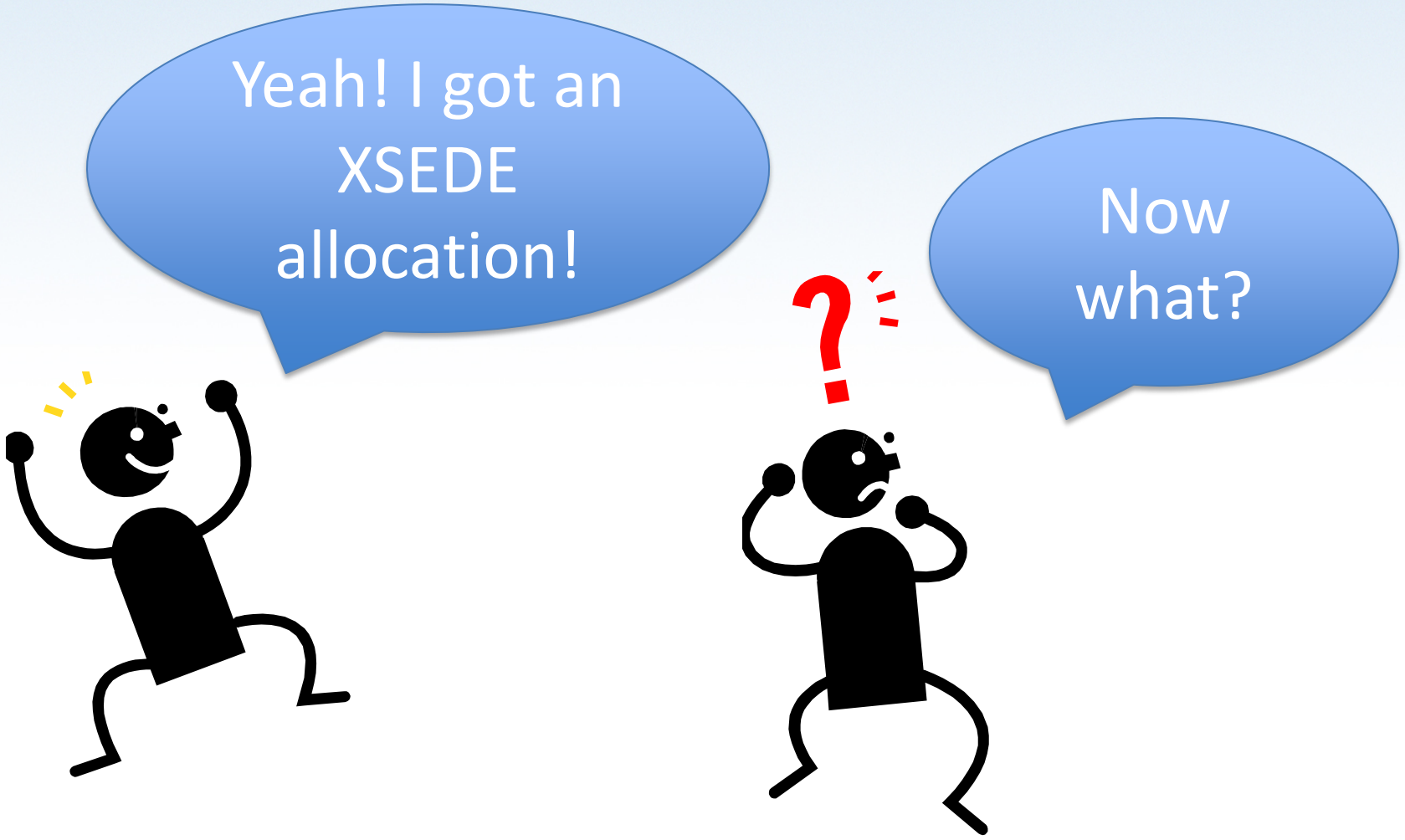
September 10, 2014

XSEDE New User Tutorial

Extreme Science and Engineering Discovery Environment

Jay Alameda
National Center for Supercomputing Applications


Yeah! I got an XSEDE allocation!
Now what?

Learning Outcomes

After completing this tutorial, you will be able to:

- Use the XSEDE User Portal
- Access your XSEDE resources
- Manage files
- Run jobs
- Get help

XSEDE User Portal (XUP)

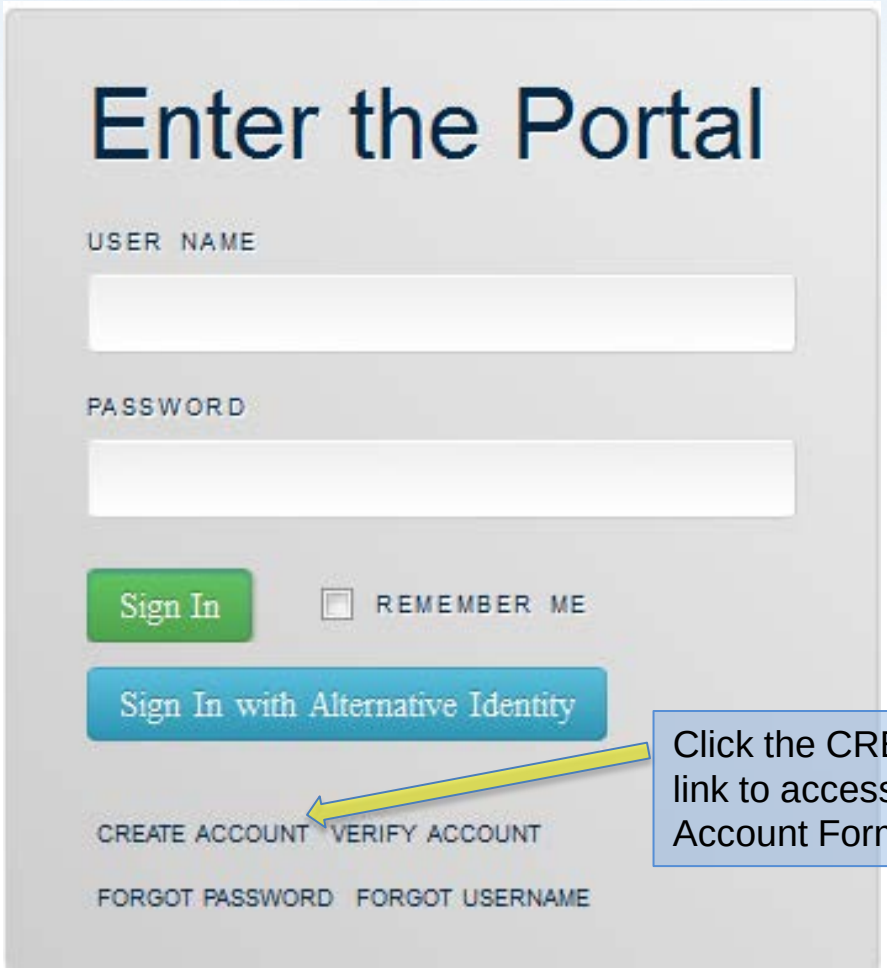
- URL: portal.xsede.org
- Single point-of-entry to information about XSEDE services and utilities for using them
- Anyone can create an XUP user account and access non-project features
- Only XSEDE allocation project members can access project features

