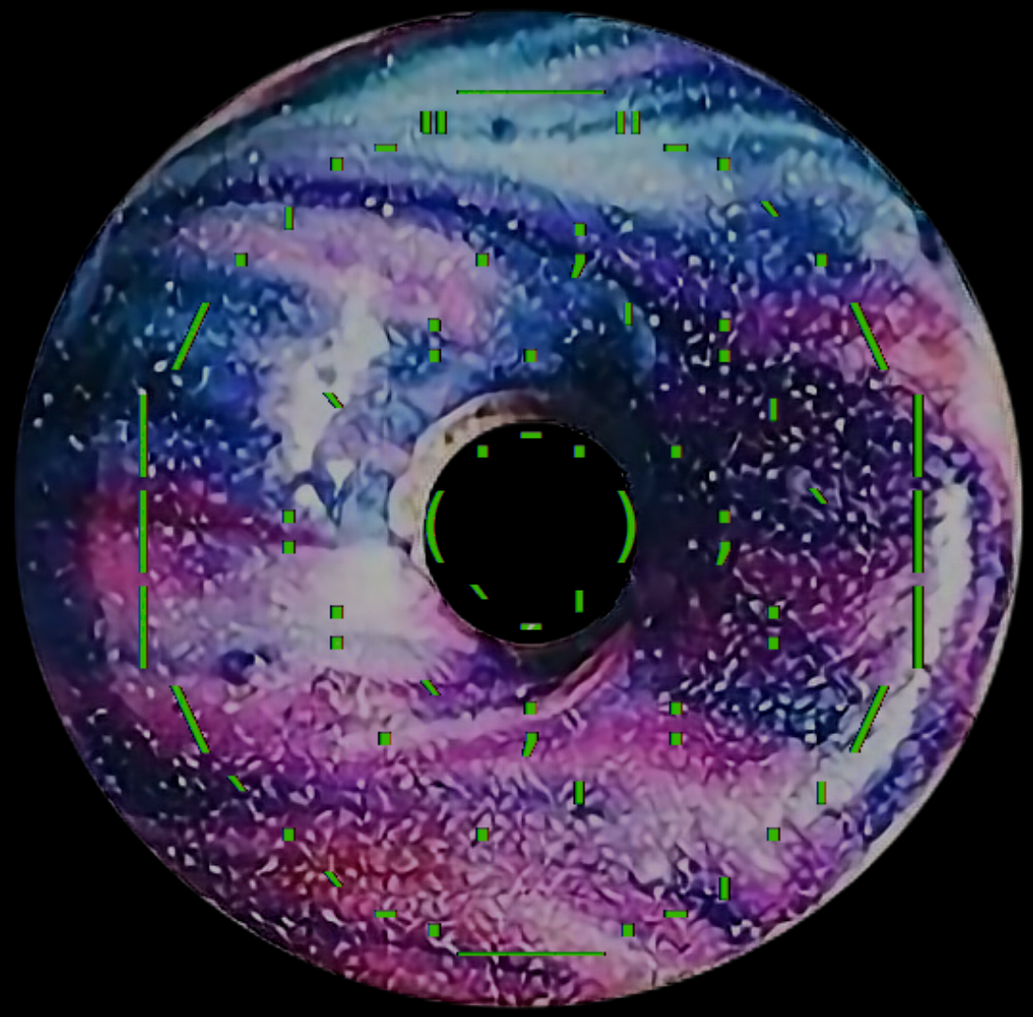
A (practical) Introduction to UA HPC



Astro Code Donuts
2 April 2020
Presented by Rachel Smullen

Goals

- Introduce you to UA HPC resources
- Teach you some basic HPC lingo so you can start to read the docs or learn to use other HPC (like XSEDE)
- Leave you with the ability to interact with Ocelote/El Gato
  - Use software, run programs, find help

I will NOT

- Teach you how to code for HPC
  - That’s an entire workshop/class/degree (and it isn’t my forte)!

