Bridges-2 Early User Workshop

John Urbanic
Parallel Computing Scientist

PSC
PITTSBURGH SUPERCOMPUTING CENTER

Carnegie Mellon University

University of Pittsburgh

Overview

- Bridges-2 Mission and Team
- Early User Program
- System Configuration
- Using Bridges-2
- Support

PSC: A place for getting research done…

National service provider for research and discovery

- Giving the research community access to the world-class computational resources and facilities

BRIDGES-2 · XSEDE Extreme Science and Engineering Discovery Environment · NSF

Research institution advancing knowledge through converged research computing

- ~30 active funded projects

MMBioS · Center for Causal Discovery · HuBMAP

Education and training

- Lead national & local workshops
- Increase diversity in STEM
- Support courses at CMU and elsewhere

PSC IS A JOINT COMPUTATIONAL RESEARCH CENTER BETWEEN **CARNEGIE MELLON UNIVERSITY** AND THE **UNIVERSITY OF PITTSBURGH**.

33 years of leadership

(research computing, data analytics, and computational science)

21 HPC systems

(10 of which were the first or unique)

Pioneering the convergence of data, AI, and research computing.

Active member in the CMU and Pitt communities

- Research collaborations
- Colocation for lower cost and greater reliability & capability

Networking and security

- Networking & security service provider
- Research networking and security

3ROX THREE RIVERS OPTICAL EXCHANGE · TRUSTED CI THE NSF CYBERSECURITY CENTER OF EXCELLENCE · ResearchSOC

Advise and support industry

- Training, access to advanced resources, collaborative research

SCIVELO University of Pittsburgh

Carnegie Mellon University · PSC Pittsburgh Supercomputing Center · University of Pittsburgh


PITTSBURGH SUPERCOMPUTING CENTER
Hewlett Packard Enterprise
Bridges-2
NSF
Carnegie Mellon University
Bridges-2 Overview

Bridges-2

Ecosystem for Rapidly Evolving, Data-Intensive Science & Engineering

BRIDGES

Converged HPC, AI, and Big Data

A PSC RESOURCE

Connecting new communities to advanced research computing.

Converged computation and data ecosystem to empower users to explore new ways of doing computing.

Pioneered HPC+AI+Big Data
Bridges-AI expansion
Intel OPA first installation
Has become an ecosystem

2,100	projects
16,000	users
800	institutions
122	fields of study
130	education allocations

Award OAC-1445606

Hewlett Packard Enterprise

intel

NVIDIA

For more information: https://www.psc.edu/bridges

BRIDGES-2

Converged HPC, AI & Big Data for Rapidly-Evolving Research

A PSC RESOURCE

Provides transformative capability for rapidly evolving, computation-intensive and data-intensive research, creating opportunities for collaboration and convergence research.

More Science:	Approximately 3x larger than Bridges
Faster Computing:	Latest AMD EPYC processors
Faster Storage:	Fast flash array and tiered data management
Smarter Science:	Designed for Full System AI and data-centric computing
Scalable:	Interoperability with cloud and campus resources

Award OAC-1928147

Hewlett Packard Enterprise is delivering *Bridges-2*

For more information: https://www.psc.edu/bridges-2

Carnegie Mellon University

PSC PITTSBURGH SUPERCOMPUTING CENTER

University of Pittsburgh

Edward Hanna
Associate Director of Operations

Shawn Brown
Principal Investigator

Paola Buitrago
Associate Director for AI & Data Science

Sergiu Sanielevici
Associate Director for User support

Amanda Slimick
Project Manager

Robin Scibek
Associate Director for Broader Impacts

Bridges-2 Leadership Team

Target Timeline

October 1, 2019	Award start date; preparatory activities begin • System and user environment, documentation, content, dissemination, etc. • Broadly invite researchers for the Early User Program
March 2020	XRAC proposals awarded for *Bridges*/*Bridges-AI*, extending into *Bridges-2*
June-July 2020	Accept initial round of XRAC proposals
Winter 2020	Delivery, installation, initial testing
January 2021	Early User Program, conclusion of Acceptance Testing
Mid February 2021	Start of *Bridges-2* Production Operations
Q1 2021	Transition from *Bridges* to *Bridges-2*

Early User Program

- The Bridges-2 Early User Program is an opportunity for you to port, tune and optimize your applications early, and make progress on your research.
- Starts at Noon on Tuesday January 12; ends at midnight Tuesday February 9.
- No service units will be charged during this period.
- We expect your timely feedback regarding any issues you encounter and any assistance you may need: see "Support" section at the end of this presentation.
- Our goal is for you to achieve real scientific progress during this period. At the conclusion of the EUP, we will request one or two slides from your team, summarizing this progress.
- In the final week of the EUP, we will ask you to complete a short survey regarding your experience.
- Afterwards, there will be a transparent transition. Only thing you should notice is that you will be charged.

Building on the Flexible Architecture of *Bridges*
Introducing Innovations to Scale AI and High-Performance Data Analytics

Bridges-2 provides transformative capability for rapidly-evolving, computationally-intensive and data-intensive research, supporting new and existing opportunities for collaboration and convergence research.

Bridges-2 supports traditional and nontraditional research communities and applications, integrate new technologies for converged, scalable HPC, AI, and data; prioritize researcher productivity and ease of use; and provide an extensible architecture for interoperation with complementary data-intensive projects, campuses, and clouds.