Using the XUP

- Create and login to your XUP Account
- Use XSEDE resources responsibly
- Get added to your XSEDE project
- Navigate your personal My XSEDE webpage
- Navigate the information in the XUP

Create and login to your XUP account

portal.xsede.org



Click the CREATE ACCOUNT link to access the XUP User Account Form

1. From the XUP homepage, click CREATE ACCOUNT
2. Complete the User Account Form
3. Verify your account request
4. Select your username and password
5. Login to the XUP

XSEDE Acceptable Use Policy

- Must accept the User Responsibilities Form after creating your XUP account and again at the beginning of each allocation you receive.
- Choose a strong password and protect it.
- Close SSH terminals and log out of the User Portal when you are finished with your session.
- Report Suspicious Activity : email help@xsede.org or call 1-866-907-2383 immediately, regardless of the time of day.

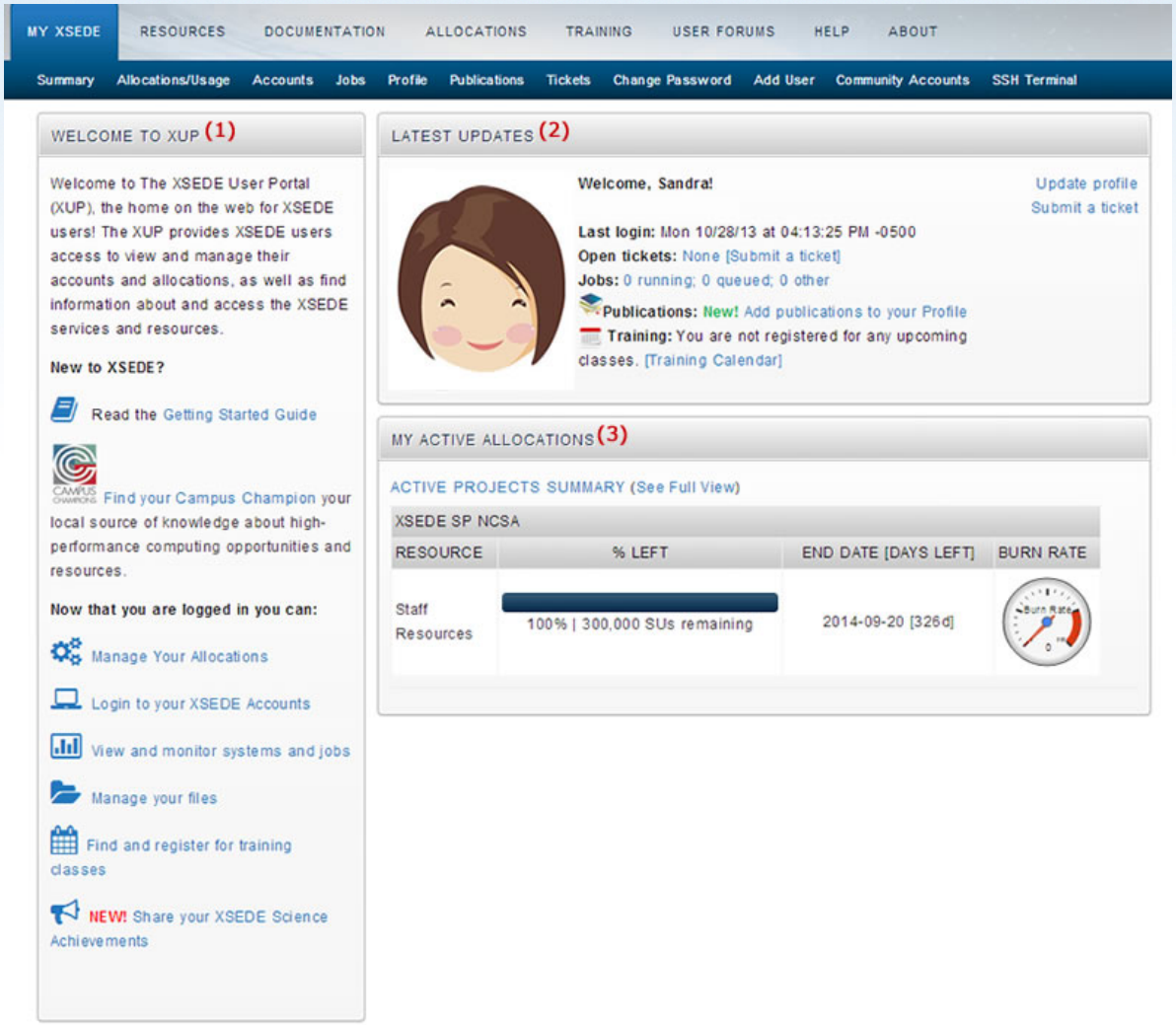
XSEDE Cybersecurity Tutorial

http://www.citutor.org

Get Added to Your XSEDE project

- If you are not the project PI, you will need to be added to your project's account in the XUP.
- Contact your project's PI or Allocation Manager and request that you be added to the project. You will need to provide them with your XSEDE User Portal user name.

Your My XSEDE webpage



(1)

WELCOME TO XUP

- Quick access to commonly used features.

(2)

LATEST UPDATES

- Latest information specific to your user account.

