ACCESSING CCR’S INDUSTRY CLUSTER

June 2021
UB Center for Computational Research
Objectives

- Introduce basic concepts of high performance computing (HPC)
- Describe how to access the CCR industry cluster utilizing:
  - CCR OnDemand Portal
  - SSH Command Line Access
- Discuss Software Modules Available
- Explain how to submit a batch job
- Describe how to utilize Singularity Containers

source: UB Photo Database
BASIC HPC CONCEPTS
Some Basic HPC Terminology

- **Compute Node** – a physical self-contained compute unit consisting of an operating system (OS), processors, memory, storage and networking interfaces. Also known as a “blade”. Compute nodes can contain multiple processors.

- **Processor** – a computing chip that can contain multiple processing cores.

- **Core** – a single processing unit within a processor. These are what ultimately perform computations.
  - Cores on the same compute node must share the node resources (e.g. memory, storage, and networking).
  - Within software, cores are often referred to as cpus or processors.
Some Basic HPC Terminology

- **Node Rack** – a cabinet that stores a stack of interconnected nodes and other compute equipment.
- **Compute Cluster** – a set of interconnected racks.
- **Shared Memory Computing** – an HPC application where each core (all residing on the same node) uses a shared block of memory.
- **Distributed Memory Computing** – an HPC application where each core (possibly residing on different nodes) uses separate blocks of memory and work is coordinated using message passing.

Two Rows of Node Racks at UB CCR
Some Basic HPC Terminology

- **Thread** – short for “thread of execution”, a sequence of instructions that is run concurrently with other threads of the same process. Threads within a process can share memory.

- **Accelerator** – a compute device that a connected processor can use to offload certain computations.

![Diagram of a process with 2 threads running concurrently.](source: NVIDIA)

![Diagram showing offload of compute-intensive functions to a GPU.](source: wikipedia)
Batch Computing

- UB CCR’s compute resources are organized as a batch computing system
  - Users submit compute requests (jobs) to the scheduler software (SLURM)
  - The scheduler examines requested resources (cores, memory, walltime, etc.) and determines when jobs should run and on what nodes
  - When jobs finish early, “holes” form in the schedule. The scheduler “backfills” jobs to fill these holes, kind of like a game of Tetris.
Accessing the Industry Cluster

- UB CCR provides secure remote access to its resources via the front-end machine.
- Services protected by:
  - Hardware and software firewalls
  - Virtual Private Network for Off-Campus
  - Secure Shell (ssh) for login nodes
  - SSL for OnDemand portal
CCR OnDemand

- Connect to CCR OnDemand to:
  - Request compute nodes
  - Load software
  - Launch batch or interactive jobs
  - Monitor job status
  - Transfer and edit files
- This is the recommended way for users to access the cluster
OnDemand provides an integrated, single access point for all of your HPC resources.
Login to CCR OnDemand

- If off-campus, must first connect to VPN
  - UBIT provides “AnyConnect” software (see buffalo.edu/ubit)
  - CCR provides industry users with guest VPN accounts
- Point a web-browser to ondemand.ccr.buffalo.edu
  - Login using your UB CCR username, password, and one-time token
  - The password and one-time token are entered on the same line with no extra characters or spaces
  - One-time tokens are generated in authentication apps such as Duo or Google Authenticator
CCR OnDemand Dashboard

- **Files**
  - Launches “File Explorer”
  - Navigate storage
  - Transfer files to/from PC
  - Edit, Delete, Copy, & Rename files

- **Jobs**
  - Check status of active jobs
  - Compose and submit new job scripts (under development)

- **Clusters**
  - Command line shell terminal

- **Interactive Apps**
  - Remote Desktop Sessions (for GUI applications)
File Explorer

- Click the links on left to navigate between user home directory, and group project or scratch areas
- “Upload” – Transfer files from PC to CCR storage
- “Download” – Transfer files from CCR storage to PC
- To view/edit/rename, select the file then click on arrow to right of file name for menu of options
Active Jobs