High Performance Computing at the University of Arizona

- We have 2—3 supercomputers on campus at any time
  - Upgraded every ~2 years!
  - Supported by 3—5 full time staff
- Available to anyone at UA

*if you email them, you can go visit the systems in person, which is super awesome

More information than you could ever want about the systems and how to use them at

docs.hpc.arizona.edu

	 **El Gato**	 **Ocelote**	**New System (unnamed)**
Operating System	CentOS 7.6	CentOS 6.1	CentOS 8
Nodes	**129 total** • 90 GPU (K20X) • 39 CPU	**420 total** • 44 GPU (P100) • 366 CPU • 1 2 TB “Fat Node”	**192 total** • 24 GPU (V100)
Cores/Node	16 (2.66 GHz)	28 (2.3 GHz)	96 (2.4 GHz)
Memory/Node	• 256 GB for GPU • 64 GB for CPU	• 192 GB for CPU • 2 TB for Fat Node	• 512 GB for regular • 3 TB for Fat Node
Maximum submission (for standard queue)	• 512 cores • 240 hours wall time • 75 individual jobs • 1000 queued	• 2016 cores • 240 hours wall time • 500 individual jobs • 3000 queued	?

Glossary

- **Supercomputer**: A collection of lots of computers that can work together for science!
- **Node**: A single unit of compute resources containing multiple CPUs, lots of memory, maybe GPUs
- **Core**: An individual CPU (or CPU thread) on a node
- **Job**: A task requiring resources that you ask the supercomputer to run
    - **jobid**: the unique identifier assigned to a job
- **Allocation**: The amount of time/resources a group has been given
- **Scheduler**: The software that organizes, prioritizes, and oversees every user's jobs. UA HPC uses "**PBS**"
- **Queue**: The set of policies and priority that limits your job; the "line" you send your job to
    - Think TSA-Pre vs. normal airport security: TSA-Pre is a higher priority queue and therefore has priority access to resources at the expense of people in the normal line
    - **Standard queue**: A higher priority queue available to all faculty groups, but has limited time in monthly allocation
    - **Windfall queue**: Wild west queue; unlimited resources, but preempted by any job with higher priority
- **Wall Time**: The time that it takes a job to run as measured by a human waiting for it to finally finish
- **CPU Time**: The total time it takes all CPUs you requested to run your job; CPU time=Wall time * # CPUs
- **Module**: A software package you can load that is managed by the system administrators

Accessing UA HPC

Your computer

Gatekeeper Node
ssh -X yourname@hpc.arizona.edu

El Gato login node
elgato **(no graphics forwarding)**
ssh -X elgato **(graphics)**

Ocelote login node
ocelote **(no graphics forwarding)**
ssh -X ocelote **(graphics)**

El Gato compute nodes
accessed via qsub job submission

Ocelote compute nodes
accessed via qsub job submission

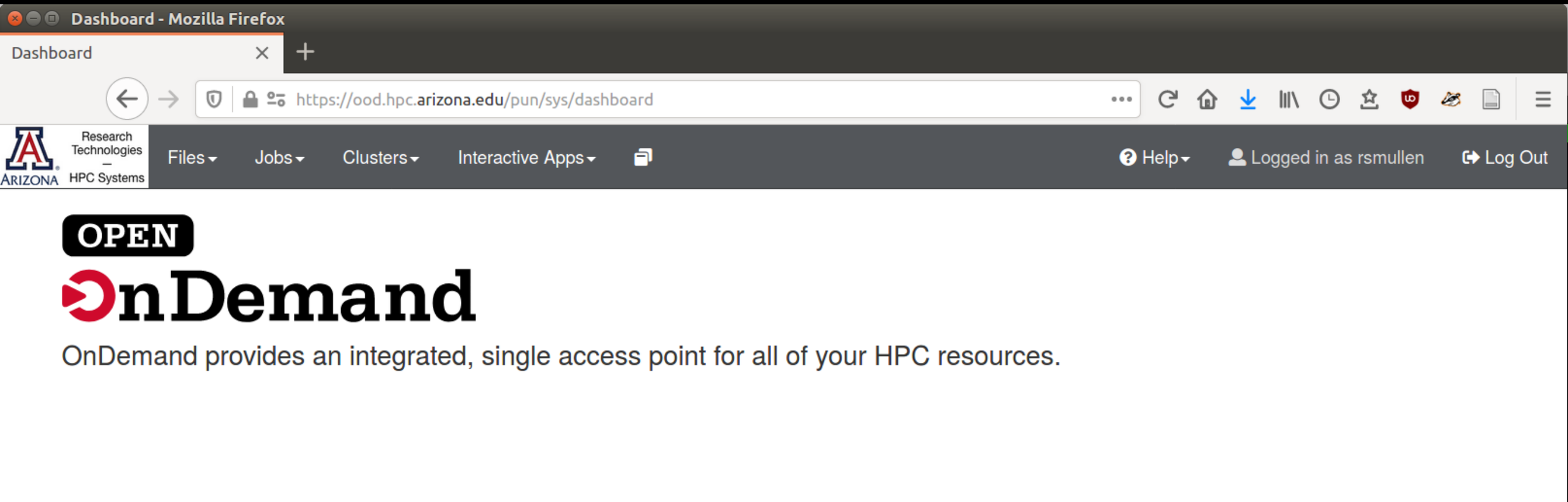
HPC Storage
Accessible by all HPC resources

