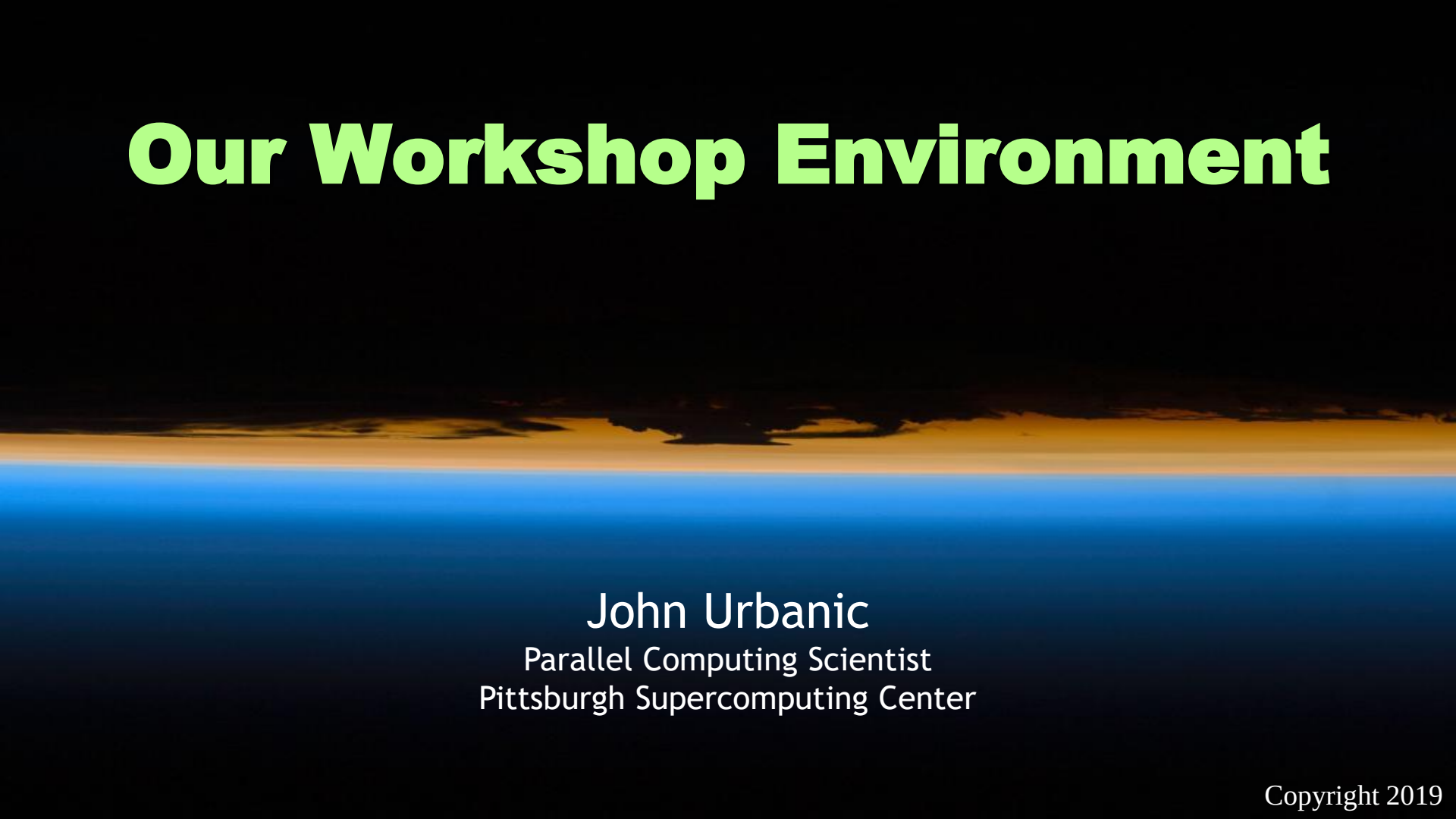
Our Workshop Environment



John Urbanic
Parallel Computing Scientist
Pittsburgh Supercomputing Center

Our Environment

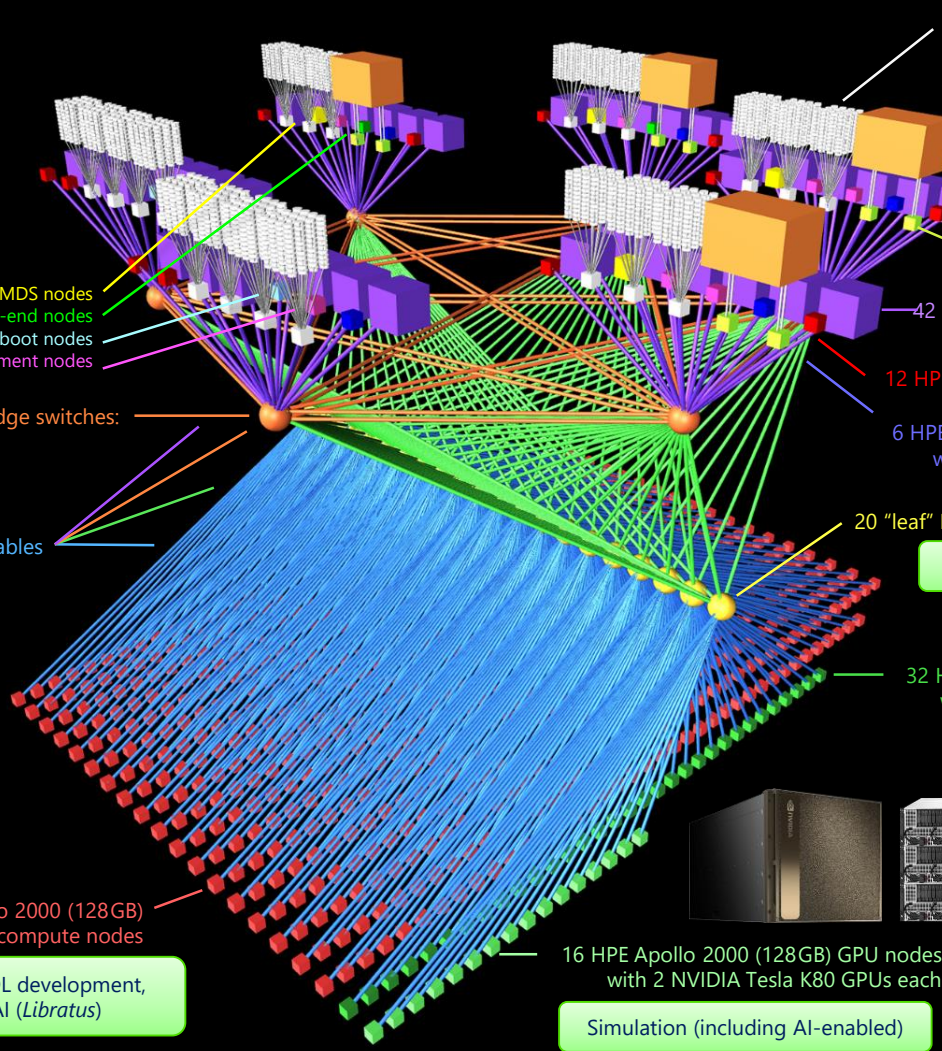
- Your laptops or workstations: only used for portal access
- Bridges is our HPC platform

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.

BRIDGES

A PITTSBURGH SUPERCOMPUTING CENTER RESOURCE



20 Storage Building Blocks, implementing the parallel Pylon storage system (10 PB usable)

Project & community datasets

4 HPE Integrity Superdome X (12TB) compute nodes …

… each with 2 gateway nodes

Large-memory Java & Python

4 MDS nodes
2 front-end nodes
2 boot nodes
8 management nodes

42 HPE ProLiant DL580 (3TB) compute nodes

12 HPE ProLiant DL380 database nodes
6 HPE ProLiant DL360 web server nodes

User interfaces for AIaaS, BDaaS

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

Intel® OPA cables

Robust paths to parallel storage

20 "leaf" Intel® OPA edge switches

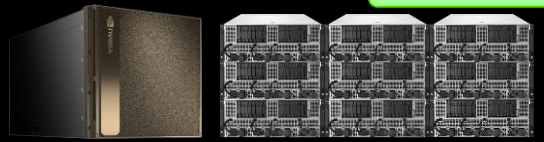
Distributed training, Spark, etc.

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

32 HPE Apollo 2000 (128GB) GPU nodes with 2 NVIDIA Tesla P100 GPUs each

Deep Learning

Maximum-Scale Deep Learning



NVIDIA DGX-2 and 9 HPE Apollo 6500 Gen10 nodes: 88 NVIDIA Tesla V100 GPUs

748 HPE Apollo 2000 (128GB) compute nodes

ML, inferencing, DL development, Spark, HPC AI (*Libratus*)

16 HPE Apollo 2000 (128GB) GPU nodes with 2 NVIDIA Tesla K80 GPUs each

Simulation (including AI-enabled)

Bridges Virtual Tour:
https://psc.edu/bvt

This tool depicts the current plan for PSC's Bridges HPC system. Details are subject to change.

