# ACCESSING CCR'S INDUSTRY CLUSTER

June 2022

UB Center for Computational Research

University at Buffalo The State University of New York

# OVERVIEW

# Objectives

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

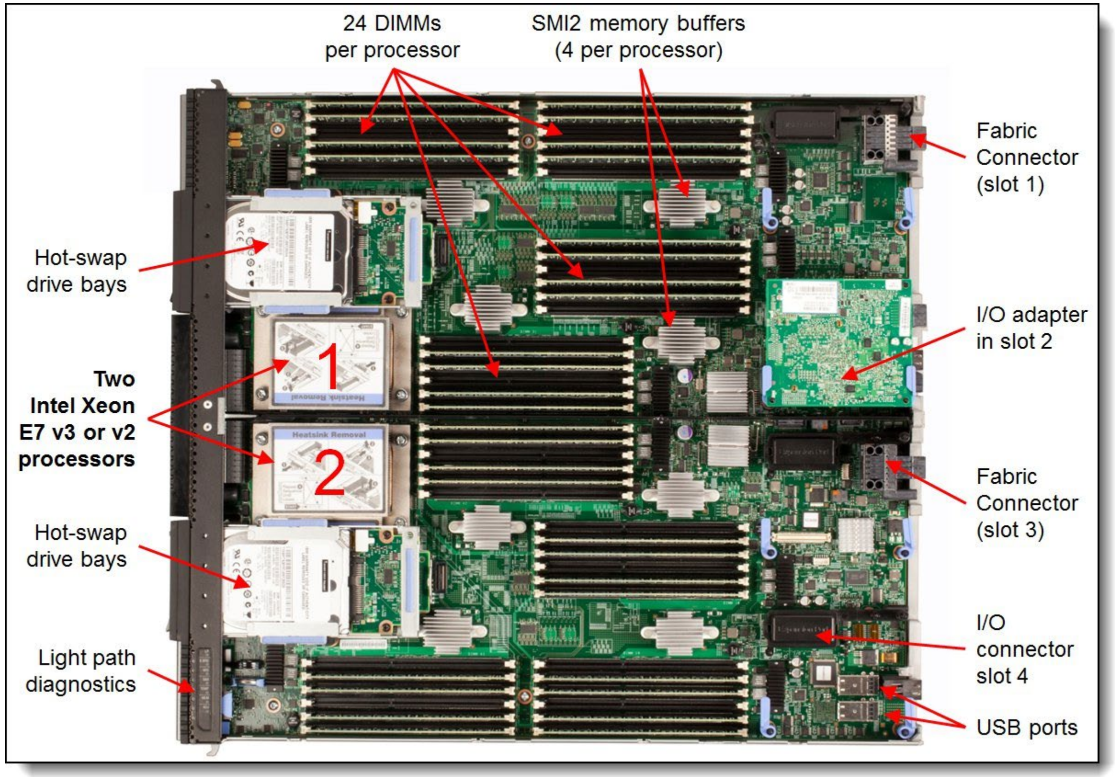
*source: UB Photo Database*

# 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

Anatomy of a compute node



*Source: lenovopress.com*

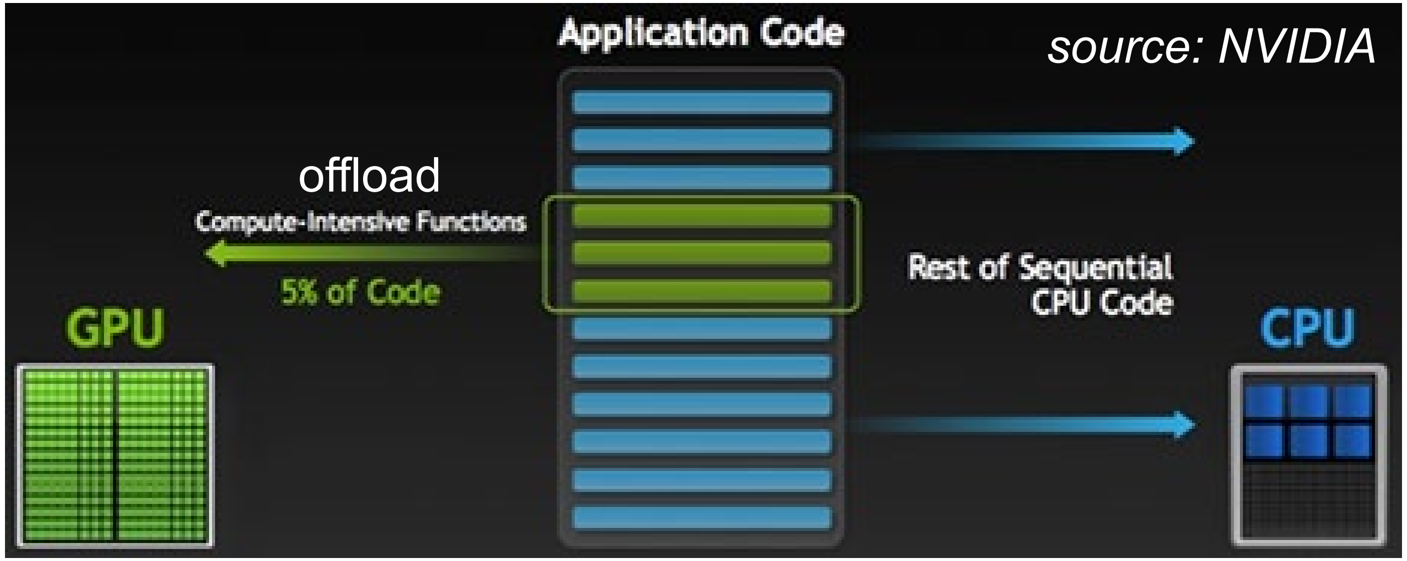
# Some Basic HPC Terminology

- *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.

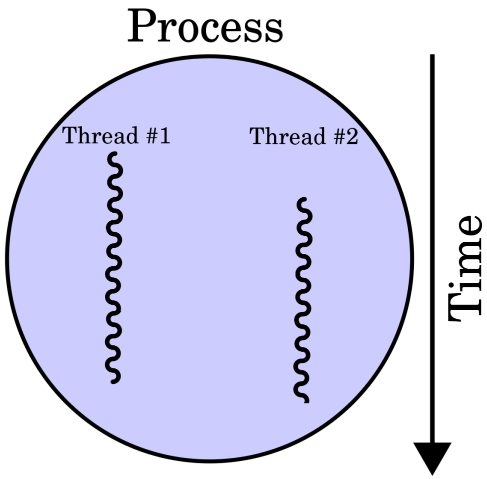
Two Rows of Node Racks at UB CCR

## 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.



source: NVIDIA

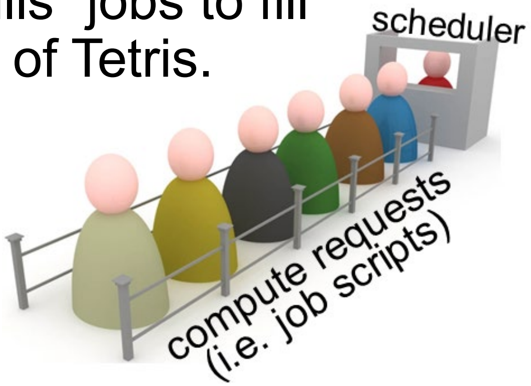


A process with 2 threads running concurrently.

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.