(3)

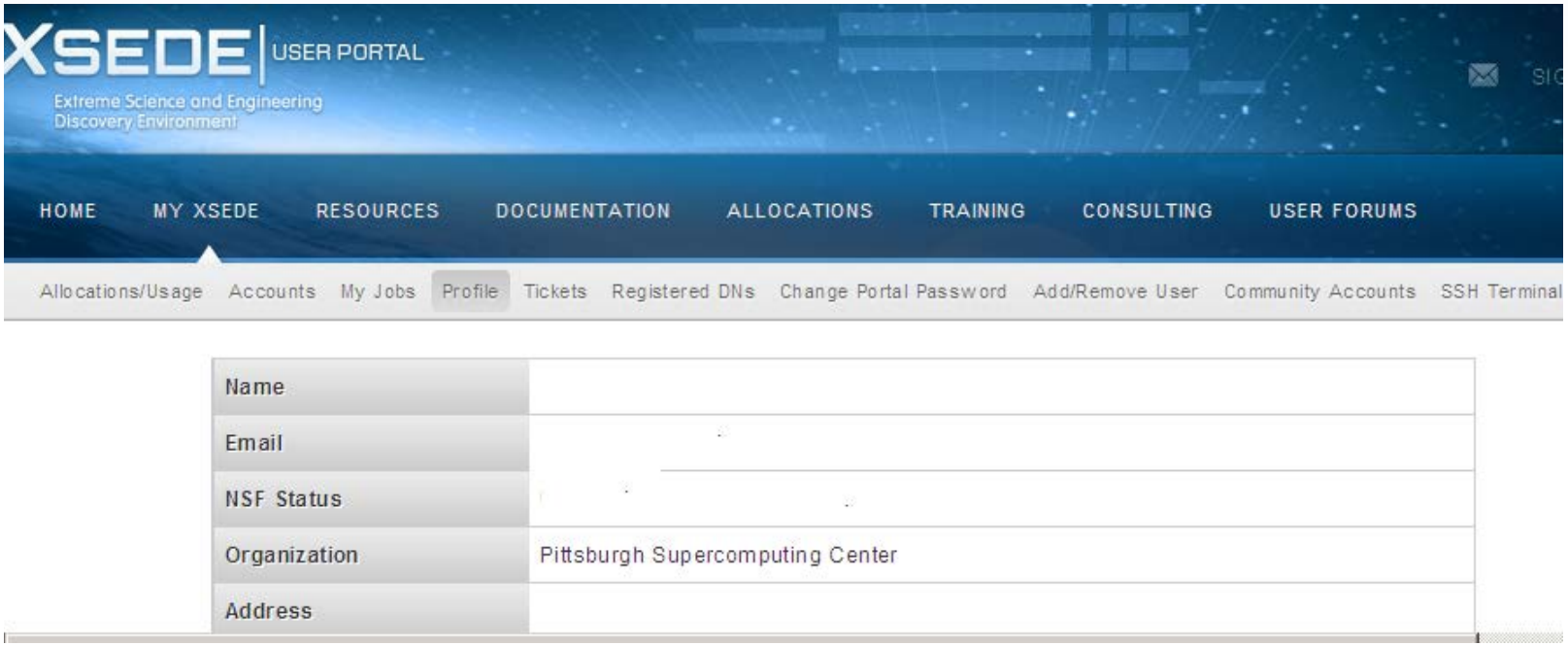
MY ACTIVE ALLOCATIONS

- Summary of the active projects for which you are either a PI or member.

Update your XUP User Profile

MY XSEDE→Profile

- View and or change your user information (organization, address).
- Make sure your email address is correct. XSEDE staff will use it to communicate with you regarding your allocation.

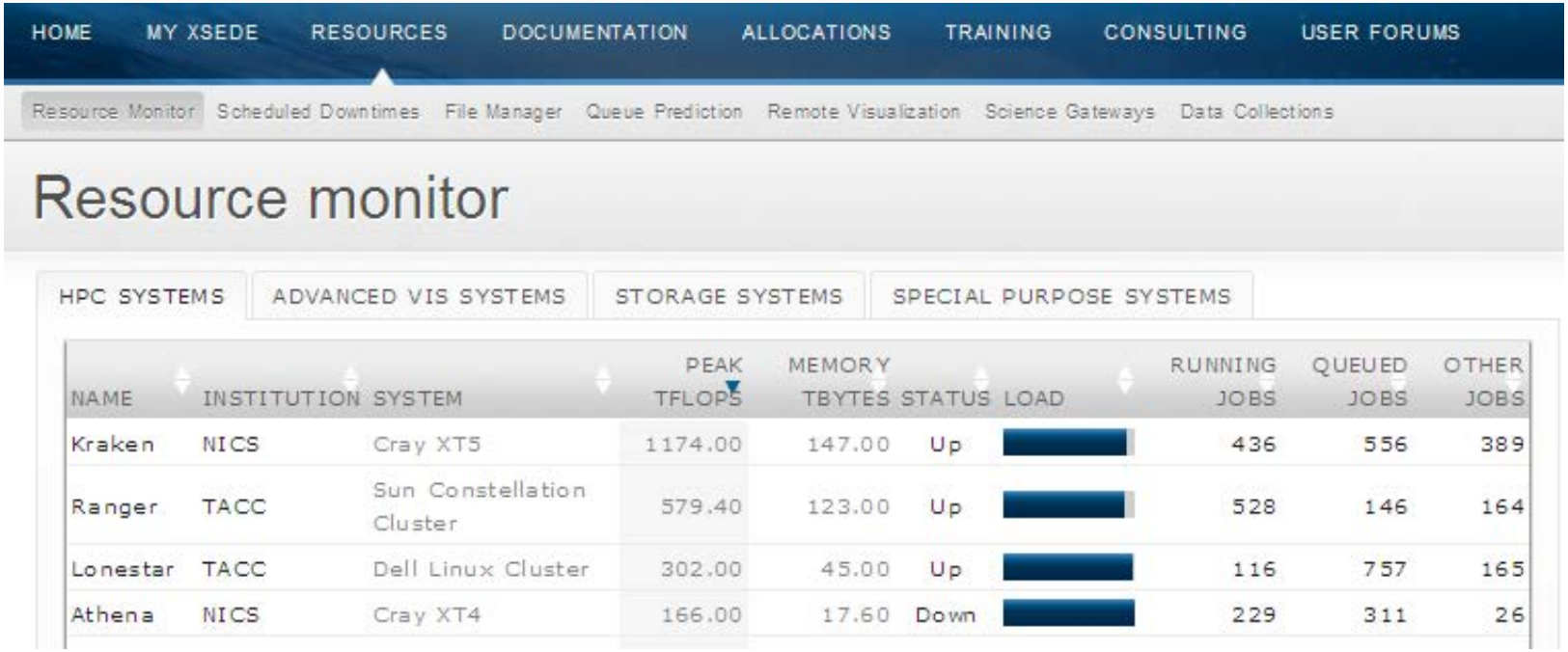


Navigating the XUP

- My XSEDE
- Resources
- Documentation
- Allocations
- Training
- User Forums
- Help
- About