Open OnDemand

ood.hpc.arizona.edu

UA HPC uses a really nice web interface called Open OnDemand to allow users to interact with the systems without SSH

I'll go through the basic features; OOD can get you 95% of the way to being a competent HPC user!



Dashboard - Mozilla Firefox

Dashboard

https://ood.hpc.arizona.edu/pun/sys/dashboard/

Research Technologies – HPC Systems

Files Jobs Clusters Interactive Apps

Home Directory

/rsgrps

OPE

OnDemand

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

File Explorer - /home/u14/rsmullen/ - Mozilla Firefox

Dashboard | File Explorer - /home/u14/rsmul

Can directly type path if you know where you're going

https://ood.hpc.arizona.edu/pun/sys/files/fs/home/u14/rsmullen/

File Explorer

Go To... | >_ Open in terminal | New File | New Dir | Upload | Show Dotfiles | Show Owner/Mode

- Home Directory
  - CodeBlocks_Fortran_v1.2_Linux64
  - Desktop
  - Documents
  - Downloads
  - Music
  - Pictures
  - Public
  - Templates
  - Videos
  - anaconda2
  - anaconda3
  - astrodendro
  - cnpy-master
  - ffmpeg-git-20160127-64bit-static
  - foo
  - intel
  - mercury
  - new_pys_7Aug_bin.new2
  - ondemand
  - openacc
  - plan_more
  - programs_10Jul2015
  - programs_17Jan2018
  - programs_22Dec2015
  - programs_23Jul2015

/home/u14/rsmullen/

View | Edit | Rename/Move | Download | Copy | Paste | (Un)Select All | Delete

name	size	modified date
..	<dir>	
CodeBlocks_Fortran_v1.2_Linux64	<dir>	04/26/2014
Desktop	<dir>	01/31/2020
Documents	<dir>	01/31/2020
Downloads	<dir>	01/31/2020
Music	<dir>	01/31/2020
Pictures	<dir>	01/31/2020
Public	<dir>	01/31/2020
Templates	<dir>	01/31/2020
Videos	<dir>	01/31/2020
anaconda2	<dir>	08/18/2016
anaconda3	<dir>	11/21/2019
astrodendro	<dir>	09/13/2016
cnpy-master	<dir>	10/12/2017
ffmpeg-git-20160127-64bit-static	<dir>	01/27/2016
foo	<dir>	10/04/2017
intel	<dir>	04/26/2016
mercury	<dir>	08/31/2015
new_pys_7Aug_bin.new2	<dir>	08/07/2019
ondemand	<dir>	03/29/2018
openacc	<dir>	10/23/2018

Easy way to upload/download/view files. Useful to see saved plots!