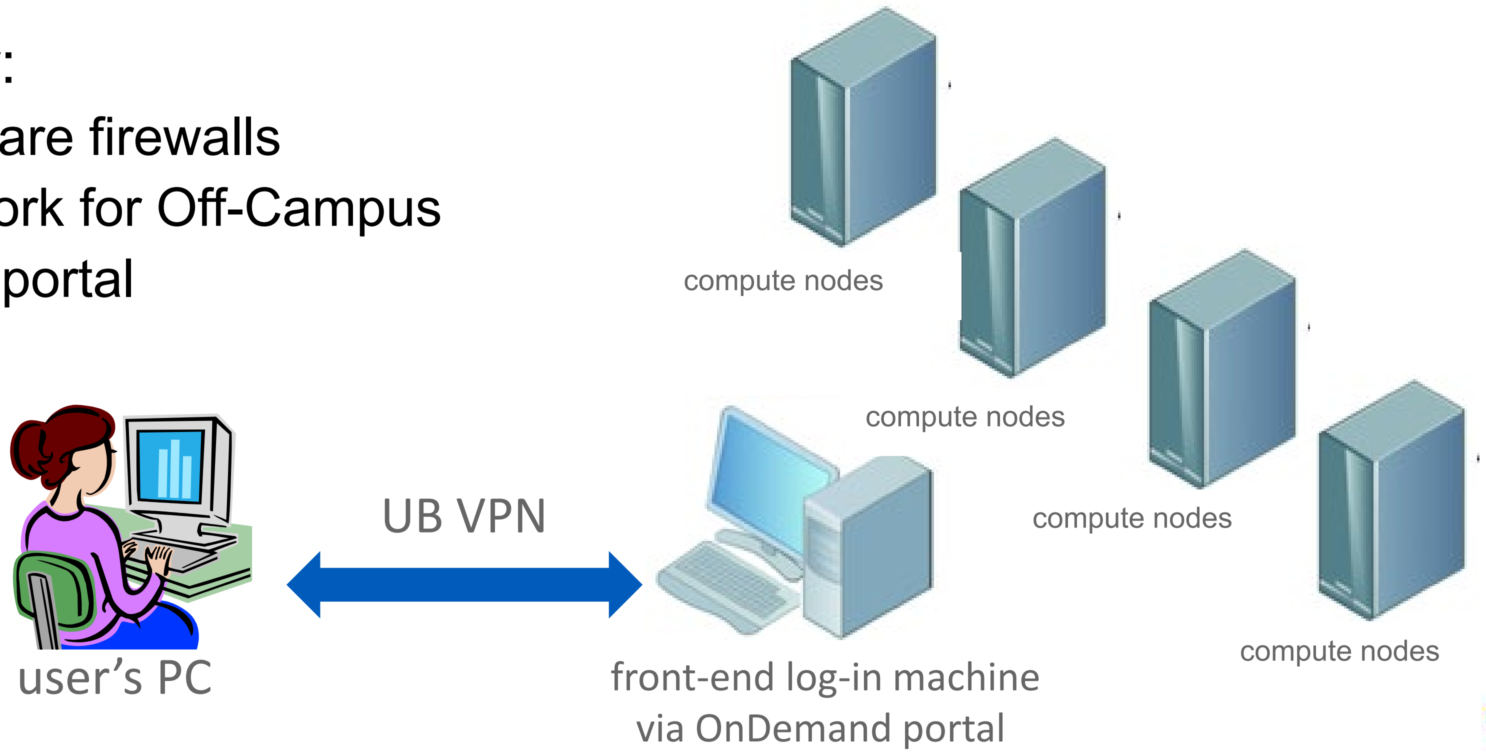


*Scheduling compute jobs can be likened to a game of Tetris*

Walltime
Cores

# Accessing the UB-HPC Cluster & Industry Compute Partition

- 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
  - SSL for OnDemand portal



compute nodes

compute nodes

compute nodes

compute nodes

UB VPN

user's PC

front-end log-in machine
via OnDemand portal

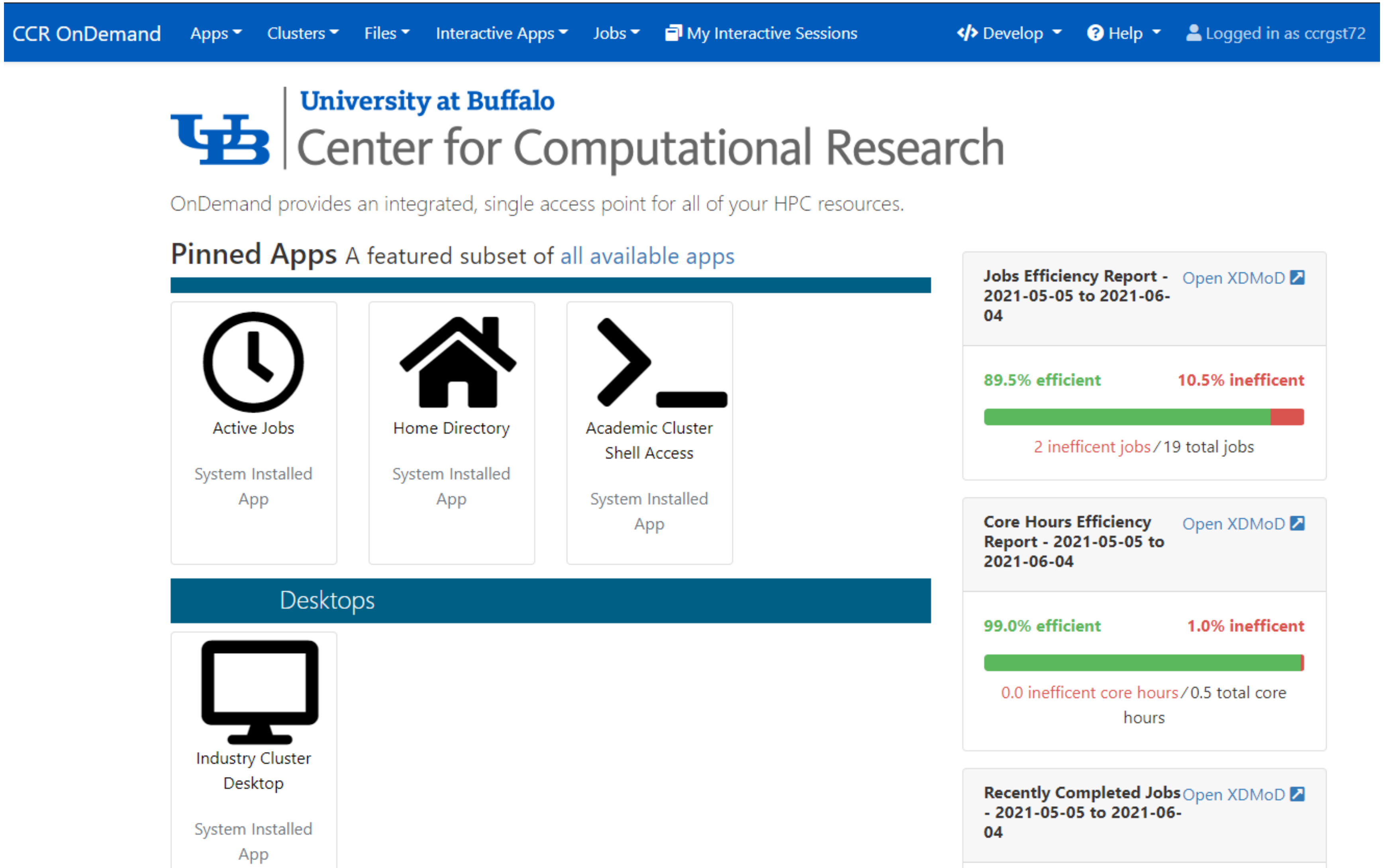
University at Buffalo The State University of New York

# USING CCR ONDEMAND

# 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

# CCR OnDemand Dashboard



University at Buffalo
Center for Computational Research

OnDemand provides an integrated, single access point for all of your HPC resources.

