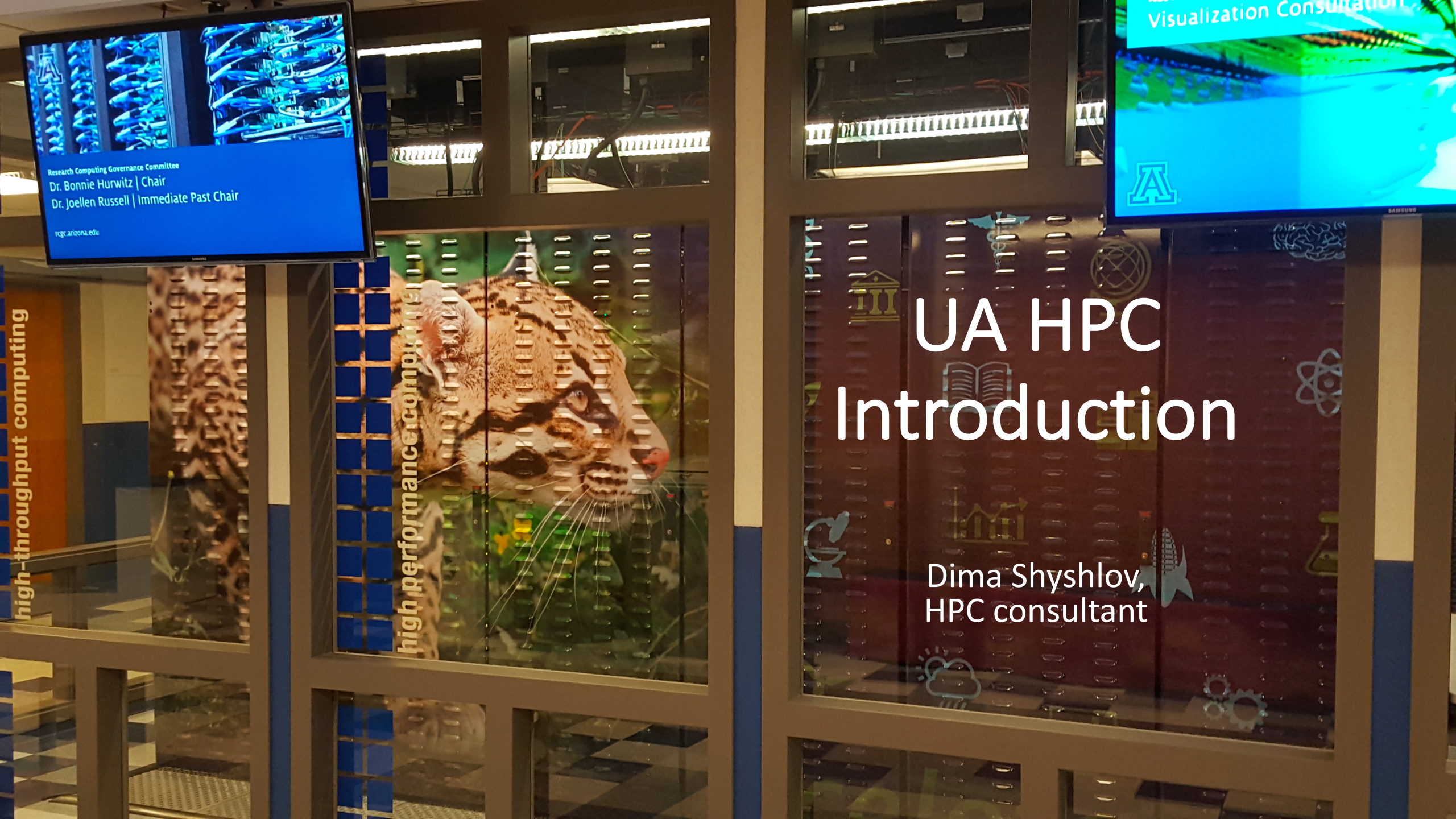


UA HPC Introduction

Dima Shyshlov,
HPC consultant

Requesting an HPC account

- accounts.arizona.edu

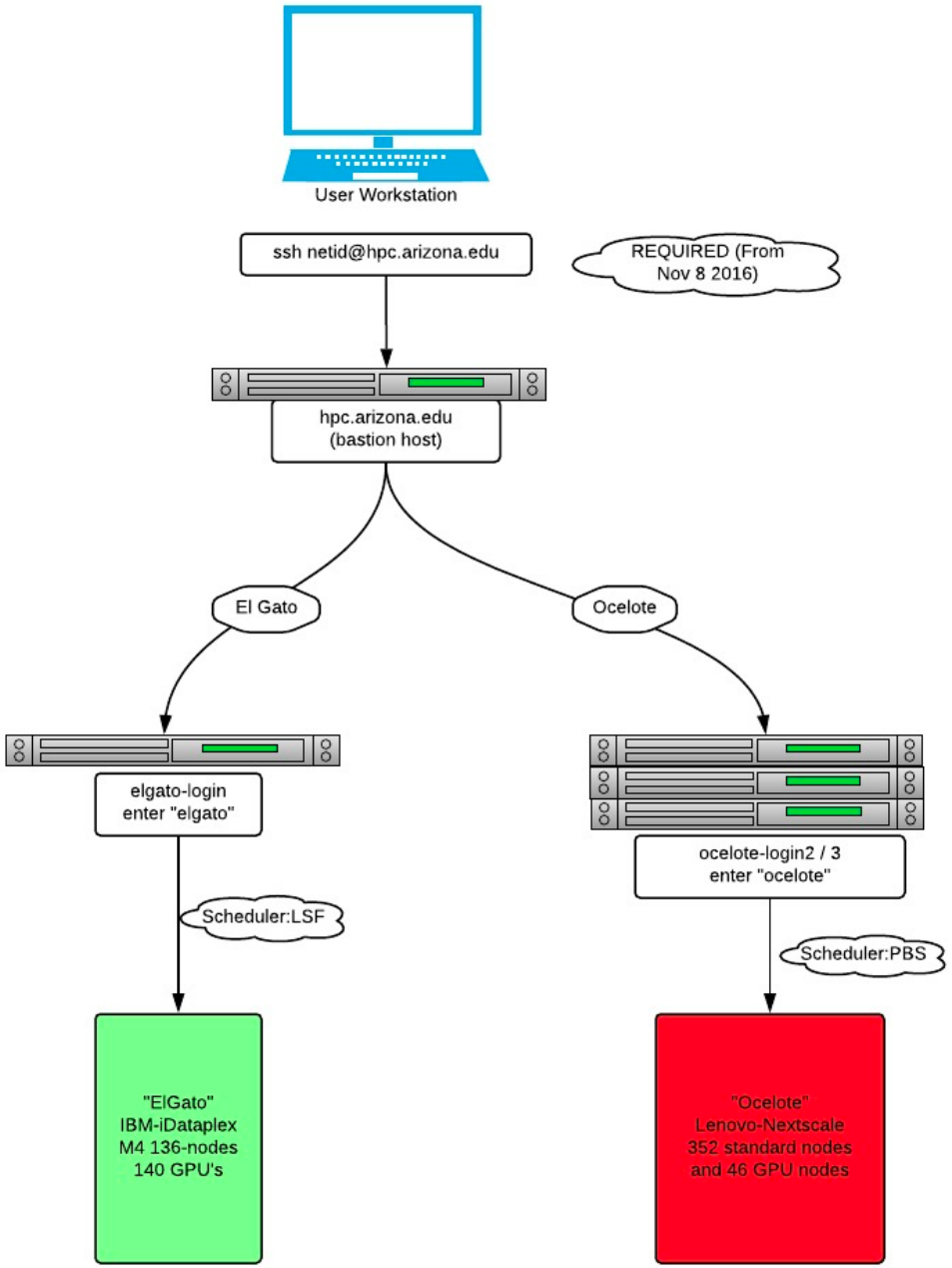
  -> manage your accounts

  -> HPC account

  -> Notify your sponsor of your request at the HPC sponsorship page

HPC systems in UofA

- El Gato
- Ocelote:
  - over 300 compute nodes
  - 28 cores per node
  - 6GB of memory per core



