

# Our Workshop Environment

John Urbanic  
Parallel Computing Scientist  
Pittsburgh Supercomputing Center

# Our Environment This Week

- Your laptops or workstations: only used for portal access
- Bridges is our HPC platform

We will here briefly go through the steps to login, edit, compile and run before we get into the real materials

We want to get all of the distractions and local trivia out of the way here. Everything *after* this talk applies to any HPC environment you will encounter.

20 Storage Building Blocks, implementing the parallel *Pylon* filesystem (~10PB) using PSC's SLASH2 filesystem

2 MDS nodes  
2 front-end nodes  
2 boot nodes

8 management nodes  
6 "core" Intel® OPA edge switches: fully interconnected, 2 links per switch

Intel® OPA cables

800 HPE Apollo 2000 (128GB) compute nodes

4 HPE Integrity Superdome X (12TB) compute nodes

42 HPE ProLiant DL580 (3 TB) compute nodes  
12 HPE ProLiant DL380 database nodes  
6 HPE ProLiant DL360 web server nodes

20 "leaf" Intel® OPA edge switches

32 RSM nodes with NVIDIA next-generation GPUs  
16 RSM nodes with NVIDIA K80 GPUs

Purpose-built Intel® Omni-Path topology for data-intensive HPC

<http://staff.psc.edu/nystrom/bvt>

# Bridges Node Types

| Type               | RAM <sup>a</sup> | Phse. | n   | CPU / GPU / other  | Server                    |
|--------------------|------------------|-------|-----|--|---------------------------|
| ESM                | 12TB             | 1     | 2   | 16 × Intel Xeon E7-8880 v3 (18c, 2.3/3.1 GHz, 45MB LLC)    | HPE Integrity Superdome X |
|                    |                  | 2     | 2   | 16 × TBA   |                           |
| LSM                | 3TB              | 1     | 8   | 4 × Intel Xeon E5-8860 v3 (16c, 2.2/3.2 GHz, 40 MB LLC)    | HPE ProLiant DL580 DL580  |
|                    |                  | 2     | 34  | 4 × TBA  |                           |
| RSM                | 128GB            | 1     | 752 | 2 × Intel Xeon E5-2695 v3 (14c, 2.3/3.3 GHz, 35MB LLC)     | HPE Apollo 2000           |
| RSM-GPU            | 128GB            | 1     | 16  | 2 × Intel Xeon E5-2695 v3 + 2 × NVIDIA K80                 |                           |
|                    |                  | 2     | 32  | 2 × Intel Xeon E5-2695 v3 + 2 × NVIDIA next-generation GPU |                           |
| DB-s               | 128GB            | 1     | 6   | 2 × Intel Xeon E5-2695 v3 + SSD                            | HPE ProLiant DL360 DL360  |
| DB-h               |                  |       | 6   | 2 × Intel Xeon E5-2695 v3 + HDDs                           | HPE ProLiant DL380 DL380  |
| Web                | 128GB            | 1     | 6   | 2 × Intel Xeon E5-2695 v3                                  | HPE ProLiant DL360 DL360  |
| Other <sup>b</sup> | 128GB            | 1     | 14  | 2 × Intel Xeon E5-2695 v3                                  | HPE ProLiant DL360, DL380 |
| Total              |                  |       |     |  |                           |

a. All RAM in these nodes is DDR4-2133

b. Other nodes = front end (2) + management/log (8) + boot (2) + MDS (4)

# Getting Connected

- The first time you use your account sheet, you must go to [apr.psc.edu](http://apr.psc.edu) to set a password. We will take a minute to do this shortly.
- We will be working on [bridges.psc.edu](http://bridges.psc.edu). Use an ssh client (a Putty terminal, for example), to ssh to the machine.
- At this point you are on a login node. It will have a name like “br001” or “br006”. This is a fine place to edit and compile codes. However we must be on compute nodes to do actual computing. We have designed bridges to be the world’s most interactive supercomputer. We generally only require you to use the batch system when you want to. Otherwise, you get your own personal piece of the machine. To get a single core use “interact”:

```
[urbanic@br006 ~]$ interact
```

```
[urbanic@r590 ~]$
```

- You can tell you are on a regular memory compute node because it has a name like “r590” or “r101”.
- Do make sure you are working on a compute node. Otherwise your results will be confusing.
- We will use several parameters with interact as we go. They will allow us to request different types of nodes (GPU for OpenACC or many cores for OpenMP, for example). In general, you can use the interact session you request for the rest of the day unless you need to request difference resources.

# Editors

For editors, we have several options:

- emacs
- vi
- nano: use this if you aren't familiar with the others

# Compiling

We will be using standard Fortran and C compilers this week. They should look familiar.

- pgcc for C
- pgf90 for Fortran

We will slightly prefer the PGI compilers (the Intel or gcc ones would also be fine for most of our work, but not so much for OpenACC). There are also MPI wrappers for these called mpicc and mpif90 that we will use. Note that on Bridges you would normally have to enable this compiler with

```
module load pgi
```

I have put that in the .bashrc file that we will all start with.



# Multiple Sessions

You are limited to one interactive compute node session for our workshop. However, there is no reason not to open other sessions (windows) to the login nodes for compiling and editing. You may find this convenient. Feel free to do so.



# Our Setup For This Workshop

After you copy the files from the instructor directory, you will have:

```
/Exercises
    /Test
    /OpenMP
        laplace_serial.f90/c
    /Solutions
    /Examples
    /Prime
    /OpenACC
    /MPI
```

# Preliminary Exercise

Let's get the boring stuff out of the way now.

- Log on to apr.psc.edu and set an initial password.

- Log on to bridges.

```
ssh username@bridges.psc.edu
```

- Copy the exercise directory from the training directory to your home directory, and then copy the workshop shell script into your home directory.

```
cp -r ~training/Exercises .  
cp ~training/.bashrc .
```

- Logout and back on again to activate this script. You won't need to do that in the future.
- Edit a file to make sure you can do so. Use emacs, vi or nano (if the first two don't sound familiar).
- Start an interactive session.

```
interact
```

- cd into your exercises/test directory and compile (C or Fortran)

```
cd Exercises/Test  
pgcc test.c  
pgf90 test.f90
```

- Run your program

```
a.out
```

(You should get back a message of "Congratulations!")