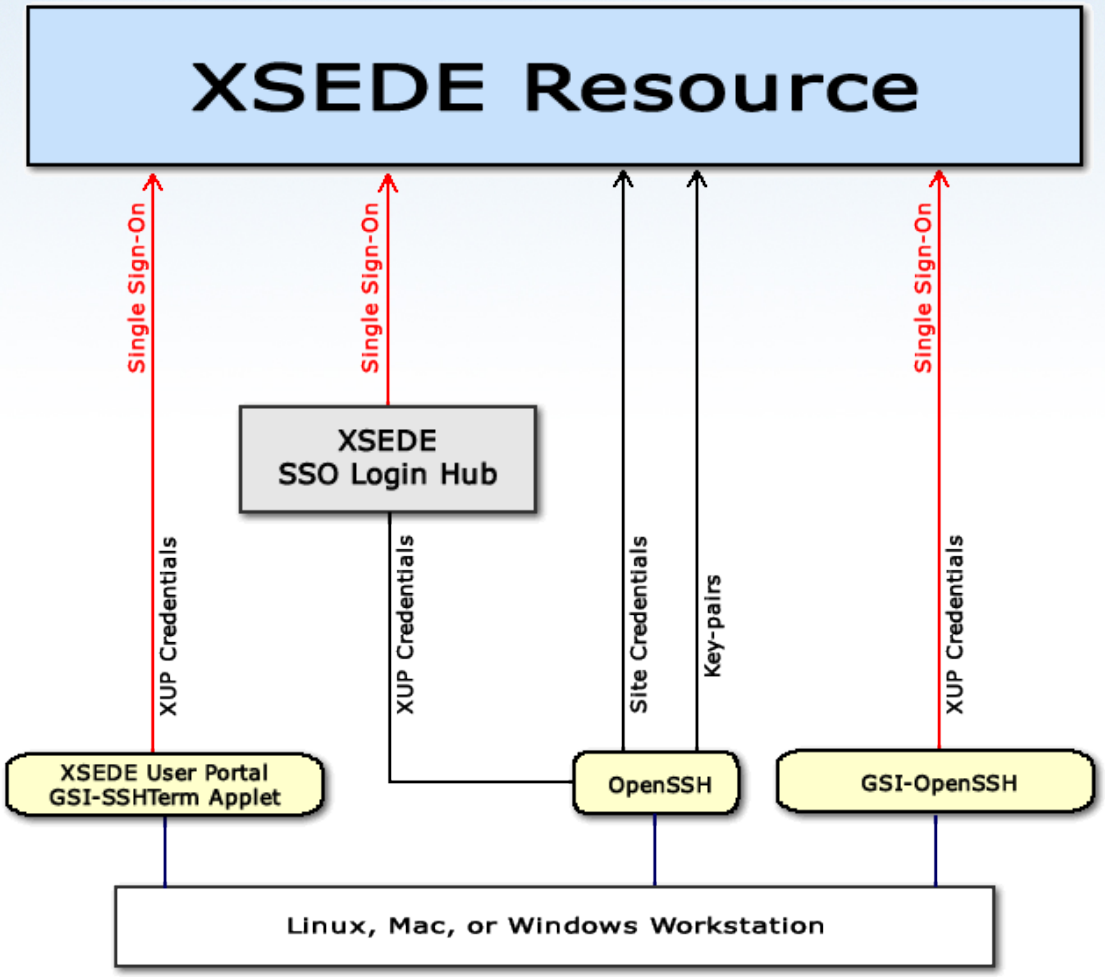
View the XSEDE Resource Monitor

- **Resources -> Systems Monitor**
  - Provides technical and status information for all of XSEDE's resources.
  - The STATUS column indicates whether the system is up or down. If down, can click on status to find when the machine is expected to come back up.



NAME	INSTITUTION	SYSTEM	PEAK TFLOPS	MEMORY TBYTES	STATUS	LOAD	RUNNING JOBS	QUEUED JOBS	OTHER JOBS
Kraken	NICS	Cray XT5	1174.00	147.00	Up		436	556	389
Ranger	TACC	Sun Constellation Cluster	579.40	123.00	Up		528	146	164
Lonestar	TACC	Dell Linux Cluster	302.00	45.00	Up		116	757	165
Athena	NICS	Cray XT4	166.00	17.60	Down		229	311	26

Accessing XSEDE Resources



Authentication Methods

1. Password
   - XUP credentials
   - Site-password
   - One-time password
2. Key-based

Single Sign-On

- Enables logging in once to access all of your allocated resources

Connection Methods

1. XUP GSI-SSHTerm
2. GSI-OpenSSH
3. OpenSSH

XSEDE User Account Mapping

- Portal logins to not necessarily match local logins
- Can access mapping from My XSEDE > Accounts
- (only needed for site passwords, or one-time passwords)

RESOURCE NAME	LOGIN NAME	INSTITUTION	USERNAME	CONNECT
Blacklight	blacklight.psc.teragrid.org	PSC	skappes	Login
Condor	tg-condor.purdue.teragrid.org	Purdue	skappes	Login
Gordon Compute Cluster	gordon.sdsc.edu	SDSC	skappes	Login
Gordon ION	gordon.sdsc.edu	SDSC		