Bridges-2 core concepts:

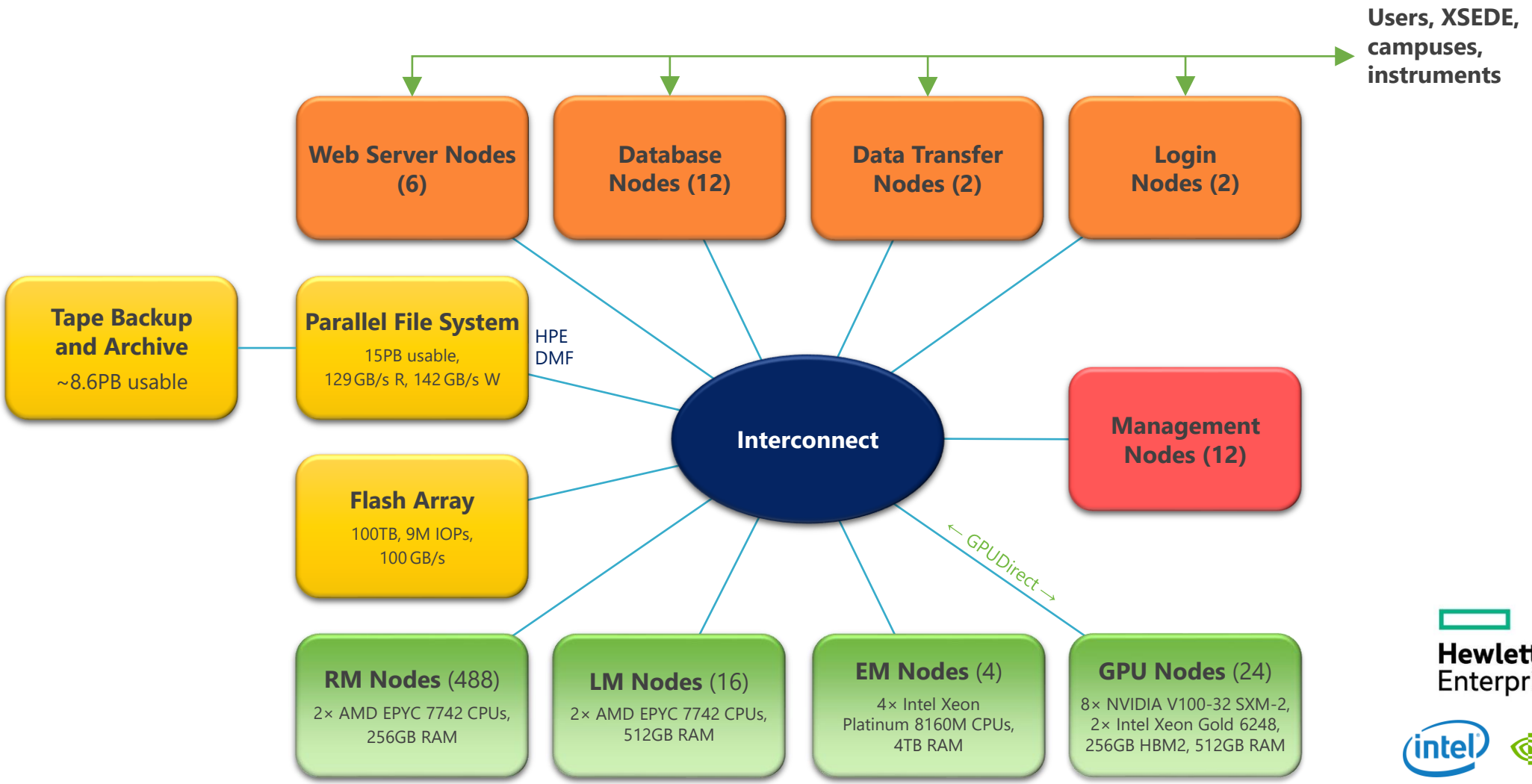
- Converged HPC + AI + Data
- Custom fat tree Clos topology optimized for data-centric HPC, AI, and HPDA
- Heterogeneous node types for different aspects of workflows
- CPUs and AI-targeted GPUs
- 3 tiers of per-node RAM (256GB, 512GB, 4TB)
- Extremely flexible software environment
- Community data collections & Big Data as a Service

Innovations beyond *Bridges*:

- AMD EPYC 7742 CPUs: 64-core, 2.25–3.4 GHz
- AI scaling to 192 V100-32GB SXM2 GPUs
- 100TB, 9M IOPs flash array accelerates deep learning training, genomics, and other applications
- Mellanox HDR-200 InfiniBand doubles bandwidth & supports in-network MPI-Direct, RDMA, GPUDirect, SR-IOV, and data encryption
- Cray ClusterStor E1000 Storage System
- HPE DMF single namespace across disk and tape for data security and expandable archiving

Carnegie Mellon University PSC PITTSBURGH SUPERCOMPUTING CENTER University of Pittsburgh

Bridges-2 High-Level Architecture



Hewlett Packard Enterprise

intel NVIDIA AMD

Carnegie Mellon University PSC PITTSBURGH SUPERCOMPUTING CENTER University of Pittsburgh

“Regular-Memory” (RM) and “Large-Memory” (LM) Nodes

Bridges-2 RM and LM nodes provide extremely powerful general-purpose computing and AI inferencing, with 128 cores per node and great memory bandwidth.