- Toggle between ‘All Jobs’ and ‘Your Jobs’
- Toggle between ‘Industry Cluster’ and other clusters or ‘All Clusters’

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>User</th>
<th>Account</th>
<th>Time Used</th>
<th>Queue</th>
<th>Status</th>
<th>Cluster</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>7446814</td>
<td>w27-100</td>
<td>mtorres</td>
<td>bsa</td>
<td>00:00:00</td>
<td>industry</td>
<td>Queued</td>
<td>Industry Cluster</td>
<td></td>
</tr>
<tr>
<td>7446813</td>
<td>w27-50</td>
<td>mtorres</td>
<td>bsa</td>
<td>23:37:37</td>
<td>industry</td>
<td>Running</td>
<td>Industry Cluster</td>
<td></td>
</tr>
<tr>
<td>7449984</td>
<td>300</td>
<td>nishagen</td>
<td>ezurek</td>
<td>00:01:56</td>
<td>scavenger</td>
<td>Completed</td>
<td>Industry Cluster</td>
<td></td>
</tr>
<tr>
<td>7449985</td>
<td>300</td>
<td>nishagen</td>
<td>ezurek</td>
<td>00:01:56</td>
<td>scavenger</td>
<td>Completed</td>
<td>Industry Cluster</td>
<td></td>
</tr>
<tr>
<td>7449981</td>
<td>300</td>
<td>nishagen</td>
<td>ezurek</td>
<td>00:01:48</td>
<td>scavenger</td>
<td>Completed</td>
<td>Industry Cluster</td>
<td></td>
</tr>
</tbody>
</table>
Cluster Shell

- Linux terminal window opens in new browser tab
- Automatically places you in your home directory
- Use Linux commands to navigate to group project or scratch directory
- View/edit files
- Submit batch jobs and monitor your job status in the queue

Click here for Linux and Slurm Cheat Sheet
Remote Desktop – Industry Users

1. Click on Industry Cluster Desktop
2. Choose the duration of session (max 72 hours)
3. Enter email address if desired and click Launch
4. Wait for Session to Begin (Queued, Starting, Running)
5. When running, click “Launch Industry Cluster Desktop”
Remote Desktop

Remote Desktop allows access to the clipboard, which is useful for copying/pasting text to/from the session and the PC. It also launches a shell prompt which can be used to load and launch software.
USING SECURE SHELL (SSH)
Command Line Cluster Access

- Command line access to front end login nodes is accomplished using Secure Shell (SSH)
- Alternative to using OnDemand portal
- Users must upload SSH public key to their CCR account
- Use SSH keys to login; no passwords accepted
- Linux and MacOS users can use ‘terminal’ for ssh
- Connection info:
  
  ```
  ssh -i mykey.pub CCRusername@vortex.ccr.buffalo.edu
  ```

  The -i option specifies the SSH public key you want to use

* REMINDER:* You must be connected to UB’s VPN first!
MobaXterm for Windows

• We recommend MobaXterm
• Commercial users may need to purchase but it’s worth the small cost
• Alternatives include putty and git bash
• We provide documentation for generating SSH keys and connecting to CCR using MobaXterm
Configure MobaXterm

Sessions → New Session
Configure MobaXterm – Step 1

Select SSH Session

Remote host: [Input Field]
Specify username: [Input Field]
Port: 22
Configure SSH Session:

- Host: vortex.ccr.buffalo.edu
- Username: your CCR username
- Port: 22
- Use private key: specify location of your private key
- Name: Name your session – i.e. CCR or vortex
Launch MobaXterm and double click on your session

If your SSH key works, you will be logged in and put in your home directory: /user/YourUsername
After a slight delay, the RStudio application will be launched in a new Xterm window.
MobaXterm Example – Editing Files

• Navigate to desired file using the File Explorer

• Double-click on file or right-click and choose “Open with default text editor”
SOFTWARE MODULES
Software Modules

• CCR supports numerous software packages
• Installed as “modules”
• Important module commands:
  - module avail – what’s installed?
  - module load – load module
• Industry cluster users only have access to software without license restrictions or licenses supplied by the company
• Example (try from MobaXterm or CCR OnDemand cluster shell app):
  