Dashboard - Mozilla Firefox

Dashboard

https://ood.hpc.arizona.edu/pun/sys/dashboard/

Research Technologies — HPC Systems

Files Jobs Clusters Interactive Apps

Active Jobs

Job Composer

OPEN OnDemand

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

Active Jobs - Mozilla Firefox

Dashboard | Active Jobs

https://ood.hpc.arizona.edu/pun/sys/activejobs?jobcluster=all&jobfilter=all

All Jobs | All Clusters

Active Jobs

Show 50 entries

Filter:

	ID	Name	User	Account	Time Used	Queue	Status	Cluster
⌄	68593.elgato-a…	LM	xunyuchen		00:00:00	el_standard	Queued	ElGato

Queued

LM
68593.elgato-adm

Cluster	ElGato
PBS Id	68593.elgato-adm
Job Name	LM
User	xunyuchen
Group List	judee
Walltime	02:00:00
Walltime Used	
Node Count	4
Total CPUs	20
CPU Time	00:00:00
Memory	0 b

Can see your jobs/other user's jobs on the clusters. Also has a nice delete button.

Dashboard - Mozilla Firefox

Dashboard

https://ood.hpc.arizona.edu/pun/sys/dashboard/

ARIZONA Research Technologies — HPC Systems

Files Jobs Clusters Interactive Apps

Active Jobs

Job Composer

OPEN OnDemand

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

Job Composer - Mozilla Firefox

Dashboard | Job Composer

https://ood.hpc.arizona.edu/pun/sys/myjobs/workflows

Open OnDemand / Job Composer Jobs Templates Help

Job was successfully created.

Jobs

+ New Job ☆ Create Template

Edit Files | Job Options | Open Terminal | Submit | Stop | Loading | Delete

Show 25 entries Search:

Created	Name	ID	Cluster	Status
April 2, 2020 9:56am	**Ocelote Sequential Job**		Ocelote	Not Submitted

Showing 1 to 1 of 1 entries Previous 1 Next

Job Details

Job Name:

Ocelote Sequential Job

Submit to:

Ocelote

Account:

Not specified

Script location:

