batch Jobs?
  o Headless execution of automated processes
  o Long runtimes
  o Large core counts
  o A scheduler packs jobs in hardware to maximize utilization, reduce latency, etc.
Batch Example

• The Job Script (hello.pbs): A BASH shell script containing #PBS directives and commands

```bash
#!/bin/bash
#PBS -l walltime=1:00:00,mem=12GB,nodes=1:ppn=4
#PBS -m abe
#PBS -M boll0107@umn.edu
#PBS -W umask=0007
#PBS -A bollige

echo "HELLOWORLD" >> ~boll0107/test_output.log
```

• Submit this to a queue on Mesabi, Itasca, or the Lab Queue*:
  qsub hello.pbs
Interactive Batch
Interactive SSH (qsub -I)

• ssh to login.msi.umn.edu, then lab (or mesabi, or itasca)
• qsub -I -lwalltime=$W, nodes=$N:ppn=$P
Interactive Remote Desktops

Details for each later
Linux VMs (NICE and NX)

- [http://nice.msi.umn.edu](http://nice.msi.umn.edu), and [nx.msi.umn.edu](http://nx.msi.umn.edu)
- VMs run on the Lab Queue (Also NOT a Supercomputer)
Windows VMs (Citrix Xen)

- [http://xen.msi.umn.edu](http://xen.msi.umn.edu)
- (Four load-balanced pools)*(Eight cores and 61 GB RAM) = NOT a Supercomputer!
Interactive Web Portals
Jupyter Notebooks Server

- [https://notebooks.msi.umn.edu](https://notebooks.msi.umn.edu)
- Access web-based environment providing Jupyter Notebook document-oriented computing
- Currently supporting Python 2, Python 3, R
- In-browser terminal and file browsing
- MSI Beta service
Galaxy Portal

- http://galaxy.msi.umn.edu
- Workflow manager, principally for ‘Omics and Informatics research

For more details on Galaxy, check out our other Tutorials.

e.g., Quality Control of Illumina Data with Galaxy
http://msi.umn.edu/tutorials
Recap: Interactive vs. Batch Computing

<table>
<thead>
<tr>
<th></th>
<th>Batch (qsub)</th>
<th>Interactive SSH (qsub -I)</th>
<th>Interactive Desktop (Linux)</th>
<th>Interactive Desktop (Windows)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall clock limit</td>
<td>696 hours</td>
<td>696 hours$^{1,2}$</td>
<td>24 hours</td>
<td>48 hours</td>
</tr>
<tr>
<td>Requires SUs</td>
<td>✔</td>
<td>✔$^3$</td>
<td>✗$^4$</td>
<td>✗</td>
</tr>
<tr>
<td>Memory limit</td>
<td>1 TB</td>
<td>1 TB</td>
<td>16 GB</td>
<td>61 GB</td>
</tr>
<tr>
<td>Core Limit</td>
<td>8640</td>
<td>8640</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Software Modules</td>
<td>400+</td>
<td>400+</td>
<td>400+</td>
<td>50+</td>
</tr>
<tr>
<td>GPUs</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>GUIs</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

1. Don’t be a jerk  2. Larger requests receive lower priority  3. Resource dependent  4. Subject to change
Choosing a Cluster

- Tiny jobs and desktops run on the Interactive Cluster (a.k.a. “lab”)
- Long running jobs and batch jobs run on Mesabi/Itasca
  - Shorter queues
  - Better hardware
  - Same software
  - “qsub -I” also supported
  = More research; Faster.
- Itasca requires ppn=8
- Mesabi compute nodes cannot connect to internet
- Mesabi compute nodes are heterogeneous (e.g., up to 1TB Memory, GPUs, SSDs, etc.)
Hands-On: Getting Connected
Pre-requisite: VPN (for Non-UMN Networks)

https://it.umn.edu/virtual-private-network-vmn
Windows Pre-requisite: SSH and File Transfer

https://winscp.net/eng/download.php

Download and install both WinSCP and PuTTY
Interactive Jobs with “qsub -I”
Selecting Resources

- SSH to any cluster headnode:
  - `ssh login.msi.umn.edu`
  - `ssh mesabi`

- Queue an Interactive (-I) job on any cluster:
  - `qsub -I [options]`
Selecting Resources

Options to QSUB

- `qsub -I [options]`
  - `-l walltime=W`
  - `-l nodes=X:ppn=Y`
  - `-l pmem=M` OR `-l mem=M`
  - `-q "QueueName"
  - `-A groupname`
  - `-l gres=MATLAB+4`

- Enable graphics via X-tunneling (`-x`)
Software Modules

- See all modules available
  - `module avail`
- Load a module (adds commands to your shell)
  - `module load matlab`
- Run the software from the module
  - `matlab`
Build a PBS Script Interactively