XSEDE SSO Login Hub

An SSO enabled connection point to XSEDE resources

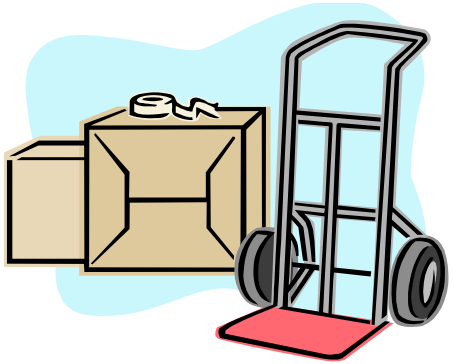
- Move among resources using **gsissh** command

- SSH to **login.xsede.org** using your XUP credentials

Managing your XSEDE files

1. **Where to store files**
   - Home directory
   - Scratch directory
   - Archival storage

2. **How to move files**
   - Command line using globus-url-copy, uberftp, scp, or sftp
   - **Globus Online**



XSEDE File Systems

- **Home directory**
  - Location specified in the environment variable $HOME.
  - Use to store project files you want to keep long term such as source code, scripts, and input data sets.
  - Not backed up regularly and not purged.
  - Quotas typically set to limit amount of disk space available.
- **Scratch directory**
  - Location specified in environment variable varies among resources but will include the term SCRATCH, e.g. $SCRATCH_DIR.
  - Use to temporarily store files produced during application runs.
  - Not backed up and routinely purged.
  - No quotas. Available space depends on cumulative use by all users.
- Archival storage
  - Must request through allocation process

Your XSEDE Compute Environment

- Your default XSEDE compute environment provides access to the compilers, directories, and software you will need to efficiently use your XSEDE resources.
- Customize it using **Modules**

Modules Package

- A command line interface used to configure the shell for an application. Two components:
  1. Modulefiles - contain configuration information
  2. Module command - interprets modulefiles
- Pre-written modulefiles available for compilers, mpi implementations
- Pre-written modulefiles available for common software, e.g. NAMD, GAMESS

Module Commands

Module command	Description
module avail [path...]	List all modulefiles available on the system.
module list	List the modulefiles currently loaded in the shell environment.
module help modulefile	Print help information for the **modulefile** specified in the argument.
module display modulefile	Display the changes made to the environment when the specified modulefile is loaded.
module load modulefile	Interpret the commands contained within the specified modulefile.
module switch modulefile1 modulefile2	Remove the environment changes made by **modulefile1** and make the changes specified in **modulefile2**.
module unload modulefile	Remove the environment changes made by **modulefile**.

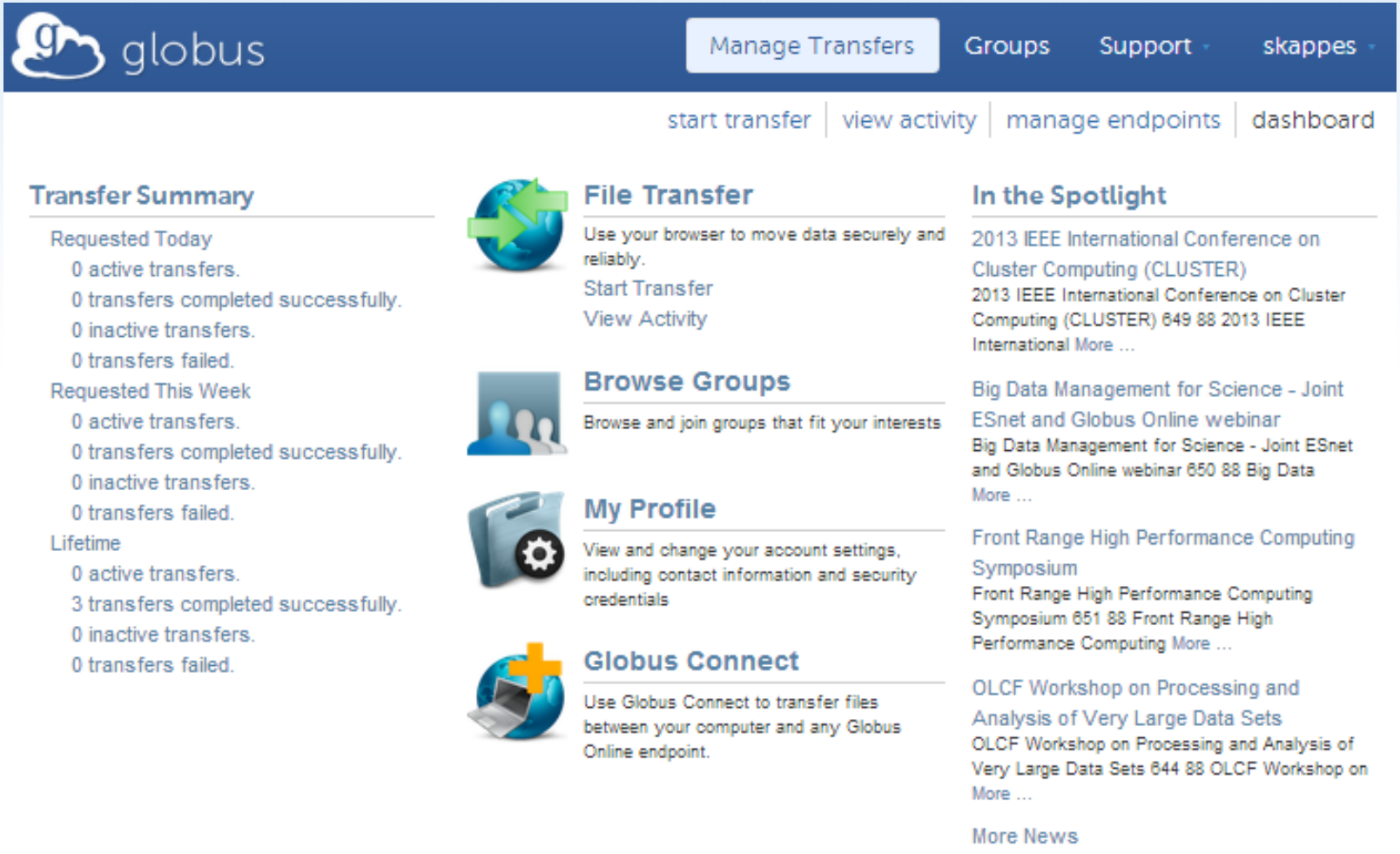
Module Commands Example

```
% module list
Currently Loaded Modulefiles:
  1) torque/2.3.13_psc   4) icc/14.0.0          7) globus/5.2.2
  2) mpt/2.04            5) imkl/10.3.3         8) xdusage/1.0-r7
  3) ifort/14.0.0        6) psc_path/1.0
% module avail gcc
------------------------- /usr/local/opt/modulefiles --------------------------
gcc/4.3.5 gcc/4.4.6 gcc/4.5.3 gcc/4.6.0 gcc/4.7.2 gcc/4.8.0 gcc/4.8.1
% module load gcc/4.8.1
% module list
Currently Loaded Modulefiles:
  1) torque/2.3.13_psc   5) imkl/10.3.3         9) mpfr/3.1.0
  2) mpt/2.04            6) psc_path/1.0       10) gmp/5.0.5
  3) ifort/14.0.0        7) globus/5.2.2       11) mpc/0.8.2
  4) icc/14.0.0          8) xdusage/1.0-r7     12) gcc/4.8.1
% module unload gcc
% module list
Currently Loaded Modulefiles:
  1) torque/2.3.13_psc   4) icc/14.0.0          7) globus/5.2.2
  2) mpt/2.04            5) imkl/10.3.3         8) xdusage/1.0-r7
  3) ifort/14.0.0        6) psc_path/1.0
```