Pinned Apps A featured subset of all available apps

Jobs Efficiency Report - 2021-05-05 to 2021-06-04 Open XDMoD

89.5% efficient 10.5% inefficent

2 inefficent jobs / 19 total jobs

Core Hours Efficiency Report - 2021-05-05 to 2021-06-04 Open XDMoD

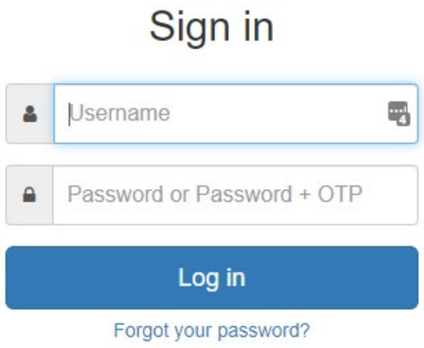
99.0% efficient 1.0% inefficent

0.0 inefficent core hours / 0.5 total core hours

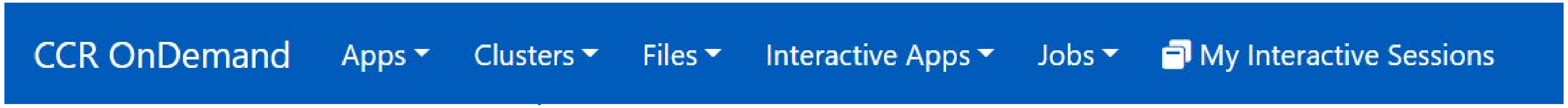
Recently Completed Jobs - 2021-05-05 to 2021-06-04 Open XDMoD

# Login to CCR OnDemand

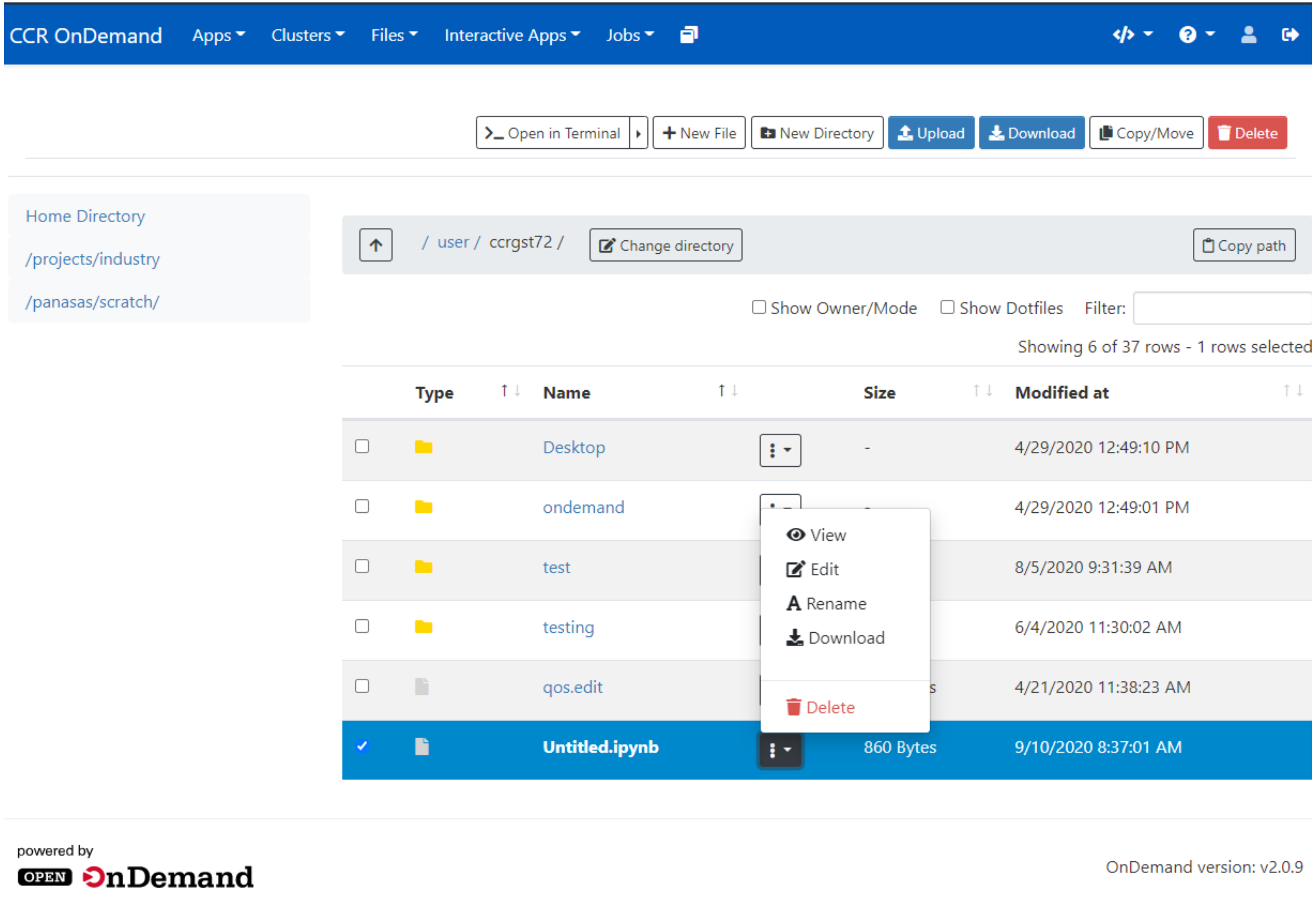
- If off-campus, must first connect to VPN
  - CCR provides industry users with VPN accounts
  - CCR will provide access to UBIT’s Cisco AnyConnect software
  - When connecting to the VPN, select the ‘CCR’ group from the drop down list
- Point a web-browser to ondemand.ccr.buffalo.edu
  - Login using 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)