Each *Bridges-2* RM node contains:

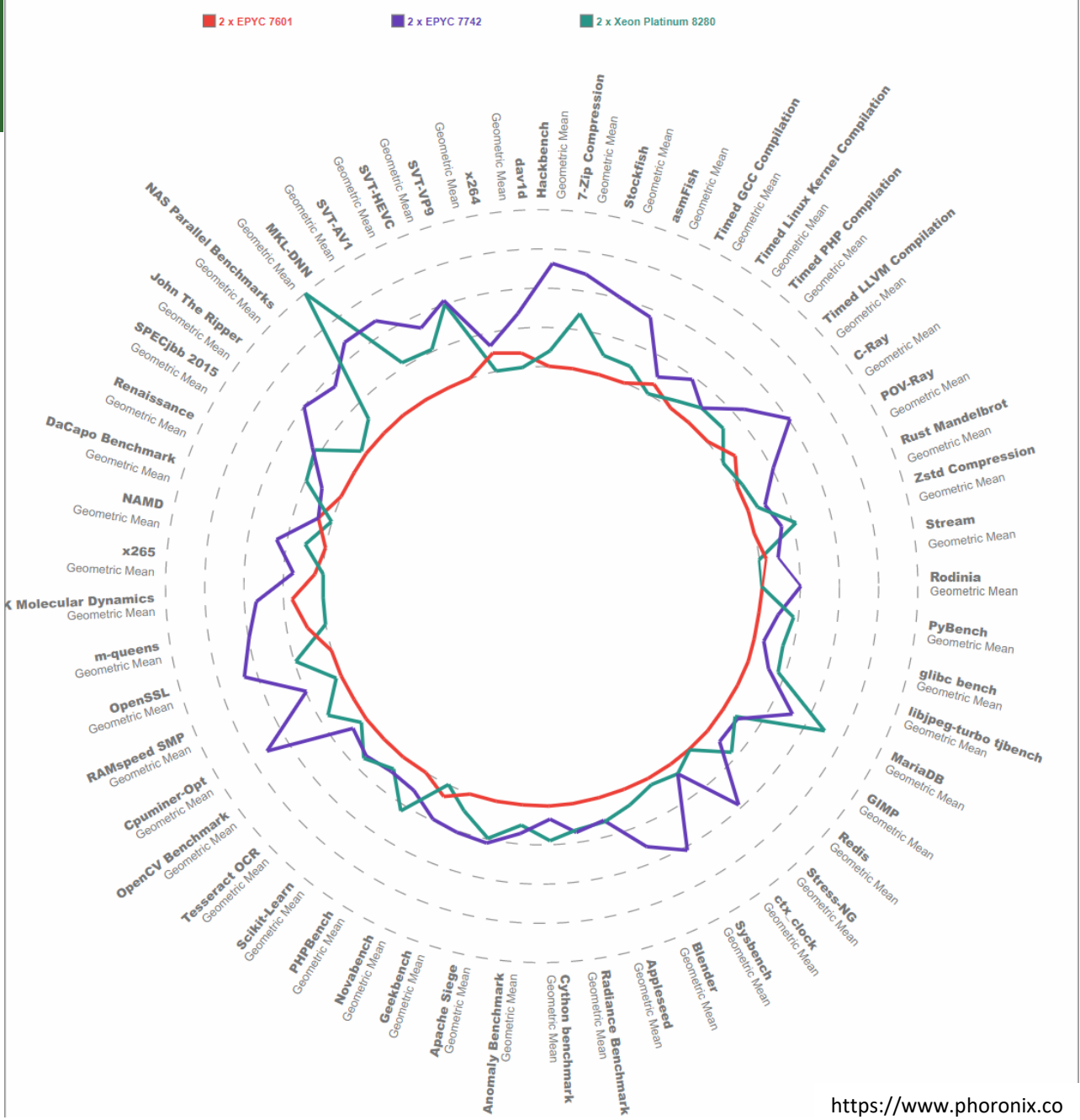
- 2× AMD EPYC “Rome” 7742 CPUs:
  - Each 64 cores, 128 threads, 2.25–3.40GHz, 256MB L3, 8 memory channels
- 488 RM nodes with 256GB of RAM
  16 LM nodes with 512GB of RAM
  - DDR4-3200 memory
- 3.84TB NVMe SSD
- Mellanox ConnectX-6 HDR InfiniBand 200Gb/s Adapter.

How about those EPYCs?

For those of you new to EPYCs, they are widely regarded as more than competitive with their Xeon brethren. They do not have AVX-512, but instead double the speed of AVX2.

This suite is from Phoronix. They also have an interesting, but a bit outdated, compiler comparison: GCC 9 and 10, Clang and AOCC.

Tomorrow is our last XSEDE multi-core workshop on Bridges-1. I am really looking forward to our first such event on Bridges-2 EPYCs.



https://www.phoronix.com/scan.php?page=news_item&px=AMD-EPYC-7742-2P-Summary

“Extreme-Memory” (EM) Nodes

Bridges-2 EM nodes provide additional memory for applications such as genome sequence assembly and graph analytics.

- Each *Bridges-2* EM node contains:
  - 4× Intel Xeon Platinum 8260M “Cascade Lake” CPUs:
    - Each 24 cores, 48 threads, 2.40–3.90GHz, 35.75MB LLC, 6 memory channels
  - 4TB of RAM: DDR4-2933
  - 7.68TB NVMe SSD
  - Mellanox ConnectX-6 HDR InfiniBand 200Gb/s Adapter.