/home/u14/rsmullen/ondemand/data/sys/myjobs/projects/def

Has templates/fill in the blank examples of job submission scripts

Dashboard - Mozilla Firefox

Dashboard

https://ood.hpc.arizona.edu/pun/sys/dashboard/

Research Technologies — HPC Systems

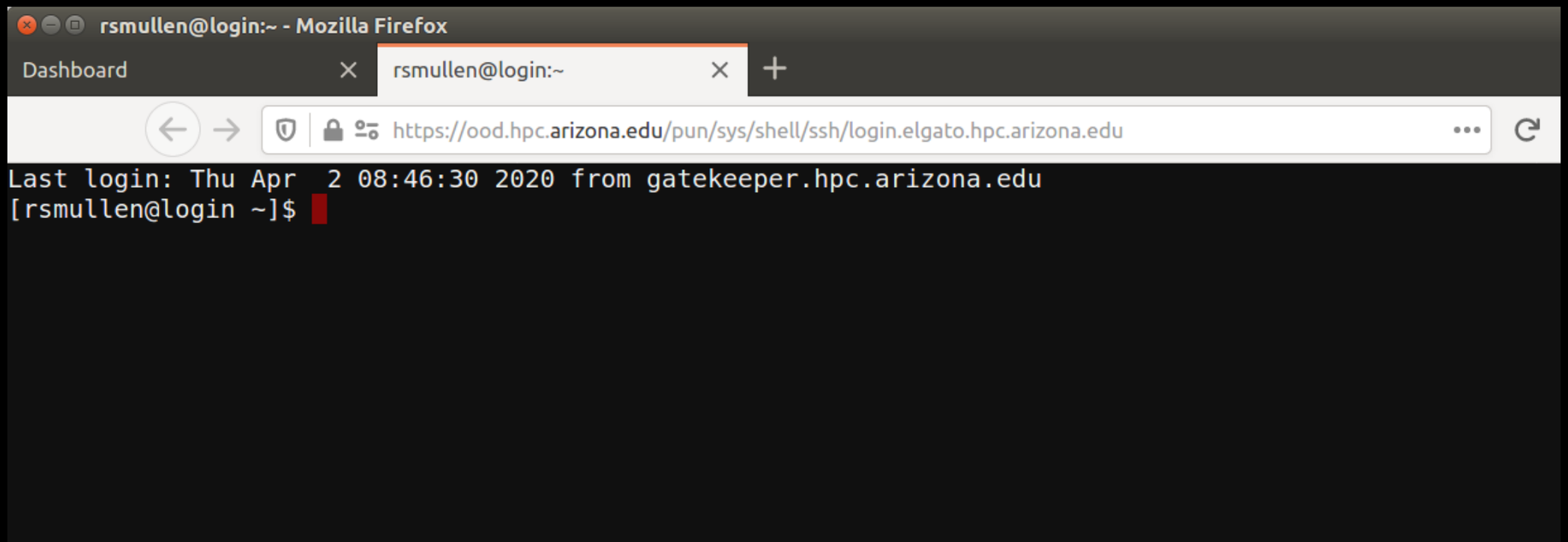
Files Jobs Clusters Interactive Apps

>_ElGato Shell Access

>_Ocelote Shell Access

OPEN OnDemand

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



*no X-forwarding and some funky keyboard interactions
If you want to run matplotlib, use the ‘Agg’ backend

Dashboard - Mozilla Firefox

Dashboard

https://ood.hpc.arizona.edu/pun/sys/dashboard/

ARIZONA Research Technologies – HPC Systems

Files Jobs Clusters Interactive Apps

OPEN OnDemand

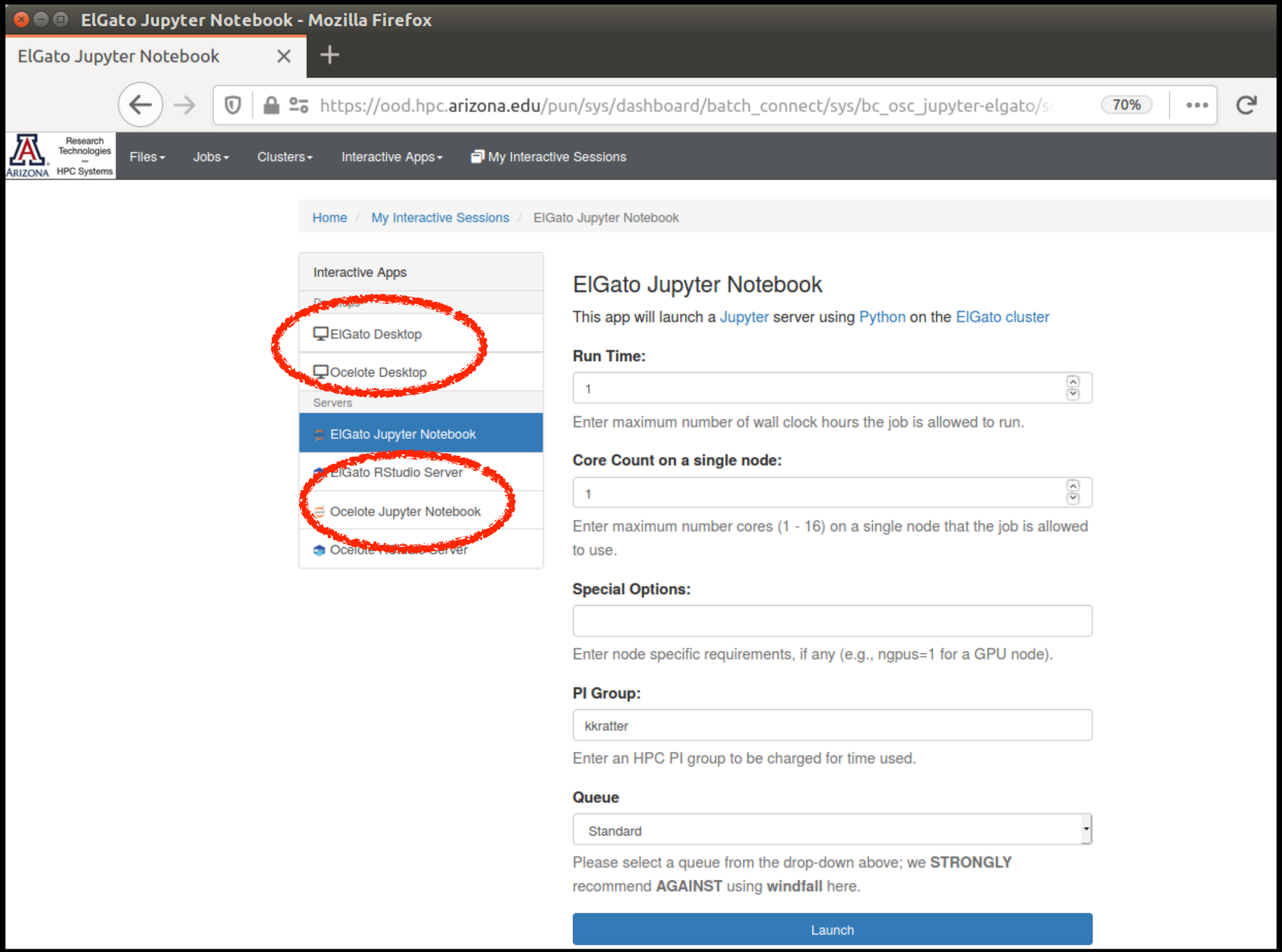
OnDemand provides an integrat for all of your HPC resources.