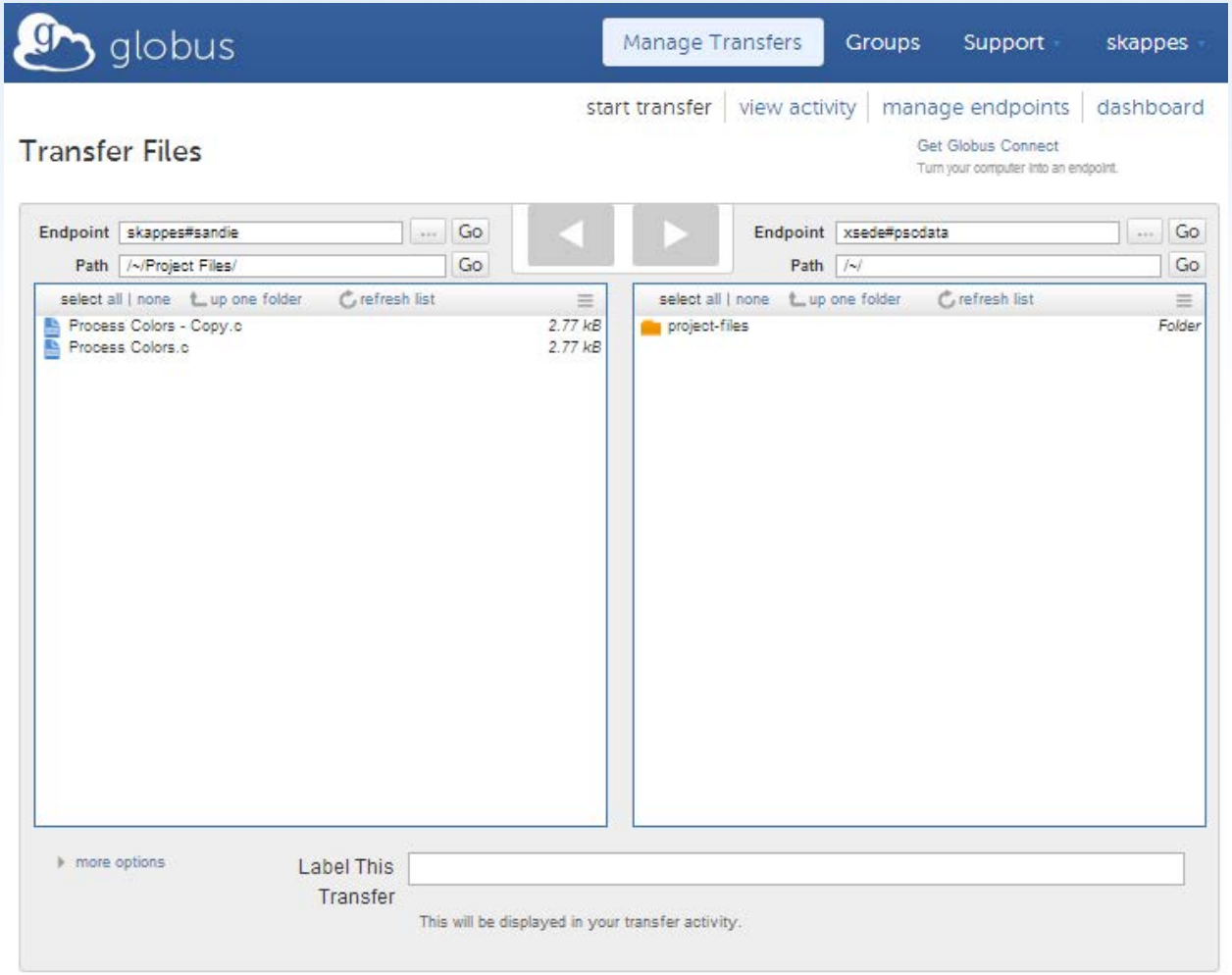
Moving Files - Globus Online

- A fast, reliable, and secure file transfer service geared to the big data needs of the research community.
- Moves terabytes of data in thousands of files
- Automatic fault recovery
- Easy to use
- No client software installation
- Consolidated support and troubleshooting
- Supports file transfer to any machine
- Accounts are free - www.globusonline.org

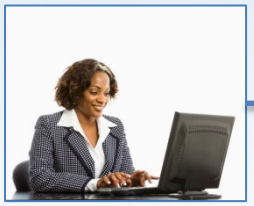
Globus Online Dashboard



Globus Online File Transfer



Running Jobs Overview



Researcher

Login

Command Line

When you connect to a resource, you are on a login node shared by many users.

Batch Script

Commands for code execution, copy input files to scratch,… Specify number/type of nodes, length of run, output directory, …

Data

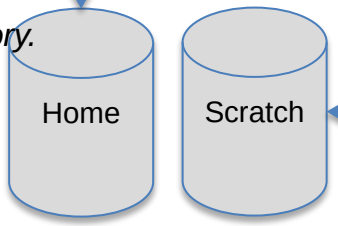
Job

Login Nodes

Use for tasks such as file editing, code compilation, data backup, and job submission.