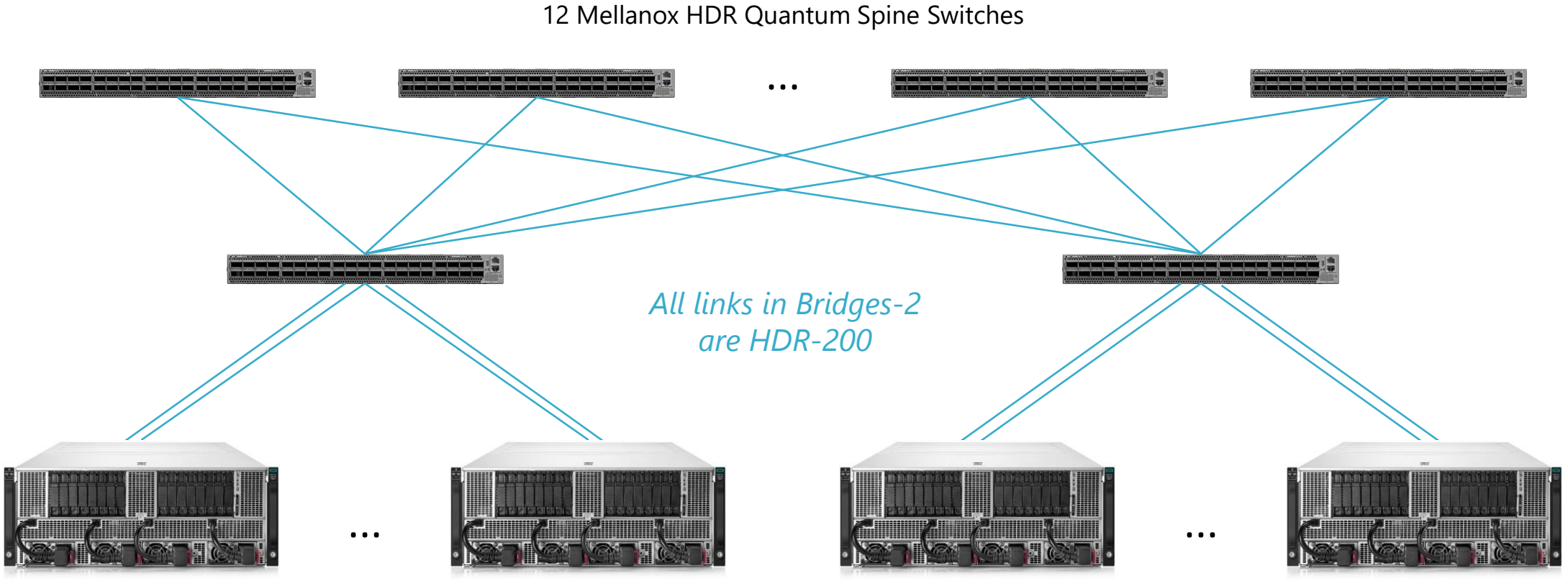
GPU Nodes

Bridges-2 GPU nodes provide exceptional performance and scalability for deep learning and accelerated computing.

Each *Bridges-2* GPU node contains:

- 8× NVIDIA Tesla V100-32GB SXM2 GPUs:
  - 40,960 CUDA cores and 5,120 tensor cores;
    1 Pf/s tensor, 125 Tf/s 32b, 64 Tf/s 64b
- 2× Intel Xeon Gold 6248 “Cascade Lake” CPUs:
  - 20 cores, 40 threads, 2.50–3.90GHz,
    27.5MB LLC, 6 memory channels
- 512GB of RAM: DDR4-2933
- 7.68TB NVMe SSD
- 2× Mellanox ConnectX-6 HDR InfiniBand 200Gb/s Adapter.

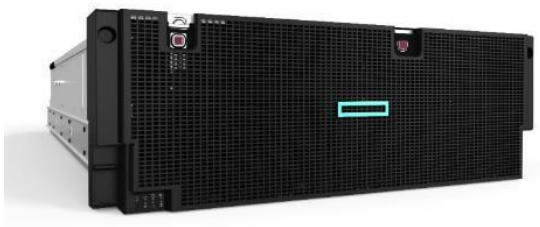
GPUDirect RDMA



Data Infrastructure

Bridges-2 filesystem (b2fs): Managed by HPE Data Management Framework (DMF) to provide a single namespace and user-friendly, rule-based migration.

ClusterStor E1000



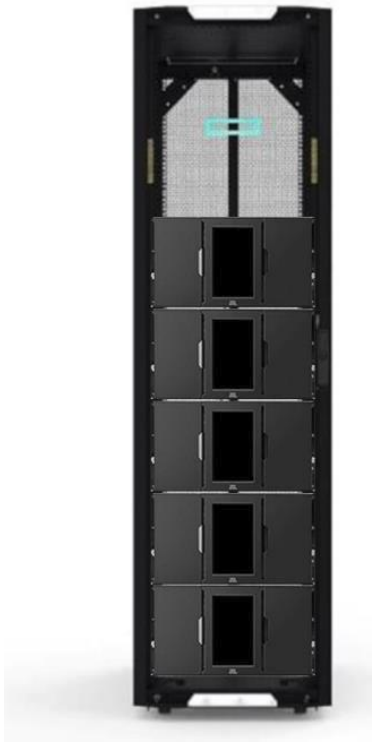
Lustre filesystem

- 15 PB usable, 21 PB raw
- 129 GB/s read, 142 GB/s write
- RAIDZ2
- 10 data server pairs, each serving 2.1 PB (raw)
- To be allocated through XSEDE