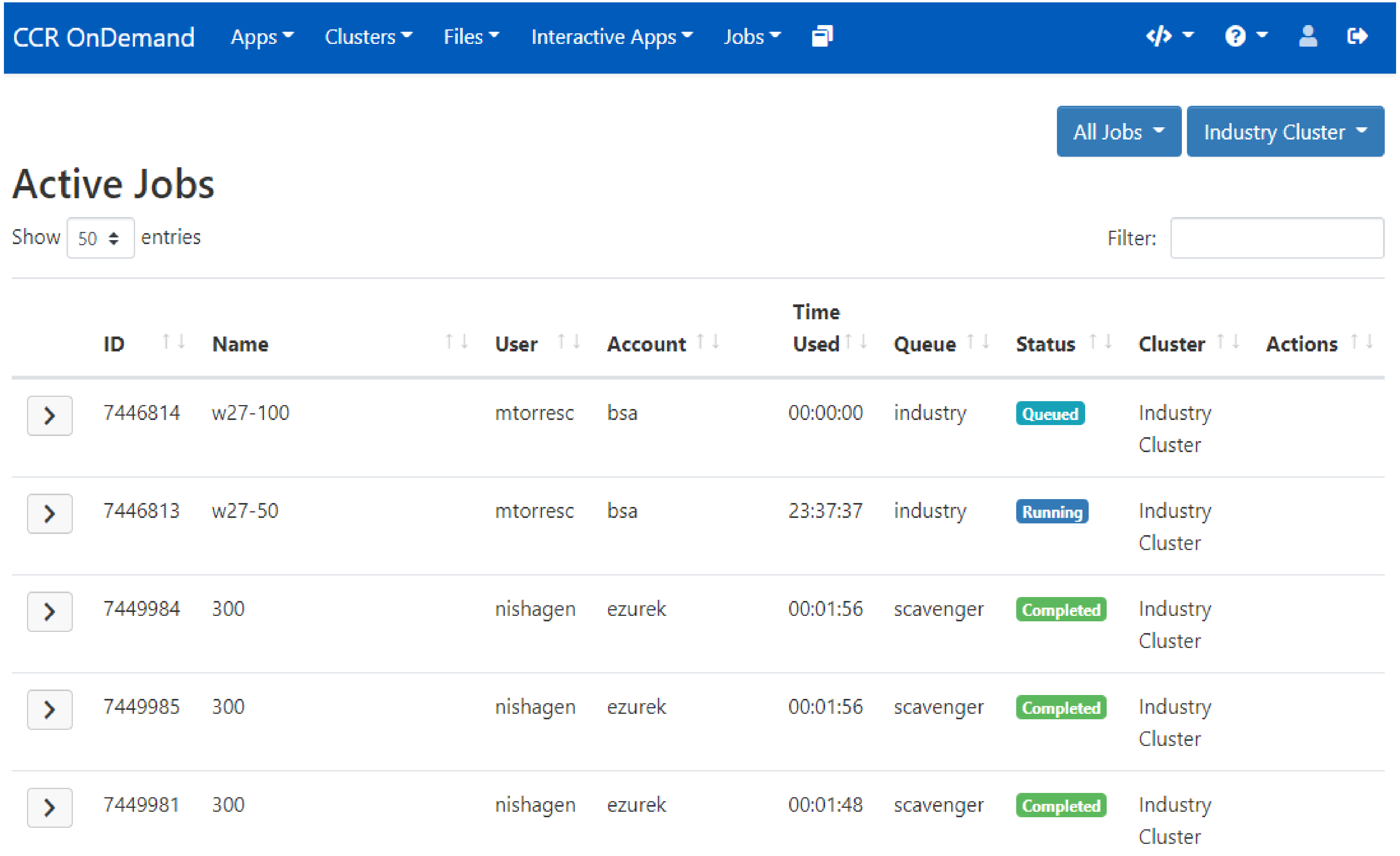
Showing 6 of 37 rows - 1 rows selected

|  | Type | Name | Size | Modified at |
|---|---|---|---|---|
| □ |  | Desktop | - | 4/29/2020 12:49:10 PM |
| □ |  | ondemand | - | 4/29/2020 12:49:01 PM |
| □ |  | test |  | 8/5/2020 9:31:39 AM |
| □ |  | testing |  | 6/4/2020 11:30:02 AM |
| □ |  | qos.edit | s | 4/21/2020 11:38:23 AM |
| ☑ |  | **Untitled.ipynb** | 860 Bytes | 9/10/2020 8:37:01 AM |

OnDemand version: v2.0.9

# 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

Show 50 entries

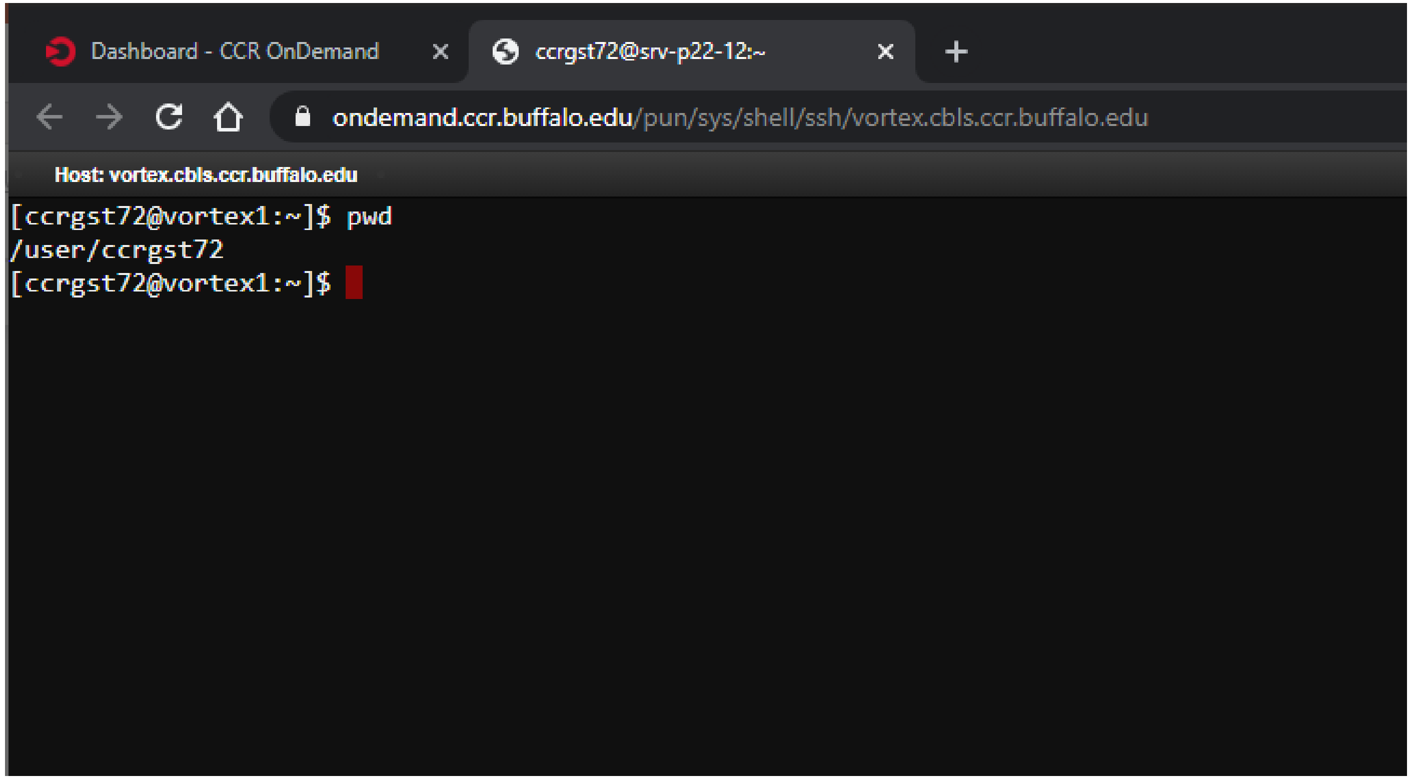
Filter:

| | ID | Name | User | Account | Time Used | Queue | Status | Cluster | Actions |
|---|---|---|---|---|---|---|---|---|---|
| > | 7446814 | w27-100 | mtorresc | bsa | 00:00:00 | industry | Queued | Industry Cluster | |
| > | 7446813 | w27-50 | mtorresc | bsa | 23:37:37 | industry | Running | Industry Cluster | |
| > | 7449984 | 300 | nishagen | ezurek | 00:01:56 | scavenger | Completed | Industry Cluster | |
| > | 7449985 | 300 | nishagen | ezurek | 00:01:56 | scavenger | Completed | Industry Cluster | |
| > | 7449981 | 300 | nishagen | ezurek | 00:01:48 | scavenger | Completed | Industry Cluster | |

# Active Jobs

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

# Cluster Shell


Dashboard - CCR OnDemand
ccrgst72@srv-p22-12:~
ondemand.ccr.buffalo.edu/pun/sys/shell/ssh/vortex.cbls.ccr.buffalo.edu
Host: vortex.cbls.ccr.buffalo.edu