Desktops

- ElGato Desktop
- Ocelote Desktop

Servers

- ElGato Jupyter Notebook
- ElGato RStudio Server
- Ocelote Jupyter Notebook
- Ocelote RStudio Server



Can launch Jupyter Notebooks (using HPC Python; otherwise Jupyter a pain in the butt).
Can also launch Desktop if you need graphical interface that is hard over SSH.

Helpful Commands for HPC Command Line

- Figuring out your resources

  - **va**: find your available compute resources

```
[rsmullen@login ~]$ va
rsmullen current allocations:
-----------------------------
Group                    Type                 Available Time
-----                    ----                 --------------
kkratter                 Standard             19200:00
kkratter                 ElGato Standard      7000:00
kmkshare                 ElGato High Priority 398305:37
kmkshare                 el_high_pri_time     500000:00
```

Queue

CPU-hours

  - **uquota**: find your storage resources and usage

```
[rsmullen@login ~]$ uquota
                                    used  soft limit  hard limit      files/limit
/rsgrps/kkratterstudents          83.78T         85T         85T   40480886/52224000
/home                              11.4G       50.0G       50.0G
```

  - **nodes-busy**: find out if there are free nodes

```
[rsmullen@login ~]$ nodes-busy
Thu Apr  2 10:37:28 MST 2020
Total of Nodes:  129

Core count assigned by PBS:    2049

Total Nodes Running Jobs:       129
```

Helpful Commands for HPC Command Line

- Interacting with jobs
  - **`qsub (args)`**: submit a job
  - **`qdel (jobid)`**: delete a job
  - **`qpeek (jobid)`**: check job output
  - **`qstat`**: check the status of your job
    - **`-u $USER`** or **`username`**: only display jobs for user
    - **`-a`**: display all queued and running jobs
    - **`-t`**: display all sub-jobs
    - **`-s`**: display extra information about job, including the node(s) the job is running on

```
[rsmullen@login ~]$ qsub < pbs_test.sh
94232.elgato-adm
[rsmullen@login ~]$ qpeek 94232
hello
[rsmullen@login ~]$ qstat -ast -u $USER

elgato-adm:

                                                            Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
94232.elgato-ad rsmullen el_stand test         6341   1   2    6gb 01:00 R 00:00
   Job run at Thu Apr 02 at 10:57 on (cpu35:ncpus=2:mem=6291456kb)
[rsmullen@login ~]$ qdel 94232
[rsmullen@login ~]$ qstat -ast -u $USER
```

How to: Transfer Files

https://public.confluence.arizona.edu/display/UAHPC/Transferring+Files

- Using OOD file interface
- Using Globus (globus.org): allows end-to-end file transfers
- Using gdrive: instructions on HPC docs
- SCP/FTP: HPC asks that you use filexfer.hpc.arizona.edu for large transfers to reduce strain on login nodes

How to: Load Software

- Can install user-managed software (e.g., Anaconda Python)
- HPC maintains a lot of useful software/drivers so you don't have to
- Available resources: https://public.confluence.arizona.edu/display/UAHPC/Software+Resources