Flash Array

- >100 TB usable, 9M IOPs
- Use cases: training on large data, genomics, databases

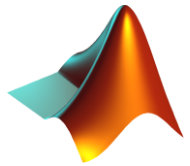
HPE StoreEver MSL6480 Tape Library



- 5 modules (scalable to 7); 80 LTO-8 Type M tape slots per module
- 7.2 PB uncompressed, ~8.6 PB compressed
- 50 TB/hour
- Use cases: archiving, disaster recovery
- To be implemented as a resource to be allocated through XSEDE
- Option for external groups to fund project-specific expansion

Using Bridges-2

- Interactivity
- Popular languages and frameworks:
  Python, Anaconda, R, MATLAB, Java, Spark, Hadoop
- AI frameworks: TensorFlow, Caffe2, PyTorch, etc.
- Containers and virtual machines (VMs)
- Databases
- Gateways and distributed (web) services
- Large collection of applications and libraries








Using Bridges-2

- Connecting
- Account Management
- File Spaces
- Transferring Files
- Programming Environment
- Software
- Running Jobs
- Containers, Datasets, Gateways
- Support

Connecting

- Login Node: managing files, submitting batch jobs and launching interactive sessions. Don't try to compute here.
  - If any question, look at the command prompt.

- ssh: I recommend just bridges2.psc.edu on the default port (22).
  - You can use your XSEDE credentials via port 2222 (with single sign on and DUO Multi-Factor Authentication).

- OnDemand? We are installing it as per Bridges 1. Let us know if you require it.

- Can use Public-private keys (but have to register keys via form).

Accounts

- Passwords
  - Use the web-based PSC password change utility
  - Use the kpasswd command when logged into a PSC system. Do not use the passwd command.
- Managing your allocation: all via the XSEDE Portal
- Change your shell: change_shell newshell
  - `change_shell -l` will show you options
- File retention: 3 months after your grant expires

File Spaces

- $HOME, your home directory on Bridges-2.
- $PROJECT, persistent file storage on Ocean.
- $LOCAL, Node-local scratch storage on the local SSD on the node running a job.

Permissions

- chmod: as per normal Unix practice

- ACLs: Finer grained control.
  - Ex: setfacl -m user:janeuser:r filename
  - Create any groups you want
  - man pages for setfacl and getfacl

$HOME

- The usual location for your batch scripts, source code and parameter files.
- Path is /jet/home/username.
- You can refer to your home directory with the environment variable $HOME.
- Your home directory is visible to all of Bridges-2's nodes.
- Backed up daily.
- If you need to recover a home directory file from backup send email to help@psc.edu. The process of recovery will take 3 to 4 days.
- 25GB quota. You can check your home directory usage using the quota command or the command du -sh.

$PROJECT (Ocean)

- Path is /ocean/project/groupname/username.
- groupname is the Unix group id associated with your grant.
  - Use the id command to find your group name.
  - The command id -Gn will list all the Unix groups you belong to.
  - The command id -gn will list the Unix group associated with your current session.
- If you have more than one grant, you will have an Ocean directory for each grant. Be sure to use the appropriate directory when working with multiple grants.
- Your quota for each of your grants is the Ocean storage allocation you received when your proposal was approved. If your total use in Ocean exceeds this quota you won't be able to run jobs on Bridges-2 until you are under quota again.
- Use the du -sh or projects command to check your Ocean usage. You can also check your usage on the XSEDE User Portal.
- If you have multiple grants, it is very important that you store your files in the correct Ocean directory.

$LOCAL

- Each of Bridges-2's nodes has a local file system attached to it. This local file system is only visible to the node to which it is attached. The local file system provides fast access to local storage.

- If your application performs a lot of small reads and writes, then you could benefit from using this space.

- Node-local storage is only available when your job is running, and can only be used as working space for a running job. Once your job finishes your local files are inaccessible and deleted. To use local space, copy files to it at the beginning of your job and back out to a persistent file space before your job ends.

- If a node crashes all the node-local files are lost. You should checkpoint theses files by copying them to Ocean during long runs.

Transferring Files

- Know the path to that space on Bridges-2. The start of the full paths for your Bridges-2 directories are:
  - Home directory /jet/home/*username*
  - Ocean directory /ocean/project/*groupname/username*
- You can use rsync, scp or sftp to copy files to and from Bridges-2.
  - scp *username@machine-name*:path/filename *username@machine-name*:path/filename where one of the machine names should be data.bridges2.psc.edu
  - rsync -rltpDvp -e 'ssh -l *username*' *source_directory* data.bridges2.psc.edu:*target_directory*
  - sftp *username@machine-name If bridges is the remote machine, use data.brigdes2.psc.edu. Then use put, get, etc.*
- IF you use TFA you must use XSEDE credentials.
- Globus also available. See PSC webpage for numerous details.
- From Bridges-1 to Bridges-2 we have a special *filemover* script. Available in the next few days.

Modules

- The environment management package *module* is essential for running software on most PSC systems.
- Check if there is a module for the software you want to use by typing *module spider software-name*.
- To load the environment for a software package type *module load software-name*.
- *module help software-name* lists any additional modules that must also be loaded.

Programming Environment

AMD (AOCC - AMD Optimizing C/C++ Compiler), Intel, Gnu and PGI compilers for C, C++ and Fortran are available on Bridges-2. The compiler commands are:

	C	C++	Fortran
AMD	clang	clang	flang
Intel	icc	icpc	ifort
Gnu	gcc	g++	gfortran
PGI	pgcc	pgc++	pgfortran

The best documentation is the official User Guide from each developer.