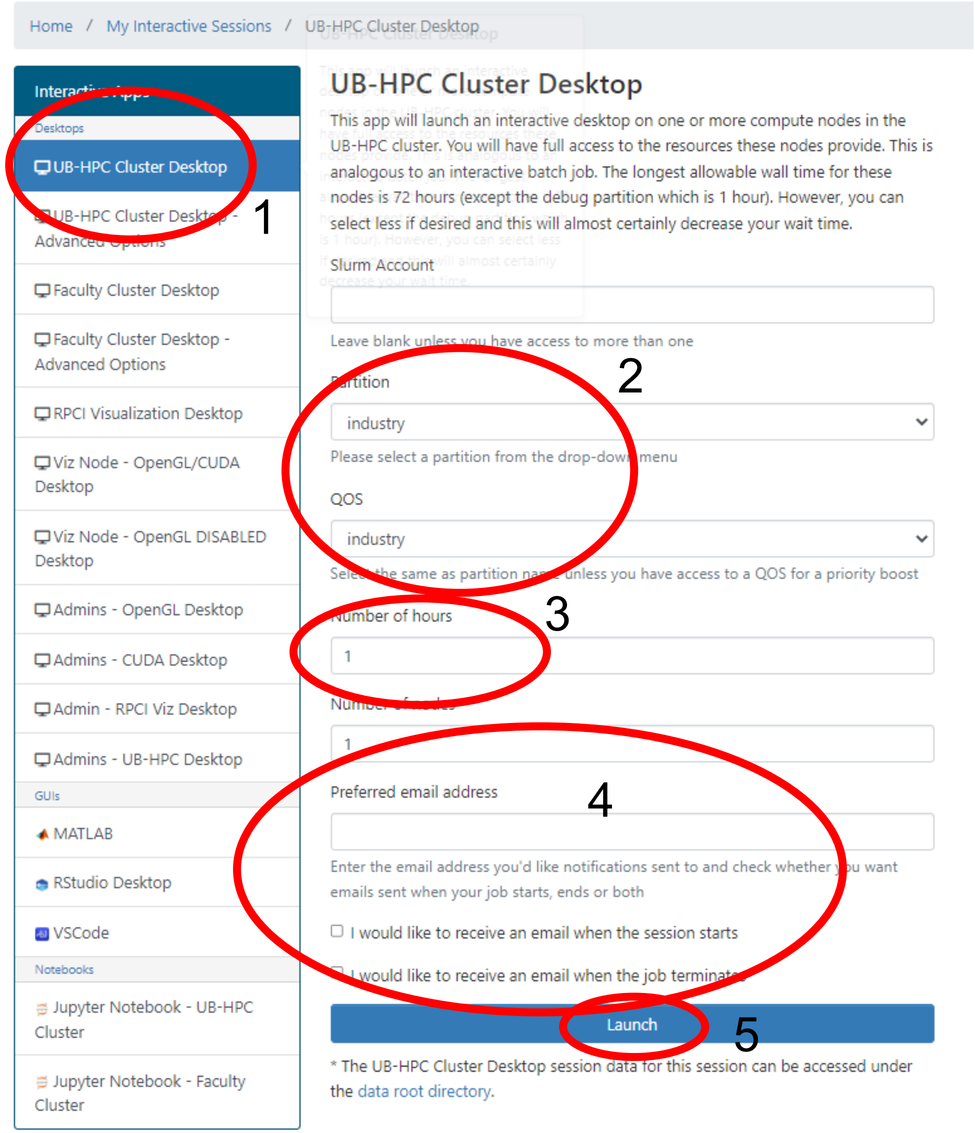
```
[ccrgst72@vortex1:~]$ pwd
/user/ccrgst72
[ccrgst72@vortex1:~]$
```

- 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

The Cluster Desktop apps in OnDemand provide a remote desktop style experience on the cluster nodes (servers). This is great for using GUI-based applications and running interactive code. If your application requires heavy graphics, please use the Viz Node application options.

1. Click on UB-HPC Cluster Desktop
2. Select the ‘industry’ option from the partition AND QOS drop down menus
3. Choose the duration of session (max 72 hours)
4. Enter email address if desired and select which types of notification emails you’d like to receive
5. Click the Launch button



Home / My Interactive Sessions / UB-HPC Cluster Desktop

Interactive Apps

Desktops

- UB-HPC Cluster Desktop
- UB-HPC Cluster Desktop - Advanced Options
- Faculty Cluster Desktop
- Faculty Cluster Desktop - Advanced Options
- RPCI Visualization Desktop
- Viz Node - OpenGL/CUDA Desktop
- Viz Node - OpenGL DISABLED Desktop
- Admins - OpenGL Desktop
- Admins - CUDA Desktop
- Admin - RPCI Viz Desktop
- Admins - UB-HPC Desktop

GUIs

- MATLAB
- RStudio Desktop
- VSCode

Notebooks

- Jupyter Notebook - UB-HPC Cluster
- Jupyter Notebook - Faculty Cluster

## UB-HPC Cluster Desktop

This app will launch an interactive desktop on one or more compute nodes in the UB-HPC cluster. You will have full access to the resources these nodes provide. This is analogous to an interactive batch job. The longest allowable wall time for these nodes is 72 hours (except the debug partition which is 1 hour). However, you can select less if desired and this will almost certainly decrease your wait time.

Slurm Account

Leave blank unless you have access to more than one

Partition: industry

Please select a partition from the drop-down menu

QOS: industry

Select the same as partition name unless you have access to a QOS for a priority boost

Number of hours: 1

Number of nodes: 1

Preferred email address

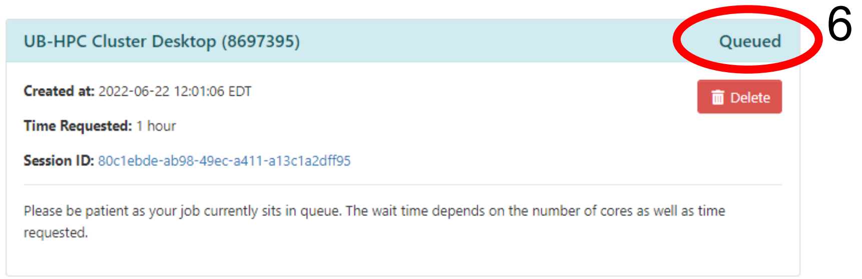
Enter the email address you'd like notifications sent to and check whether you want emails sent when your job starts, ends or both

□ I would like to receive an email when the session starts

□ I would like to receive an email when the job terminates

Launch

* The UB-HPC Cluster Desktop session data for this session can be accessed under the data root directory.