How to: Load Software

- If you want to load HPC software (modules):
  - **`module avail`**: list everything that you can load
  - **`module load (packagename)`**: load the software
  - **`module list`**: show everything you have loaded

```
[rsmullen@login ~]$ module avail

--------------------------- /opt/ohpc/pub/moduledeps/gnu8-openmpi3 ---------------------------
   boost/1.71.0     mpiP/3.4.1              petsc/3.12.0       scorep/6.0
   dimemas/5.4.1    netcdf-cxx/4.3.1        phdf5/1.10.5       sionlib/1.7.4
   extrae/3.7.0     netcdf-fortran/4.5.2    pnetcdf/1.12.0     tau/2.28
   fftw/3.3.8       netcdf/4.7.1            scalapack/2.0.2    trilinos/12.14.1
   imb/2018.1       omb/5.6.2               scalasca/2.5
```

…spliced for space. There's a lot of software.

```
[rsmullen@login ~]$ module load python/3.5
[rsmullen@login ~]$ module list

Currently Loaded Modules:
  1) autotools   3) gnu8/8.3.0   5) openmpi3/3.1.4
  2) prun/1.3    4) ohpc         6) python/3.5/3.5.5
```

How to: Run Interactive Jobs

- If you want to have resources for you to use live (e.g., for code development or testing), use an interactive node!

```
qsub -I -X -W group_list=kkratter -q standard
-l select=1:ncpus=16:ngpus=1:mem=250gb:pcmem=16gb
 -l walltime=48:0:0 -l place=pack:shared
```

- **qsub**: submit to PBS
- **-I**: interactively
- **-X**: with X-forwarding (graphics)
- **-W group_list=kkratter**: use the resources from a group you belong to
- **-q standard**: use the standard queue. Can also be ‘windfall’, ‘medium’, etc.
- **-l select=1:ncpus=16:ngpus=1:mem=250gb:pcmem=16gb**: specify the resources you want to use. 1 node : 16 CPUs :1 GPU : 250 GB total memory : 16 GB of memory per core
  - **mem** should be ~**pcmem*ncpus**
- **-l walltime=48:0:0**: run for 48 hours at maximum
- **-l place=pack:shared**: put my resources on as few nodes as possible, and let other people run on remaining resources (this is, for most of the purposes I’ve seen, a good default)

An Aside on pcmem

- **`pcmem`** (the per-core memory) is the parameter that sets the type of node that you are assigned.

- On El Gato, there are CPU nodes with 64 GB of RAM per node and GPU nodes with 250 GB of RAM per node.
  - **`pcmem=16gb`** puts you on **GPU** nodes
  - **`pcmem=4gb`** puts you on **CPU** nodes

- On Ocelote, there are CPU/GPU nodes with 188GB of available RAM and a “fat node” with 2TB or RAM
  - **`pcmem=6gb`** puts you on a **normal** node
  - **`pcmem=42gb`** puts you on the **fat** node

How to: Run a Simple Job

(Credit Jenn Kadowaki)

```
# Your job will use 1 node, 28 cores, and 224gb of memory total.
#PBS -q standard
#PBS -l select=1:ncpus=16:mem=250gb:ngpus=1

### Specify a name for the job
#PBS -N bert_olid_predict

### Specify the group name
#PBS -W group_list=dfz

### Used if job requires partial node only
#PBS -l place=pack:exclhost

### CPUtime required in hhh:mm:ss.
### Leading 0's can be omitted e.g 48:0:0 sets 48 hours
#PBS -l cput=32:00:00

### Walltime is how long your job will run
#PBS -l walltime=02:00:00

### Joins standard error and standard out
#PBS -o bert_olid_predict.o
#PSB -e bert_olid_predict.e

### Sends e-mail when job aborts/ends
#PBS -m ae
#PBS -M jkadowaki@email.arizona.edu

################################################################

### DIRECTORIES / NAMES ###
USER=$(basename $HOME)
export CONTAINER=/extra/jkadowaki/SemEval2019/container/nvidia-tensorflow.18.09-py3.simg

cd /extra/$USER/SemEval2019
module load singularity

date +"Start - %a %b %e %H:%M:%S %Z %Y"
singularity exec --nv $CONTAINER bash predict_bert.sh
date +"End - %a %b %e %H:%M:%S %Z %Y"
```