Bridges Hardware

Bridges-DL

Type	RAM	#	CPU / GPU / SSD	Server
ESM	12 TB[b]	2	16 × Intel Xeon E7-8880 v3 (18c, 2.3/3.1 GHz, 45MB LLC)	HPE Integrity Superdome X
	12 TB[c]	2	16 × Intel Xeon E7-8880 v4 (22c, 2.2/3.3 GHz, 55MB LLC)	
LSM	3 TB[b]	8	4 × Intel Xeon E7-8860 v3 (16c, 2.2/3.2 GHz, 40 MB LLC)	HPE ProLiant DL580
	3 TB[c]	34	4 × Intel Xeon E7-8870 v4 (20c, 2.1/3.0 GHz, 50 MB LLC)	
RSM	128 GB[b]	752	2 × Intel Xeon E5-2695 v3 (14c, 2.3/3.3 GHz, 35MB LLC)	
RSM-GPU	128 GB[b]	16	2 × Intel Xeon E5-2695 v3 + 2 × NVIDIA Tesla K80	HPE Apollo 2000
	128 GB[c]	32	2 × Intel Xeon E5-2683 v4 (16c, 2.1/3.0 GHz, 40MB LLC) + 2 × NVIDIA Tesla P100	
GPU-AI16	1.5 TB[d]	1	16 × NVIDIA V100 32GB SXM2 + 2 × Intel Xeon Platinum 8168 + 8 × 3.84 TB NVMe SSDs	NVIDIA DGX-2 delivered by HPE
GPU-A8	192 GB[d]	9	2 × Intel Xeon Gold 6148 + 2 × 3.84 TB NVMe SSDs	HPE Apollo 6500 Gen10
DB-s	128 GB[b]	6	2 × Intel Xeon E5-2695 v3 + SSD	HPE ProLiant DL360
DB-h	128 GB[b]	6	2 × Intel Xeon E5-2695 v3 + HDDs	HPE ProLiant DL380
Web	128 GB[b]	6	2 × Intel Xeon E5-2695 v3	HPE ProLiant DL360
Other[a]	128 GB[b]	16	2 × Intel Xeon E5-2695 v3	HPE ProLiant DL360, DL380
Gateway	64 GB[b]	4	2 × Intel Xeon E5-2683 v3 (14c, 2.0/3.0 GHz, 35MB LLC)	HPE ProLiant DL380
	64 GB[c]	4	2 × Intel Xeon E5-2683 v3	
	96 GB[d]	2	2 × Intel Xeon	
Storage	128 GB[b]	5	2 × Intel Xeon E5-2680 v3 (12c, 2.5/3.3 GHz, 30 MB LLC)	Supermicro X10DRi
	256 GB[c]	15	2 × Intel Xeon E5-2680 v4 (14c, 2.4/3.3 GHz, 35 MB LLC)	
Total	286.5 TB	920		

a. Other nodes = front end (2) + management/log (8) + boot (4) + MDS (4)
b. DDR4-2133
c. DDR4-2400
d. DDR4-2666

Getting Connected

- The first time you use your account sheet, you must go to apr.psc.edu to set a password. We will take a minute to do this shortly.

- We will be working on 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 “login001” or “login006”. 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 GPU use “interact –gpu”:

```
[urbanic@login006 ~]$ interact -gpu
[urbanic@gpu016 ~]$
```

- However when we have many of you looking for very quick turnaround, we may fall back on the queuing system to help. We will keep it very simple today:

```
[urbanic@login006 ~]$ sbatch gpu.job
```

Or maybe not!

Depending on how many faces I am staring at, we may be able to use interact now. At any rate, it will probably by most convenient after the class.

However, as we want to assure snappiness, we reserved a slice of Bridges for all of our important sessions. You can request these special slices, that only you have access to, with the -R option on the interact command.

One minor detail is that these reservation are in flux this week. So, you will need to be aware of the currently active reservation. These are:

- Today: openacc1
- Tomorrow: openacc2
- All week (but limited): challenge

In other word, in our upcoming exercise session, you should try:

```
[urbanic@login006 ~]$ interact -gpu -R openacc1
[urbanic@gpu016 ~]$
```

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. They should look familiar.

- pgcc for C
- pgf90 for Fortran

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

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

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

- Run your program

```
sbatch gpu.job
```

(Wait a minute, or see how your job is doing with *squeue –u username*)

- Look at the results

```
more slurm-55838.out        (The exact job number will differ)
```

It should say *"Congratulations!"*

OR

interact -gpu -R openacc1