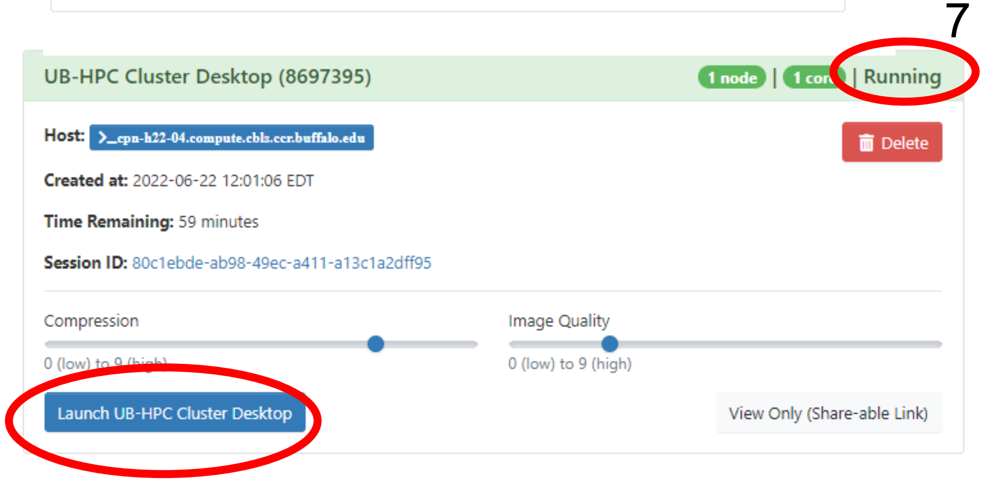
UB-HPC Cluster Desktop (8697395)

**Created at:** 2022-06-22 12:01:06 EDT

**Time Requested:** 1 hour

**Session ID:** 80c1ebde-ab98-49ec-a411-a13c1a2dff95

Please be patient as your job currently sits in queue. The wait time depends on the number of cores as well as time requested.



7
1 node | 1 core | Running
Delete
Launch UB-HPC Cluster Desktop
View Only (Share-able Link)

UB-HPC Cluster Desktop (8697395)

**Host:** >_cpn-h22-04.compute.cbls.ccr.buffalo.edu

**Created at:** 2022-06-22 12:01:06 EDT

**Time Remaining:** 59 minutes

**Session ID:** 80c1ebde-ab98-49ec-a411-a13c1a2dff95

Compression
0 (low) to 9 (high)

Image Quality
0 (low) to 9 (high)

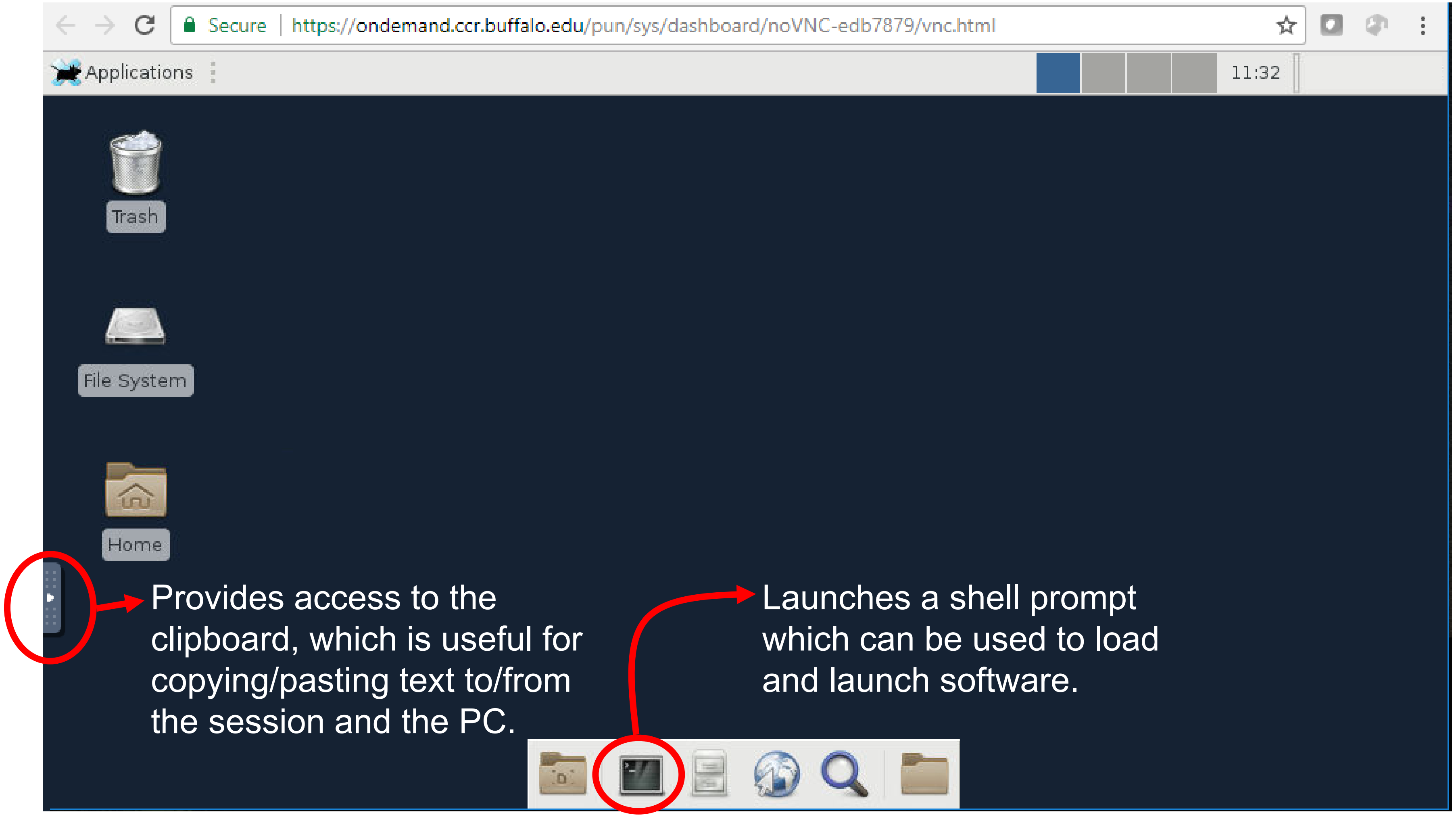
# Remote Desktop (con’t)

6. As the job prepares to start, the status will display ‘queued’
7. Once the job is running on the node, the status will display as ‘running’ and you can click on the ‘Launch UB-HPC Cluster Desktop” which launches the Linux desktop

Options:

- Host: Link opens up a terminal window into the node your job is running on
- View Only Link: This can be provided to CCR Help for assistance with troubleshooting problems
- Session ID: Clicking this link opens the File App and redirects you to the location of all the output files for this job. Here you’ll find .err and .out files containing useful troubleshooting info.
- Compression and Image Quality toggles: Use to adjust if you’re having trouble with the connection quality

# Remote Desktop

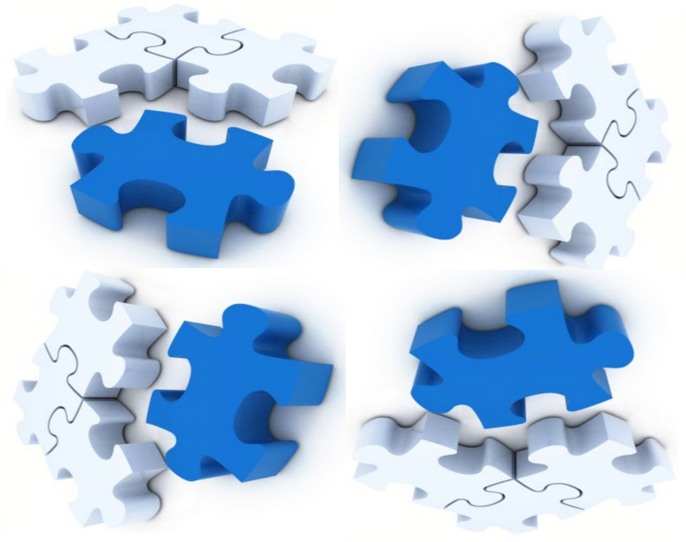

Secure
https://ondemand.ccr.buffalo.edu/pun/sys/dashboard/noVNC-edb7879/vnc.html
Applications
11:32
Trash
File System
Home

Provides access to the clipboard, which is useful for copying/pasting text to/from the session and the PC.