• Use a terminal to complete your work
  o Automate the process
  o All commands go into a single BASH script (e.g., workflow.bash)

• Remember:
  o Errors will interrupt the script and fail your job (work around them)
  o Use relative File Paths in case script is moved (e.g., /home/bollige/boll0107/test.m -> ~/test.m -or- $HOME/test.m)
  o Use BASH variables whenever possible to generalize
Connecting to an Existing Session

- Use screen to run multiple terminals on the remote application
- Find the node name
  - `cat $PBS_NODEFILE`
  - `qstat -f $PBS_JOBID`
- Connect using SSH
  - `ssh login.msi.umn.edu`
  - `ssh mesabi`
  - `ssh [node]`
Simple SSH Keys

• Generate a new key
  o `ssh-keygen -t rsa -f ~/.ssh/id_rsa`

• Authorize the public key for SSH
  o Append `~/.ssh/id_rsa.pub` to `login.msi.umn.edu:~/.ssh/authorized_keys`

• Add the key to your Agent keychain
  o `ssh-add ~/.ssh/id_rsa`

• Login using the Agent for authentication
  o `ssh -AX login.msi.umn.edu`
Clean up your session

• Delete the session (choose one)
  o Exit out of all connections
  o qdel

• What happens on disconnect/delete?
  o All computation stops

• How to resume?
  o Start from zero; or load intermediate files (if you created them)
Interactive Linux Desktops

NICE and NX
Setup NX

- [https://www.msi.umn.edu/support/faq/how-do-i-get-started-nx](https://www.msi.umn.edu/support/faq/how-do-i-get-started-nx)
  - Install specific client for your system; required to connect
- Create new connection
  - NX protocol to “nx.msi.umn.edu” port 4000
  - Password or Private Key are OK
  - Don’t use a proxy
  - Save
- Open Connection
  - Authenticate
  - Create a new Desktop
NX Restrictions

- NO GRAPHICS ACCELERATION
- Do not run jobs on NX nodes. You must “isub” or “ssh” to another cluster (e.g., “ssh mesabi”)
  - No modules are available on these nodes
- NX nodes are shared by all NX Clients. Don’t Be a Jerk.
  - Do not use more than one core
  - Do not use more than 1G of memory
- Remember to Logout
  - Sessions persist on disconnect
  - Sessions persist without formal time limit (sessions get cleaned up by routine MSI maintenance).
Setup NICE

- Login to [http://nice.msi.umn.edu](http://nice.msi.umn.edu)
- Click on the "**Download DCV Client**" Link
  - DCV Client is a specialized VNC Client; required to connect
- Choose your OS (Windows, OSX, Linux)
- Install the application
- Start new session
NICE Queue Types

• **Non-GPU** - Regular, non-graphics accelerated sessions. These are sufficient for most applications.

• **GPU** – Graphics accelerated sessions for **Interactive Visualization**. Do not use these unless you require a GPU!
  
  ○ Don’t Be a Jerk
Copy Data In/Out

• The Linux VM runs out of Panasas (i.e., Group Home directories)
  o DCV Client has a built-in file transfer (Look for )
  o Use SCP, SFTP, etc. to transfer data to/from Panasas:
    `scp -r localDir login.msi.umn.edu:~/remoteDir/`.
  o Move data in/out from Windows using Windows FileExplorer

• Leverage `/scratch.global` and `/scratch.local` when possible
  o NO BACKUP!!
  o Cleanup after yourself
Session Clean-up and Resume

• What is lost when the connection drops?
  o Instance: if not expired, reconnect to the same Instance
  o Data: no loss

• What if you Logout?
  o Instance: Terminated; Reconnect and start fresh
  o Data: Scratch data is deleted periodically* (clean up after yourself to be sure); No loss for data in group home

• How do you reconnect?
  o Login again at http://nice.msi.umn.edu or via the NX Client
Interactive Windows Desktops

Following

https://www.msi.umn.edu/support/faq/how-do-i-connect-citrix-windows-virtual-machine
Install Citrix Client

• Login to http://xen.msi.umn.edu

• Follow the on-screen guide to install the Citrix Receiver (if not yet installed)
  o Plugin available for Windows, Linux (RedHat/CentOS, Debian/Ubuntu), and MacOSX operating systems
VM Types

- **Default** -- A standard Windows session with 2D graphics.
  - More than adequate for most user needs.
- **Node-locked Software** -- Designated titles like GenomeStudio with restricted licenses.
- **GPU** -- A shared GPU for multiple users on the same server.
  - For applications with low to moderate graphics requirements.
- **GPU Exclusive** -- A single GPU for each user; access is granted by request ONLY (help@msi.umn.edu).
Access your Data