I will note that LLVM/clang, and hence AOCC, makes an effort to use the same flags as Gnu.

OpenMP

- Make sure you have loaded the compiler you desire.

Compiler	Option
Intel	-qopenmp for example: icc -qopenmp yprog.c
Gnu	-fopenmp for example: gcc -fopenmp myprog.c
PGI	-mp for example: pgcc -mp myprog.c

MPI

- Three types of MPI are supported on Bridges-2: MVAPICH2, OpenMPI and Intel MPI.

- To compile an MPI program, you must do these things:
  1. Load the module for the compiler environment that you want to use.
  2. Load the module for the MPI varient that you want to use.
  3. Issue the appropriate MPI wrapper command to compile your program.

Other Languages

- Many other languages are available, including Java, Python, R, and MATLAB. Many of these have multiple versions.
- Just load the appropriate module to obtain the environment.
- Some languages have more complex environments. We document these issues on the associated Software pages.

Software

Bridges-2 has a broad collection of applications installed.

- See the official list at https://www.psc.edu/resources/bridges-2/software/
- There are also useful user instructions there.
- Additional software may be installed by requ
  applications be installed by using the Softwa
- Most of these use the *module* command to e

Software	Description
Anaconda	Open data science platform, includes popular Python, R and SCALA packages
ANSYS	General purpose finite element modeling package
Aspera	Allows for rapid transfer of large files and data sets
ATLAS	ANSI C and Fortran77 interfaces for the entire BLAS API, and a some of the LAPACK AP
BAMTools	C++ API and toolkit for analyzing and managing BAM files
BCFTools	Utilities for variant calling and manipulating VCFs and BCFs
Bedops	Tools to address common questions raised in genomics studies
Bedtools	Utilities for comparing, summarizing and intersecting genomic features in the UCSC Genor format
Bismark	Bisulfite read mapper and methylation caller
BLAST	Search tool that finds regions of local similarity between nucleotide or protein sequences
BLAT	Alignment tool like BLAST, but structured differently
Boost	Peer reviewed C++ source libraries
Bowtie	Ultrafast, memory-efficient short read aligner. See related tool Bowtie2.
Bowtie2	Ultrafast, memory-efficient tool for aligning sequencing reads to long reference sequences tool Bowtie.
BWA	Fast light-weight tool that aligns relatively short sequences to a sequence database
C, C++	Programming languages
Caffe	Deep learning framework
CMake	Tools to control the compilation process, build, test and package software
CP2K	Atomistic simulations of solid state, liquid, molecular, periodic, material, crystal, and biolo

Running Jobs

There are 3 likely ways for you to run your jobs:

- Interactive sessions - where you type commands and receive output back to your screen as the commands complete.

- Batch mode - where you first create a batch (or job) script which contains the commands to be run, then submit the job to be run as soon as resources are available.

- OnDemand - a browser interface that allows you to run interactively, and also provides a graphical interface to tools like RStudio, Jupyter notebooks, and IJulia. More information is in the OnDemand section of the user guide.

interact

Interact is our own PSC command to make interactive access to the nodes simple. From the login node you just type "interact", and maybe some options.

It is very convenient and I use it by default for much of my debugging and development.

Option	Description	Default value
`-p partition`	Partition requested	RM-small
`-t HH:MM:SS`	Walltime requested The maximum time you can request is 8 hours.	60:00 (1 hour)
`-N n`	Number of nodes requested	1
`--ntasks-per-node=n` `Note the "--" for this option`	Number of cores to allocate per node	1
`-n NTasks`	Number of tasks spread over all nodes	N/A
`--gpu:n` `Note the "--" for this option`	Specifies the number of GPUs requested. Valid choices for *'n'* are 1-8	N/A
`-A account id`	SLURM account id for the job Find or change your default account id Note: Files created during a job will be owned by the Unix group in effect when the job is submitted. This may be different than the account id for the job. See the discussion of the `newgrp` command in the Account Administration section of this User Guide to see how to change the Unix group currently in effect.	Your default account id
`-R reservation-name`	Reservation name, if you have one Use of -R does not automatically set any other interact options. You still need to specify the other options (partition, walltime, number of nodes) to override the defaults for the interact command. If your reservation is not assigned to your default account, then you will need to use the -A option when you issue your interact command.	No default
`-h`	Help, lists all the available command options	N/A

Batch Jobs

GPU partitions	
Node description	8 NVIDIA Tesla V100 GPUs with 32GB of GPU memory each 2 Intel “Cascade Lake” CPUs with 20 cores each
Default number of nodes	1
Max nodes/job	
Default number of GPUs	8
Max GPUs/job	64
Default runtime	
Max runtime	2 days

- A “Bridges-2 GPU” allocation allows you to use Bridges-2’s GPU nodes. The GPU partition handles jobs for these nodes.

Slurm

Slurm is quite powerful, and it benefits you to take a look through the official documentation. However many of you will get by with just submitting jobs with *sbatch*, monitoring with *squeue* and occasionally canceling one with *scancel*.