Launches a shell prompt which can be used to load and launch software.

# 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 (found in /util/common)
- Example (try from CCR OnDemand cluster shell app):

```
$ bedtools
bedtools: command not found
$ module load bedtools
$ bedtools
bedtools is a powerful toolset for genome arithmetic.
Version:   v2.27.1
```

# 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

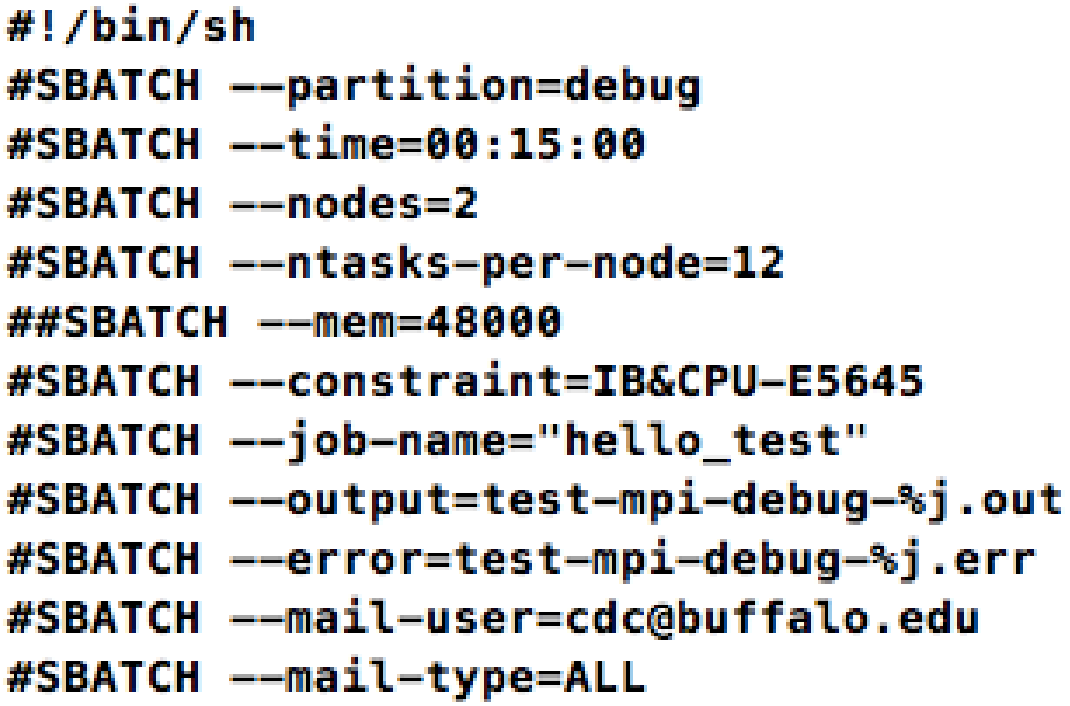
Require something not listed above?  Let us know what you use we can look into it!

# UB CCR: SUBMITTING BATCH JOBS

# About the Scheduler

- SLURM
  - Simple Linux Utility for Resource Management
- Useful scheduler commands (issue these from a MobaXterm shell prompt or an On Demand “Clusters shell prompt):
  - `sbatch` – submit a job script
  - `squeue`/`sqstat` – check the status of a job
  - `scancel` – delete a job
  - `snodes` – show node info and status
  - `sranks` – show job priorities
  - `stimes` – show when jobs are expected to start

# Decomposing An Example Job Script

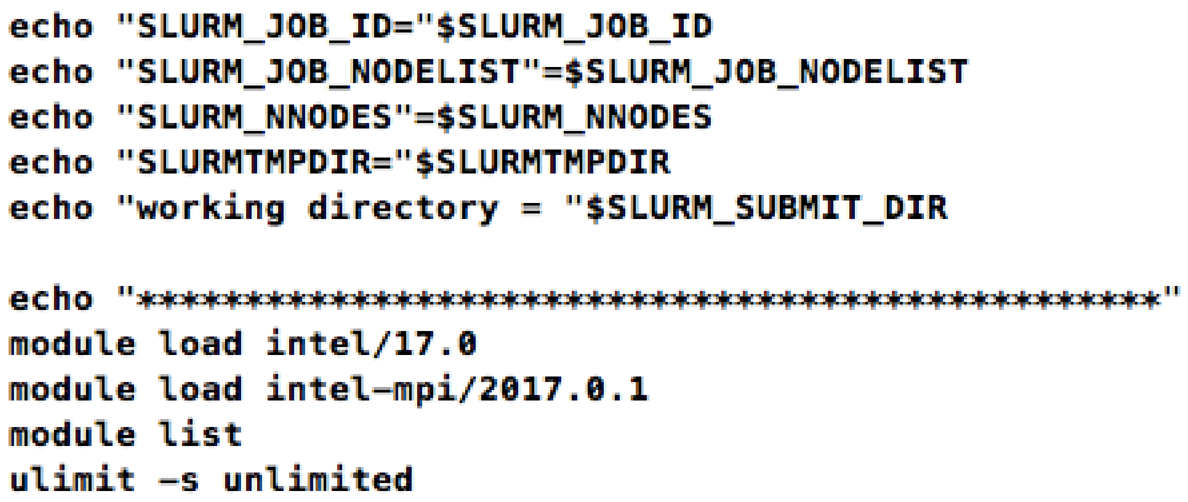


```
#!/bin/sh
#SBATCH --partition=debug
#SBATCH --time=00:15:00
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=12
##SBATCH --mem=48000
#SBATCH --constraint=IB&CPU-E5645
#SBATCH --job-name="hello_test"
#SBATCH --output=test-mpi-debug-%j.out
#SBATCH --error=test-mpi-debug-%j.err
#SBATCH --mail-user=cdc@buffalo.edu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

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.

NOTE:
Industry cluster Slurm directives are:
--cluster=ub-hpc
--partition=industry
--qos=industry

## Decomposing An Example Job Script



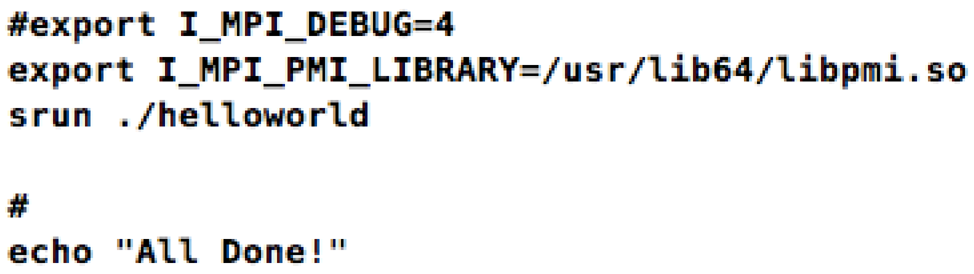
```
echo "SLURM_JOB_ID="$SLURM_JOB_ID
echo "SLURM_JOB_NODELIST"=$SLURM_JOB_NODELIST
echo "SLURM_NNODES"=$SLURM_NNODES
echo "SLURMTMPDIR="$SLURMTMPDIR
echo "working directory = "$SLURM_SUBMIT_DIR