Store project files Such as source code, scripts, and input data sets to Home directory.

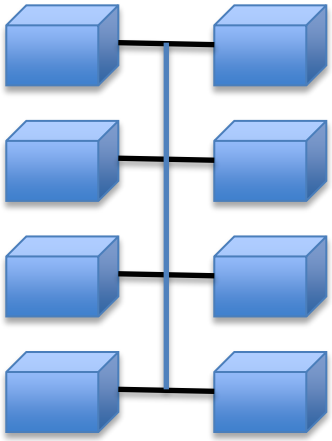
Run jobs by submitting your batch script to the compute nodes using the "qsub" command.



File System

Data

Read/write data from compute nodes to Scratch directory.



Compute Nodes

Your job is submitted to a queue and will wait in line until nodes are available. Queues are managed by a job scheduler that allows jobs to run efficiently.

Login nodes

- When you login, you are in the login node.
- Login nodes should only be used for basic tasks such as file editing, code compilation, data backup, and job submission.
- Login nodes should not be used to run production simulations. Production work should be performed on the system's compute resources.

Batch Jobs

- Compute jobs *cannot* be run on the login nodes.
- All XSEDE compute resources use some form of batch scheduler.
- There are several batch systems in use, but all work basically the same way. Create a job script specifying:
  - Number/type of nodes you need.
  - How long you need to run.
  - Where your output files should be written to.

Create a script

```
#!/bin/csh
#PBS -l ncpus=16
#ncpus must be a multiple of 16
#PBS -l walltime=5:00
#PBS -j oe
#PBS -q batch

set echo

ja

#move to my $SCRATCH directory
cd $SCRATCH

#copy executable to $SCRATCH
cp $HOME/mympi .

#run my executable
mpirun -np $PBS_NCPUS ./mympi

ja -chlst
```

Example script for running an MPI job on Blacklight at PSC.

Actual commands are site and machine specific, but they follow general principles.

Needs to be modified to run on other XSEDE machines.

Submitting/Manipulating Batch jobs

- Batch system should be used to run your job.
- Do not run on the login nodes.
- Submit the script that you have created:

Actual commands are machine specific, but they follow general principles.

qsub jobname

qstat –a

qstat -u username

qdel jobid

man qsub

Batch command examples

- **qsub amber.job**
- **qstat –a**

```
Job ID  Username Queue  Jobname  SessID NDS  Tasks Memory Time  S Time
------- -------- ------ -------- ------ ---- ----- ------ ----- - -----
29668   user1   batch   job2    21909   1    256   --   08:00  R    02:28
29894   user2   batch   run128   --     1    128   --   02:30    Q   --
29895   user3   batch   STDIN   15921   1     1   --   01:00  R 00:10
29896   user2   batch   jobL    21988   1   2048   --   01:00 R 00:09
29897   user4   batch   STDIN   22367   1     2   --   00:30   R 00:06
29898   user1   batch   amber    25188   1     1   --   01:10  R    00:00
```

- **qdel 29668**
- **After job 29898 runs: user1 should get file amber.job.o29898 with output/errors (log file)**

Why has my job not run?

- Never made it to the queue:

- Job not accepted by the queue:

Core requests on Kraken must be a multiple of twelve. You have requested an invalid number of cores (8). Please resubmit the job requesting an appropriate number of cores.

- Solution: Change the job script to request correct number of cores or memory for the resource.

My job did not complete

- Check the log files created job.e.89890
  job.o.89890
- One common problem: job run out of CPU time.
- Check the job script: time and memory requested, directory where you are writing files to.
- Do the input files exist in the directory where you specified?
- Do you have permission to use software?
- Waiting a very long time in the queue…
- If all fails… submit a ticket

Queue structure: job priority

- Job priority in the batch queues is based on the number of cores and wall clock time requested. Differs by site. Examples:

- Blacklight: *approx*. FIFO system. (Mechanisms in place to prevent a single user from dominating the batch queue and to prevent idle time on the machine).

- *Flexible time request* can improve your turnaround. *Packing small jobs*.

- Kraken: Priority to jobs that request large number of cores (over 32K processors) (except capability and dedicated jobs). Jobs with smaller core counts run on other systems (Trestles). However, they can run effectively on Kraken as *backfill*.

- *Backfill*: While the scheduler is collecting nodes for larger jobs, those with short wall clock limits and small core counts may use those nodes without delaying the start time of the larger job.

Queue structure: Backfill, Flexible time

- *Backfill*: While the scheduler is collecting nodes for larger jobs, those with short wall clock limits and small core counts may use those nodes without delaying the start time of the larger job.

- The system will not start a job that will not finish before the system maintenance time begins. Ex. Will run a 512 core jobs next. Waiting to finish a 256 core job that will take 4 more hours. Can run jobs that add up to 256 cores and will finish in 4 hours.

- To take advantage of this, request flexible walltime in your job script. A flexible walltime request can improve your job's turnaround in several circumstances.

Improving job turnaround

- Try to be as accurate as possible in estimating the walltime request for your job. Asking for more time than your job will actually need will almost certainly result in poorer turnaround for your job: Asking for the maximum walltime you can ask for a job will almost always result in poorer turnaround.

- Use flexible walltime

Improving job turnoaround: Flexible time

-l walltime_min=HH:MM:SS

-l walltime_max=HH:MM:SS

- Using flexible walltime limits increases the opportunity for your job to run on backfill blades.
- **Example:** if your job requests 64 cores and a range of walltime between 2 and 4 hours and a 64-core slot is available for 3 hours, your job could run in this slot with a walltime request of 3 hours. If your job had asked for a fixed walltime request of 4 hours it would not have been started.

Flexible time

- If the system starts one of your jobs with a flexible walltime request, it selects a walltime within the two specified limits. This walltime will not change during your job's execution. Can determine the walltime your job was assigned by

qstat –f $PBS_JOBID | grep Resource_List.walltime

- Your program should begin writing checkpoint files sufficiently in advance of the walltime so that the file writing is completed when the limit is reached. ***Save time to allow your job to transfer files after your program ends but before your job ends.***

timeout --timeout=$PROGRAM_TIME -- mpirun -np 32 ./mympi

Packing your jobs

- Running many small jobs places a great burden on the scheduler and is probably inconvenient for you.