- Panasas Group Directories are mounted (G:)
  - Drag-and-drop data to VM
  - Stage to Local Scratch (L:) for best I/O speeds

- Other File Transfers
  - Use WinSCP, Firefox/Chrome within the VM
  - Or allow Citrix to mount your local drives Read-Only or Read/Write and Drag-and-drop within Windows FileExplorer
Clean-Up and Resume Work

• What is lost when the connection drops?
  o Instance: if not expired, reconnect to the same Instance Type
  o Data: Scratch data (L:) is kept until the instance expires or it is terminated

• What if you Logout?
  o Instance: Terminated; Reconnect and start fresh
  o Data: Scratch data (L:) is deleted; User data (U:) is kept

• How do you reconnect?
  o Login again to http://xen.msi.umn.edu
Application Case Studies
Hands-On: Driving MATLAB Simulations On Itasca
Launch Matlab within NICE

- This demo requires a passphrase-less SSH key*
  - Do this only once (on an MSI system):
    - `ssh-keygen -t rsa -f ~/.ssh/id_rsa`
      (hit enter for all prompts)
    - `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`

- Open a Terminal (Menu -> System -> Terminal)
  - `cd $HOME`
    - `mkdir -p $HOME/Documents/MATLAB/`
    - `module load matlab/R2015a`
    - `matlab &`
Write a simple parallel script

- Create the file (run in Matlab):
  - `edit myfunc.m`

- Add Contents:
  - `parfor i = 1:2048`
    - `t = getCurrentTask();`
    - `A(i) = t.ID;`
    - `end`

  - `spmd`
    - `t = getCurrentTask();`
    - `R = rand(4);`
    - `c = labindex;`
    - `fprintf('LabIndex: %d, t.ID: %d', labindex, t.ID);`
    - `end`
Prototype in parallel on a ‘local’ cluster

- Create a local pool of 4 workers (Run in Matlab):
  - `cl=parcluster('local')`
  - `cl.parpool(4)`
- Do your parallel work
  - `myfunc`
  - `plot(A)`
- Clean up the local pool
  - `delete(gcp)`
Send Jobs to Mesabi/Itasca

- Setup the Cluster Configuration (Run in Matlab):
  - `cd ~ configCluster`
- Configure your ClusterInfo settings (i.e., Job Settings)
  - `ClusterInfo.state`
  - `ClusterInfo.setWallTime(‘1:00:00’)`
  - `ClusterInfo.setProcsPerNode(8)`
  - `ClusterInfo.setClusterHost(‘mesabi’)`
  - `ClusterInfo.setQueueName(‘small’)`
  - `ClusterInfo.setMemUsage(‘1gb’)`
Submit a Job

- Build a parcluster and submit your script
  - `cl = parcluster()`
  - `cl.batch('myfunc','Pool',15)

- One time only:
  - Enter your username
  - Specify your SSH *private* key (~/.ssh/id_rsa)
  - The key does not have a passphrase
Getting Results

• The job is submitted (Batch)
  o You can close MATLAB and return for your results later

• Wait for results
  o `cl.Jobs(1).wait();`
    
    `j1 = cl.Jobs(1).load();`
    `plot(j1.A)`

• Delete jobs and data
  o `cl.Jobs(1).delete();`

• Look for job data and diary logs in `~/MdcsDataLocation` and `~/Documents/MATLAB/MdcsDataLocation`
Additional Documentation

- Additional documentation for MATLAB Parallel Computing Toolbox (PCT) and Distributed Compute Server (MDCS):
  
Hands-On: Interactive Development and Data Interrogation with IPython
Start your Jupyter Session

- Visit [notebooks.msi.umn.edu](https://notebooks.msi.umn.edu)
- Select the default job type
- An IPython dashboard should open in your web browser
Load an Example

- Quick start by visiting [z.umn.edu/msipython](http://z.umn.edu/msipython)

- Fetch some example notebooks
  
  ```
  git clone https://github.com/mbmilligan/msi-ipython-nb-ex.git
  ```

- Browse to folder and open any *.ipynb
Tour of Notebook features

To import a notebook, drag the file onto the listing below or click here.

Start here
Additional Documentation

- Overview of IPython's architecture for parallel and distributed computing.
- Detailed discussion of IPython cluster controller and engines.
- Discussion of IPython magic commands
- Official IPython documentation
Questions?

These slides and the hands-on guide will be posted to http://msi.umn.edu (see the “MSI Tutorials” section)
.ssh/config

• Add this text to ~/.ssh/config:

  Host lab*
    ProxyCommand ssh -Aq login.msi.umn.edu nc %h 22

• SSH directly to lab (via login.msi.umn.edu)

• Use in combination with ssh-add (SSH Agent) and ssh
  -A to authenticate once and sign on everywhere