Accessing the HPC system

- Software for SSH connection:
  - Windows – Putty
  - Mac – Terminal

https://softwarelicense.arizona.edu/
> Students
>SSH

```
wncs-MacBook-Pro:~ dshyshlov$ ssh dshyshlov@hpc.arizona.edu
Password:
Duo two-factor login for dshyshlov

Enter a passcode or select one of the following options:

 1. Duo Push to XXX-XXX-0896
 2. Phone call to XXX-XXX-0896
 3. SMS passcodes to XXX-XXX-0896 (next code starts with: 7)

Passcode or option (1-3): 1
Success. Logging you in...
Last login: Wed Jan 31 16:36:22 2018 from dhcp-10-132-181-137.uawifi.arizona.edu
This is a bastion host used to access the rest of the environment.

Shortcut commands to access each resource
-----------------------------------------
Ocelote:                El Gato:
$ ocelote               $ elgato

[dshyshlov@gatekeeper ~]$ ocelote
Last login: Wed Jan 31 09:13:57 2018 from gatekeeper.hpc.arizona.edu
[dshyshlov@login3 ~]$
```

Login nodes VS Compute nodes

- Login nodes are for:
  - editing code, scripts
  - submitting jobs (calculations)
  - checking status of the jobs
  - testing and troubleshooting
  - interactive tasks
  - Ocelote has 3 login nodes

- Compute nodes are for running jobs
  - ideal for batch jobs
  - Ocelote has 331 compute nodes

Anatomy of a computing node

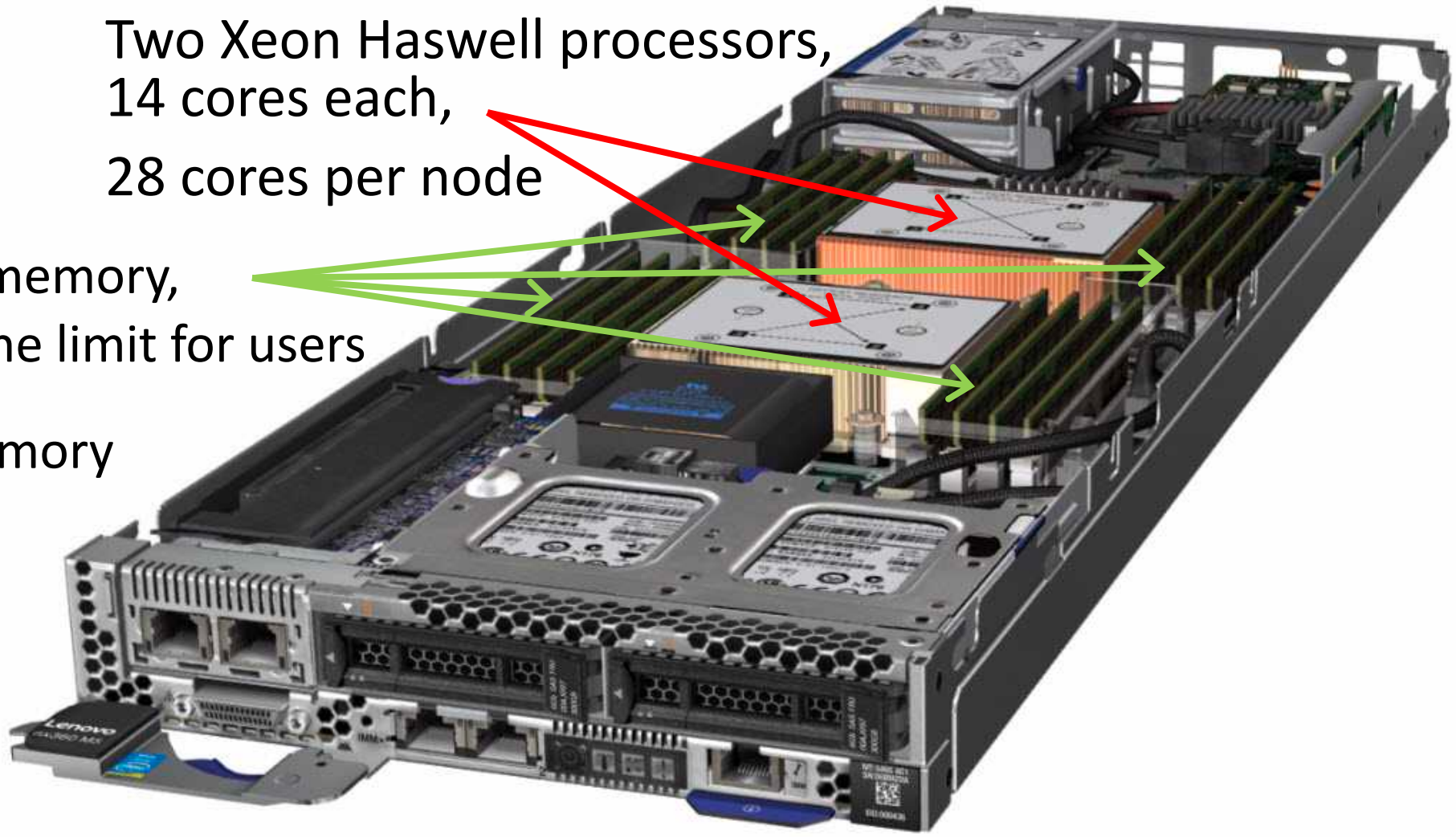
Two Xeon Haswell processors,
14 cores each,