Option	Description	Default
`-p partition`	Partition requested	RM
`-t HH:MM:SS`	Walltime requested in HH:MM:SS	30 minutes
`-N n`	Number of nodes requested.	1
`-o filename`	Save standard out and error in *filename*. This file will be written to the directory that the job was submitted from.	slurm-*jobid*.out
`--ntasks-per-node=n` `Note the "--" for this option`	Request n cores be allocated per node.	1
`--gpu:n` `Note the "--" for this option`	Specifies the number of GPUs requested. 'n' is the number of GPUs. Valid choices are 1-8.	None
`-A account id`	SLURM account id for the job. If not specified, your default account id is used. Find your default SLURM account id.	

Likewise, many of you will be happy just using the basic functionality for your jobscripts. There are many more options.

Slurm Sample RM job

What you probably really want is a sample jobscript to use as a template. I'll step through some of these here, but you should use the User Guide for your source as we will have more there, with better documentation.

```
#!/bin/bash
#SBATCH -N 2
#SBATCH -p RM
#SBATCH -t 5:00:00
#SBATCH --ntasks-per-node=128

#type 'man sbatch' for more information and options
#this job will ask for 2 full RM nodes(256 cores) for 5 hours
#this job would potentially charge 1280 RM SUs

#echo commands to stdout
set -x

# move to working directory
# this job assumes:
# - all input data is stored in this directory
# - all output should be stored in this directory
# - please note that groupname should be replaced by your groupname
# - username should be replace by your username

cd /ocean/projects/groupname/username/path-to-directory

#run pre-compiled program which is already in your project space
./runner.out
```

Slurm Sample RM-shared job

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -p RM-shared
#SBATCH -t 5:00:00
#SBARCH --ntasks-per-node=64

#type 'man sbatch' for more information and options
#this job will ask for 64 cores in RM-shared and 5 hours of runtime
#this job would potentially charge 320 RM SUs

#echo commands to stdout
set -x

# move to working directory
# this job assumes:
# - all input data is stored in this directory
# - all output should be stored in this directory
# - please note that groupname should be replaced by your groupname
# - username should be replace by your username

cd /ocean/projects/groupname/username/path-to-directory

#run pre-compiled program which is already in your project space

./a.out
```

Slurm Sample MPI job

```
#!/bin/bash
#SBATCH -N 2
#SBATCH -p RM
#SBATCH -t 5:00:00
#SBATCH --ntasks-per-node=128

#echo commands to stdout
set -x

# move to working directory
# this job assumes:
# - all input data is stored in this directory
# - all output should be stored in this directory
# - please note that groupname should be replaced by your groupname
# - username should be replace by your username

cd /ocean/projects/groupname/username/path-to-directory

#Make sure to load the appropriate MPI modules FOR YOU. These are only an example.
module load gcc/9.2.0
module load openmpi/gcc/4.0.2rc3

#run pre-compiled MPI program which is already in your project space
mpirun -np $SLURM_NTASKS ./mpia.out
```

Slurm Sample GPU job

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -p GPU
#SBATCH -t 5:00:00
#SBATCH --gpu=8

#type 'man sbatch' for more information and options
#this job will ask for 1 full GPU node(8 V100 GPUs) for 5 hours
#this job would potentially charge 40 GPU SUs

#echo commands to stdout
set -x

# move to working directory
# this job assumes:
# - all input data is stored in this directory
# - all output should be stored in this directory
# - please note that groupname should be replaced by your groupname
# - username should be replace by your username

cd /ocean/projects/groupname/username/path-to-directory

#run pre-compiled program which is already in your project space
#may want to load some particular CUDA module

./gpua.out
```

Slurm is for you, not against you.

Slurm exists to aid your throughput - not just add an additional step to getting your work done. We structure the queues to facilitate that objective, and tune them based on your feedback. If we do that well, you just submit and don't notice.

That is why I am not going to go into the specific queues behind the scenes. For example, we will have small and short high-priority queue to enable your debugging and development even during busy periods.

For very demanding or specific requirements we can create *reservations* on request.

Debugging with DDT

- Arm DDT is a debugging tool for C, C++ and Fortran 90 threaded and parallel codes. It is client-server software. Install the client on your local machine and then you can access the GUI on Bridges-2 to debug your code.

- There is an X-Windows option. I recommend against it.

Containers

- Singularity is the only container software supported on Bridges-2.
- You can create a Singularity container, copy it to Bridges-2 and then execute your container on Bridges-2.
- In your container you can use any software required by your application: a different version of CentOS, a different Unix operating system, any software in any specific version needed.
- You can install your Singularity container without any intervention from PSC staff.
- See the PSC documentation on Singularity for more details.

Containers on our shelf

We have installed many containers from the NVIDIA GPU Cloud (NGC) on Bridges-2. These containers are fully optimized, GPU-accelerated environments for AI, machine learning and HPC. They can only be used on the Bridges-2 GPU nodes. Including:

- *Caffe and Caffe2*
- *Microsoft Cognitive Toolkit*
- *DIGITS*
- *Inference Server*
- *MATLAB*
- *MXNet*
- *PyTorch*
- *Tensorflow*
- *TensorRT*
- *Theano*
- *Torch*

Again, see the PSC documentation on Singularity for a current list.

Public Datasets

A community dataset space allows Bridges-2's users from different grants to share data in a common space. Bridges-2 hosts both community (public) and private datasets.

Public datasets include:

- **2019nCoVR: 2019 Novel Coronavirus Resource**
- **COCO**
- **PREVENT-AD**
- **ImageNet**
- **Natural Language Tool Kit Data**
- **BLAST**
- **Homer**
- **MetaPhlAn2**

If there is a dataset which you need for your research on Bridges-2 and you believe it would be of widespread interest, you can request that it be installed for public use by using the Dataset Request form.