echo "*********************************************************"
module load intel/17.0
module load intel-mpi/2017.0.1
module list
ulimit -s unlimited
```

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.

# Decomposing An Example Job Script



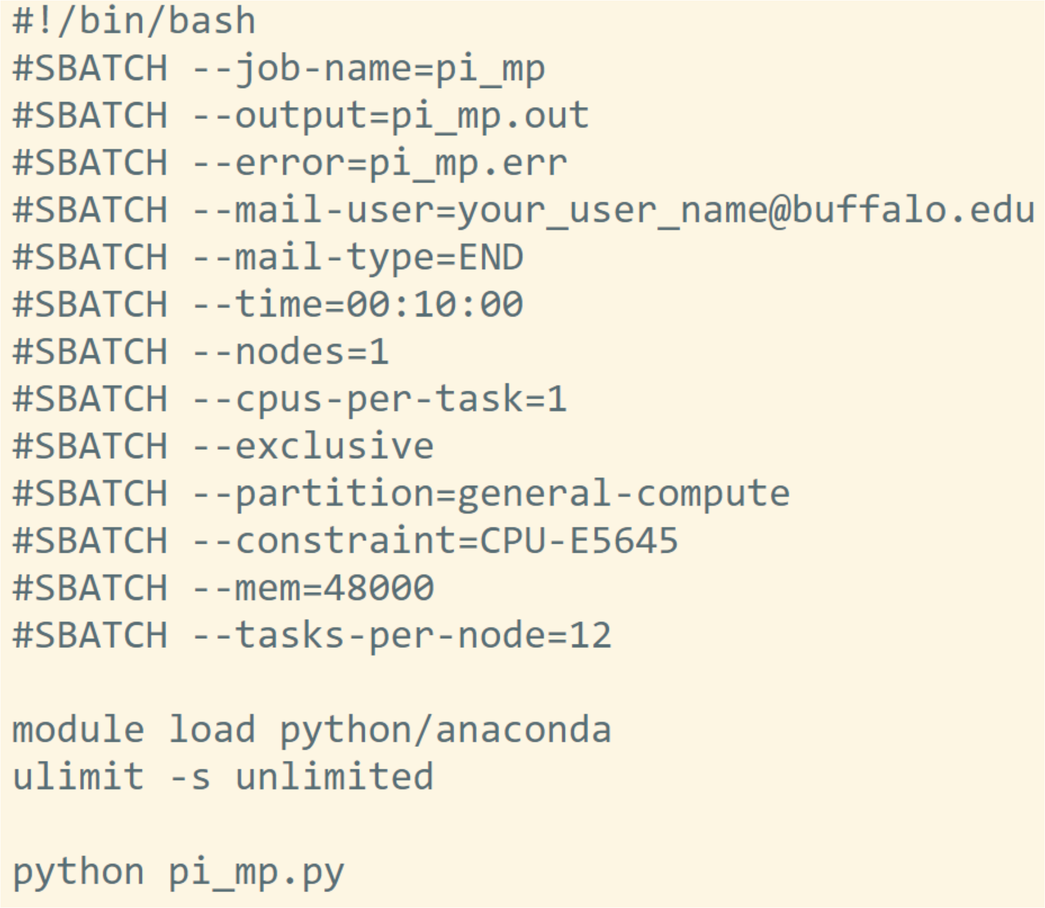
```
#export I_MPI_DEBUG=4
export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so
srun ./helloworld

#
echo "All Done!"
```

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



```
#!/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
```

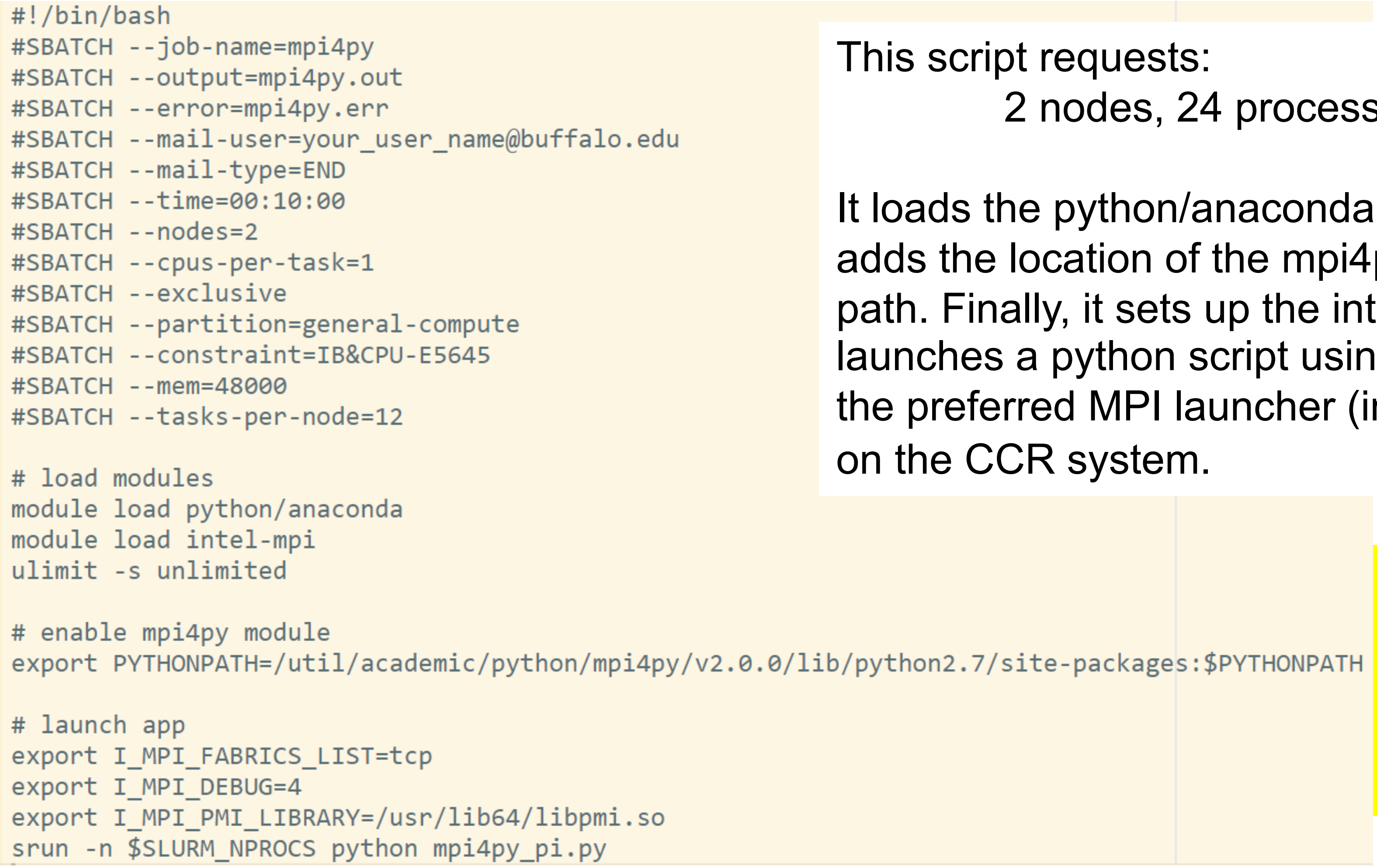
This script requests:

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.

NOTE:
Industry cluster Slurm directives are:
--cluster=ub-hpc
--partition=industry
--qos=industry

# An Example MPI for Python Job Script



```
#!/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=IB&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=ub-hpc
--partition=industry
--qos=industry

University at Buffalo The State University of New York

# RUN JOBS USING SINGULARITY CONTANERS

# 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

```
#!/bin/sh
#SBATCH --cluster=ub-hpc
#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"=$SLURM_NNODES
echo "SLURMTMPDIR="$SLURMTMPDIR
echo "working directory = "$SLURM_SUBMIT_DIR

ulimit -s unlimited

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

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