28 cores per node

192 GB of memory,

168 GB is the limit for users

6 GB of memory
per core



File transfer

- There are special nodes for data transfer
  - **filexfer.hpc.arizona.edu**
  - old hostname sftp.hpc.arizona.edu still works as well
- Connecting to file transfer node
  - sftp **NetID**@filexfer.hpc.arizona.edu
- File transfer software
  - WinSCP (Windows), Cyberduck (Windows and Mac), Fugu (Mac)
  - https://softwarelicense.arizona.edu/ssh-clients-windows-and-mac
- Other ways of file transfer:
  - Globus (large files), scp, rsync, irods

Storage and Allocation

- Storage:
  - Home directory – 15GB
  - /extra – 200GB
  - /xdisk – temporary storage up to 1TB
  - /rsgrps – rented storage by research groups
  - /tmp – local scratch, ~850GB on each node, useful for temporary files used during the job execution
  - *uquota* – Linux command to display your used/available storage

- Allocation
  - standard – limited to 24,000 hours/group/month
  - windfall – unlimited, jobs can be preempted
  - *va* – Linux command to display available allocation

Software

- Many software packages are available as modules
  - *module avail* – list all the installed modules
  - *module avail blast* – list all versions of BLAST
  - *module load blast* – load the module (the latest version is usually the default)
  - *module list* – display all the modules loaded in your environment

”Hello, World!” exercise

- Copy exercise files:
  - *git clone https://github.com/dshyshlov/ECOL-346.git*
- List the files and directories:
  - *ls*
- Change directory to UA-HPC-Intro
  - *cd ECOL-346* (use tab for autocompletion)
- List the files again:
  - *ls*

mpi_hello_world.c

- Multicore version of “Hello, World!” program in C language
  - Uses MPI to run on multiple nodes

- Enable using MPI with:
  - *module load openmpi*

- Compile with:
  - *mpicc -o mpi_hello_world mpi_hello_world.c*

PBS Script

- Parameters for scheduler
- Loading necessary software
- Navigating to the working directory
- Run the program

```
#!/bin/bash
#PBS -N JobName
#PBS -m bea
#PBS -M NetID@email.arizona.edu
#PBS -W group_list=msbarker
#PBS -q standard
#PBS -l select=1:ncpus=12:mem=72gb:pcmem=6gb
#PBS -l walltime=0:1:0
#PBS -l cput=0:12:0

module load blast


cd ~/blast_data


makeblastdb –in Ceratopteris1.fasta -out
blastn_results.tsv –outfmt ‘6’ –evalue 10e-5
```

PBS Script

- Display the content of the PBS script on the screen:
  - *cat hpc.sub.blast*
- Edit the PBS script with nano text editor:
  - *nano hpc.sub.blast*
- Fill in the group name
- Submit the script with the command:
  - *qsub hpc.sub.blast*
- Check the job status:
  - *qstat –u NetID*

Output and Error files

- Check the output file
- Check the error file
- Output and error files can be joined together with the PBS script:
  - #PBS –j oe
- You can also specify the file names:
  - #PBS –o output.txt
  - #PBS –e error.txt

Getting help

- docs.hpc.arizona.edu
- hpc-consult@list.arizona.edu
- Google, stackoverflow…