Gateways

Bridges-2 hosts a number of gateways – web-based, domain-specific user interfaces to applications, functionality and resources that allow users to focus on their research rather than programming and submitting jobs. Gateways provide intuitive, easy-to-use interfaces to complex functionality and data-intensive workflows.

Gateways can manage large numbers of jobs and provide collaborative features, security constraints and provenance tracking, so that you can concentrate on your analyses instead of on the mechanics of accomplishing them.

Some of the gateways that will be transitioning to Bridges-2 are *SciGaP, ChemCompute, SEAGrid, Future Water Indiana, SnowVision, Neuroscience Gateway, SCEC, Galaxy, USCMS (LHC), AMP*.

Support

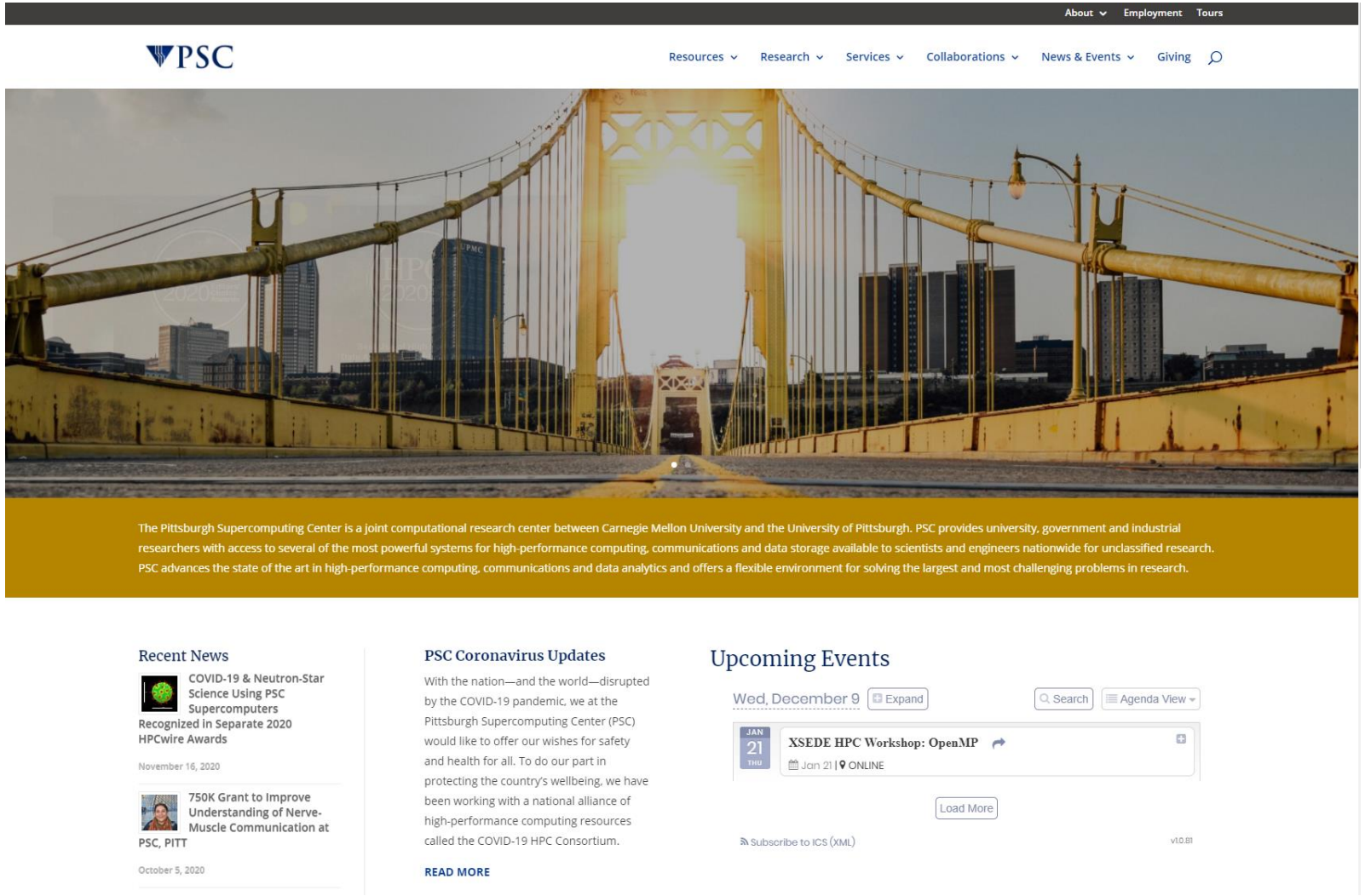
To report a problem on Bridges-2, please email help@psc.edu. Please report only one problem per email. Be sure to include:

- an informative subject line
- your username
- if the question concerns a particular job, include these in addition:
- the JobID
- any error messages you received
- the date and time the job ran
- link to job scripts, output and data files
- the software being used, and versions when appropriate
- a screenshot of the error or the output file showing the error, if possible

Office Hours

- We are adding a new dimension to our support with staff Office Hours.
- PSC staff in different domains will have dedicated office hours for you to drop in (probably via zoom).
- The logistics are being finalized currently.

For more information…



Website: www.psc.edu

Email us: help@psc.edu

Follow us on Twitter: @psc_live

Carnegie Mellon University PSC PITTSBURGH SUPERCOMPUTING CENTER University of Pittsburgh