  ```bash
  $ rstudio
  -bash: rstudio: command not found
  $ module load R
  $ rstudio
  (successfully launches rstudio)
  ```
Software Modules

CCR has experience with dozens of software modules including:

- Engineering – ABAQUS, ANSYS, COMSOL, STAR-CCM
- Machine Learning – Tensorflow, Torch
- GPU Programming – CUDA, OpenCL
- Data Analytics – R, SAS
- Molecular Dynamics – NAMD, Rosetta, Schrodinger
- Quantum Chemistry – Orca, Q-Chem
- Programming – MATLAB, Python
- Bioinformatic/ Genomics – BLAST, GATK

Require something not listed above? Let us know what you use we can look into it!
UB CCR: SUBMITTING BATCH JOBS
About the Scheduler

• SLURM
  o Simple Linux Utility for Resource Management

• Useful scheduler commands (issue these from a MobaXterm shell prompt or an On Demand “Clusters shell prompt):
  o **sbatch** – submit a job script
  o **squeue/sqstat** – check the status of a job
  o **scancel** – delete a job
  o **snodes** – show node info and status
  o **sranks** – show job priorities
  o **stimes** – show when jobs are expected to start
Decomposing An Example Job Script

Job scripts begin with a list of SBATCH directives. These directives inform the scheduler about the resources needed for the job along with who to contact when the job completes and what to do with output that would normally be printed on the screen.

```
#!/bin/sh
#SBATCH --partition=debug
#SBATCH --time=00:15:00
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=12
#SBATCH --mem=48000
#SBATCH --constraint=IB&CPU-E5645
#SBATCH --job-name="hello_test"
#SBATCH --output=test_mpi-debug-%j.out
#SBATCH --error=test_mpi-debug-%j.err
#SBATCH --mail-user=cdc@buffalo.edu
#SBATCH --mail-type=ALL
```

NOTE:
Industry cluster Slurm directives are:
--cluster=industry
--partition=industry
--qos=industry
Decomposing An Example Job Script

The next step in the job script is to configure the desired computing environment by loading required modules and setting relevant variables. We typically also print (echo) various useful SLURM variables, like the job id, the nodelist, and the working directory. These can come in handy if there is a need to troubleshoot a completed job.

```bash
echo "SLURM_JOB_ID="$SLURM_JOB_ID
echo "SLURM_JOB_NODELIST"=$SLURM_JOB_NODELIST
echo "SLURM_NNODES"=$SLURM_NNODES
echo "SLURM_TMPDIR"=$SLURM_TMPDIR
echo "working directory = "$SLURM_SUBMIT_DIR

echo "****************************"
module load intel/17.0
module load intel-mpi/2017.0.1
module list
ulimit -s unlimited
```
Decomposing An Example Job Script

The final step in the job script is to launch the desired application. How an application is launched depends on the application. `srun` is the preferred launcher for multi-node applications. For single node applications, just launch the application directly (e.g. `python ./helloworld`). Most applications also require various command line arguments.

CCR provides example job scripts for various applications. These are located in the `/util/academic/slurm-scripts` folder. You can copy the example script of your choice and edit to suit your needs.
An Example Python Multiprocessing Job Script

This script requests:
1 node, 12 processors, and 48 GB RAM

It loads the python/anaconda module and then launches a python script. The pi_mp.py script contains some Python code that takes advantage of the multiprocessing module to use the 12 requested processors.

```
#!/bin/bash
#SBATCH --job-name=pi_mp
#SBATCH --output=pi_mp.out
#SBATCH --error=pi_mp.err
#SBATCH --mail-user=your_user_name@buffalo.edu
#SBATCH --mail-type=END
#SBATCH --time=00:10:00
#SBATCH --nodes=1
#SBATCH --cpus-per-task=1
#SBATCH --exclusive
#SBATCH --partition=general-compute
#SBATCH --constraint=CPU-E5645
#SBATCH --mem=48000
#SBATCH --tasks-per-node=12
module load python/anaconda
ulimit -s unlimited
python pi_mp.py
```