- Pack many executions into a single job, which you then submit to PBS with a single qsub command.

Packing your jobs

Run each program execution in the background and place a wait command after each execution. Sample job to pack serial executions:

```
#!/bin/csh
#PBS -l ncpus=96
#PBS -l walltime=5:00
#PBS -q batch
dplace -c 0 ./myserial1 < serial1.dat &
dplace -c 32 ./myserial2 < serial2.dat &
dplace -c 64 ./myserial3 < serial3.dat &
wait
```

Packing your jobs, serial or MPI:

```
dplace -c 0 ./myserial1 < serial1.dat &
dplace -c 32 ./myserial2 < serial2.dat &
dplace -c 64 ./myserial3 < serial3.dat &
wait
```

- The dplace command insures that each execution will run on its own set of 32 cores. The executions will run concurrently.
- Same approach using the dplace command can be used to pack MPI executables.
-

Packing your jobs: OpenMP

- To pack OpenMP executables, replace the dplace command with the omplace command. Sample job to pack OpenMP executables:

- omplace -nt 32 -c 0 ./myopenmp1 < myopenmp1.dat &
- omplace -nt 32 -c 32 ./myopenmp2 < myopenmp2.dat &
- omplace -nt 32 -c 64 ./myopenmp3 < myopenmp3.dat &
- omplace -nt 32 -c 96 ./myopenmp4 < myopenmp4.dat &
- wait

Need help? Reporting and Tracking Issues

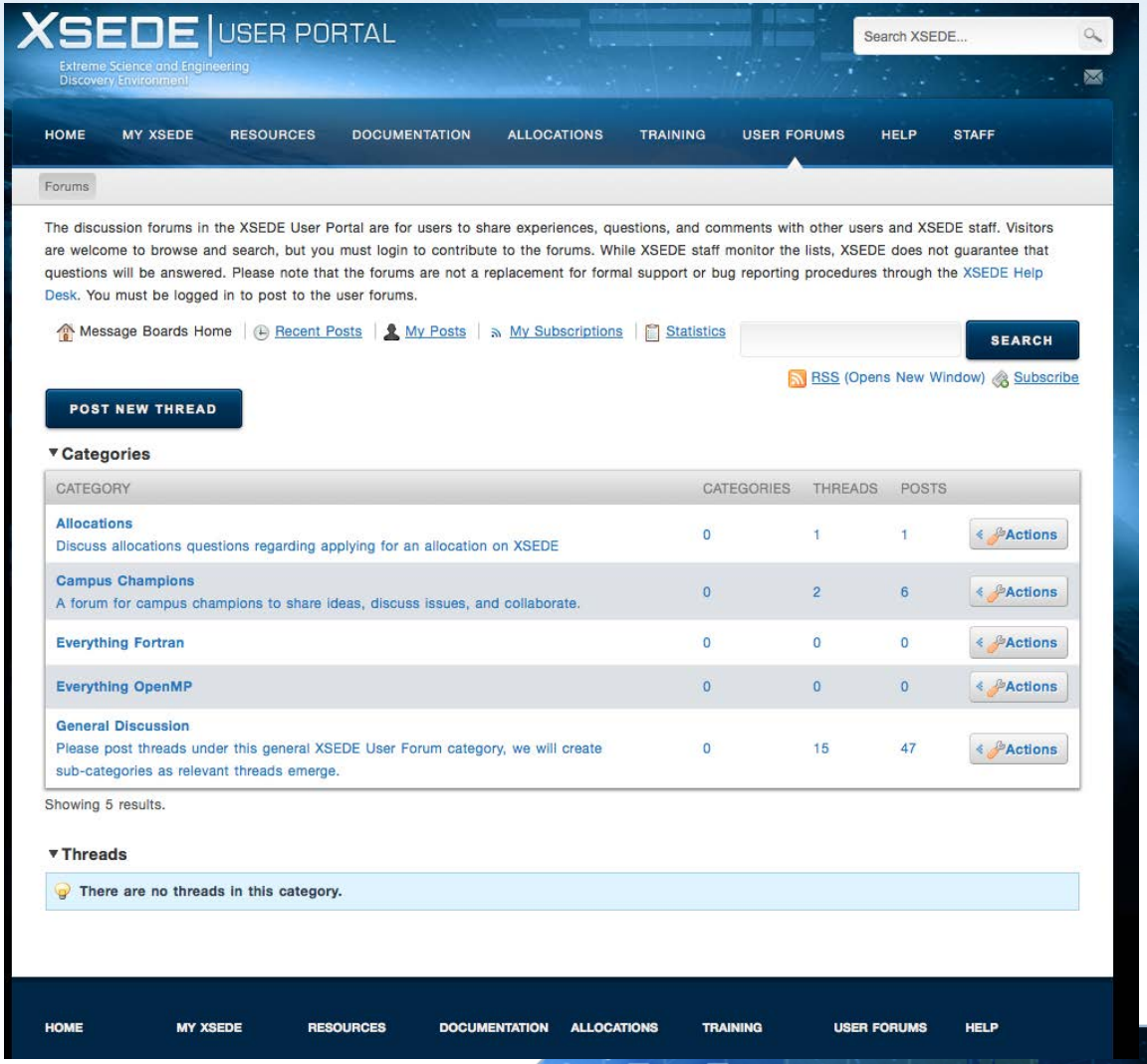
- portal.xsede.org → Help

  Submit ticket

- portal.xsede.org → My XSEDE → Tickets
  - Submit ticket
  - View past tickets (both open and closed)
- Can also email help@xsede.org or call 1-866-907-2383, at any hour (24/7)

Discussing your problems…
User Portal: User Forums

- The User Forums are a great place to ask questions, get help, or discuss ideas about XSEDE.



More "helpful" resources

xsede.org→User Services

- Resources available at each Service Provider
  - User Guides describing memory, number of CPUs, file systems, etc.
  - Storage facilities
  - Software (Comprehensive Search)
- Training: portal.xsede.org → Training
  - Course Calendar
  - On-line training
- Get face-to-face help from XSEDE experts at your institution; contact your local Campus Champions.
- Extended Collaborative Support (formerly known as Advanced User Support (AUSS))

September 10, 2014

Thanks for listening and welcome to XSEDE!

XSEDE

Extreme Science and Engineering Discovery Environment