Submit a job for a single node to the standard queue

Give your job a name so you know what it is

Set up the details for your job

Output ‘error’ (-e) and ‘output’ (-o) files at the end of the job

Have the scheduler email you when your job does things

This is the code you want to run

Put this in a file like ‘script.sh’

Submit to the scheduler with **qsub < script.sh**

How to: Run Embarrassingly Parallel Jobs

- Embarrassingly parallel computing is where you want to do the same thing (like run the same program) a bunch of times
    - Useful for making figures, computing on simulation snapshots, or running a grid of models
- To do this, we use a job array, which runs the same script for each sub-job in the job array
    - sub-jobs are accessed in the script by **`${PBS_ARRAY_INDEX}`**

How to: Run Embarrassingly Parallel Jobs

(Credit Rachel Smullen)

```
## Run on windfall queue
#PBS -q windfall
## Use 2 CPUs per job
#PBS -l select=1:ncpus=2:mem=32gb:pcmem=16gb

#PBS -N catalog

#PBS -W group_list=kkratter

## Don't care if other people run jobs on the same node
#PBS -l place=pack:shared

## Run for 24 hours
#PBS -l walltime=24:00:00

## Put error/output files in the folders
#PBS -e err/
#PBS -o out/


### JOB ARRAY: set sub_jobs with indexes 1-100
#PBS -J 1-100


#### My code

# Make sure I'm in the right directory
homedir=/rsgrps/brant/rsmullen/BinaryFormation/density_7E-20_new2/
cd $homedir

# Make sure the python I want is loaded
export PATH="/home/u14/rsmullen/anaconda3/bin:$PATH"
export PYTHONPATH=/home/u14/rsmullen/anaconda3/pkgs:$PYTHONPATH
export PYTHONPATH=/home/u14/rsmullen/.local/lib:$PYTHONPATH

# Run the program for individual sub-job ${PBS_ARRAY_INDEX}
python make_catalog.py ${PBS_ARRAY_INDEX}
```

Put this in a file like ‘script.sh’
Submit to the scheduler with **`qsub < script.sh`**

Set up the details for your job

Run sub-jobs 1-100

Load Anaconda Python

Give the sub-job index to Python as the index for what iteration to run

How to: Run Parallelized Code

(Credit Rixin Li)

Put this in a file like ‘script.sh’
Submit to the scheduler with **qsub < script.sh**

Set up the details for your job

```
### Set up Job
#PBS -N Z3P5t01_HR
#PBS -W group_list=kmkshare
#PBS -q el_high_pri
#PBS -m bea
#PBS -l place=free:shared
#PBS -l walltime=120:00:00
#PBS -l cput=15360:00:00

### Select 8 full CPU nodes
#PBS -l select=8:ncpus=16:mem=64gb:pcmem=4gb

### Load modules, including MPU for parallel
module load openmpi3 fftw

### set directory for job execution
cd /home/u5/rixin/runs/kkruns/tauZmap_SI/Z3P5t01_HR/

### run your parallel executable program
mpirun -np 128 ./Athena -i ./athinput.parsg_strat2d -t 120:00:00 > output.txt
2>& 1
```

Select many nodes for program to run on

Run a parallelized code

Rachel’s plea to users: Do’s and Don’ts

- Be a good HPCitizen

- Do check how many resources your job needs before running!
  - Your job will get killed if you exceed what you asked for; you're wasting your allocated time
  - If you request too much, you’re wasting everyone else's time because we can’t use them!

- Don’t run programs on the login nodes
  - You’re making the entire system slower for everyone else!

- Do use job arrays instead of submitting hundreds of individual jobs
  - Please.

- Don’t overdo your I/O usage (how much you’re accessing the disk)
  - Excessive (*writing to or reading from file every few iterations) I/O makes HPC sluggish and inefficient for everyone!

How to find help

- HPC is hard. I get it. It takes a while to get comfortable.
- If you have problems:
  - As a question to Slack (#code on grad student slack, or #help-computer-support on SO slack, or DM)
    - The experienced HPC users (me, Rixin, Gabriele, etc) usually respond :)
  - Email me (or most of your local theory grad students)
  - Email the HPC staff: hpc-consult@list.arizona.edu