NOTE:
Industry cluster Slurm directives are:
--cluster=industry
--partition=industry
--qos=industry
An Example MPI for Python Job Script

```bash
#!/bin/bash
#SBATCH --job-name=mpi4py
#SBATCH --output=mpi4py.out
#SBATCH --error=mpi4py.err
#SBATCH --mail-user=your_user_name@buffalo.edu
#SBATCH --mail-type=END
#SBATCH --time=00:10:00
#SBATCH --nodes=2
#SBATCH --cpus-per-task=1
#SBATCH --exclusive
#SBATCH --partition=general-compute
#SBATCH --constraint=18&CPU-E5645
#SBATCH --mem=48000
#SBATCH --tasks-per-node=12

# load modules
module load python/anaconda
module load intel-mpi
ulimit -s unlimited

# enable mpi4py module
export PYTHONPATH=/util/academic/python/mpi4py/v2.0.0/lib/python2.7/site-packages:$PYTHONPATH

# launch app
export I_MPI_FABRICS_LIST=tcp
export I_MPI_DEBUG=4
export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so
srun -n $SLURM_NPROCS python mpi4py_pi.py
```

This script requests:
2 nodes, 24 processors, and 48 GB RAM per node

It loads the python/anaconda and intel-mpi modules and then adds the location of the mpi4py installation to the Python path. Finally, it sets up the intel-mpi environment and launches a python script using the "srun" launcher. `srun` is the preferred MPI launcher (instead of `mpirun` or `mpiexec`) on the CCR system.

NOTE:
Industry cluster Slurm directives are:
--cluster=industry
--partition=industry
--qos=industry
RUN JOBS USING SINGULARITY CONTAINERS
Singularity: What it is

- A container technology that allows you to pack entire workflows, software libraries and even data into a single image file.
- Containerized applications can be run as slurm jobs on CCR’s industry cluster.
  - Simply put the command line “singularity run image.sif arg1 arg2 …” inside your slurm script as you do with any other commands.

- Allows full control of installation of software without admin privileges on CCR’s cluster.
- Portable and stable
  - Can be run in any Linux based system
  - Immune to host machine software changes, i.e. no broken dependencies.

- Container image needs to be built using a local machine with admin privileges.

- If you already have your applications packed in singularity containers, they are ready to run on CCR’s industry cluster.
Singularity: An example Slurm script

```bash
#!/bin/sh
#SBATCH --cluster=industry
#SBATCH --partition=industry --qos=industry
#SBATCH --time=24:00:00
#SBATCH --nodes=1
#SBATCH --mem-per-cpu=4000
#SBATCH --ntasks-per-node=12
#SBATCH --job-name="yourjob"
#SBATCH --output="your.log"
#SBATCH --mail-user=your-name@your-company.com
#SBATCH --mail-type=ALL
#SBATCH --account=youraccount
#SBATCH --constraint=IB

echo "SLURM_JOBID="$SLURM_JOBID
echo "SLURM_JOB_NODELIST"=$SLURM_JOB_NODELIST
echo "SLURM_NNODES"=$SLURM_NNODES
echo "SLURM_NNODES"=$SLURM_NNODES
echo "SLURM_TMPDIR"="$SLURM_TMPDIR"

ulimit -s unlimited

singularity run --B /projects/industry/yourcompany:/mnt /projects/industry/yourcompany/your-application.sif \
/mnt/yourcompany/your/input/file /mnt/yourcompany/your/outputfile
```

This script requests the necessary resources and then launches a singularity job with “your-application.sif” as the container name. It takes an input and output file as the command line arguments. Note: In order for the container to have access to the host file system, we mounted the /projects/industry/your company directory to the container’s /mnt directory. The “–B” switch is used to specify the mount point.
Need Additional Technical Help?
• Please submit a ticket to:
  ccr-help@buffalo.edu

Want to learn how your company can access UB CCR’s industry cluster?
• Contact UB CCR’s Industry Outreach lead:
  Adrian Levesque, MBA at:
  apl3@buffalo.edu