Introduction to HPC and Advanced Cyberinfrastructure (CI)

CIML Summer Institute

June 22, 2021

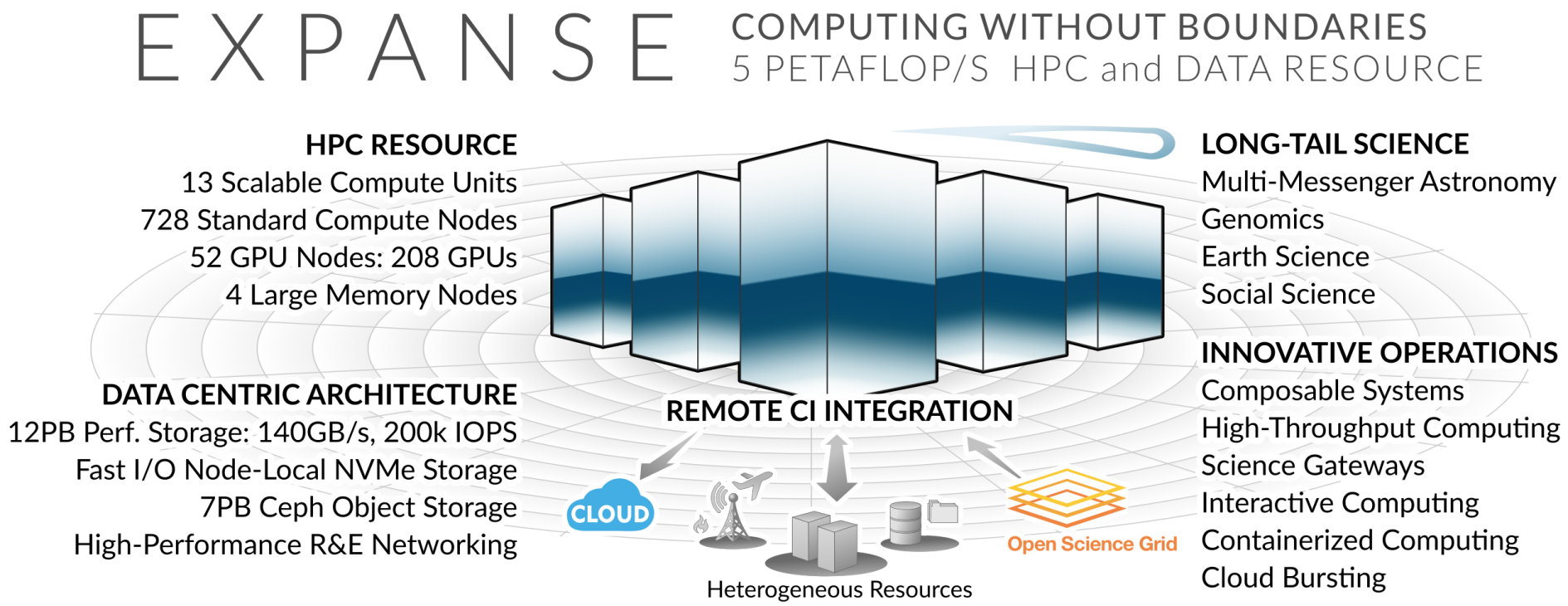
Robert Sinkovits, PhD

San Diego Supercomputer Center

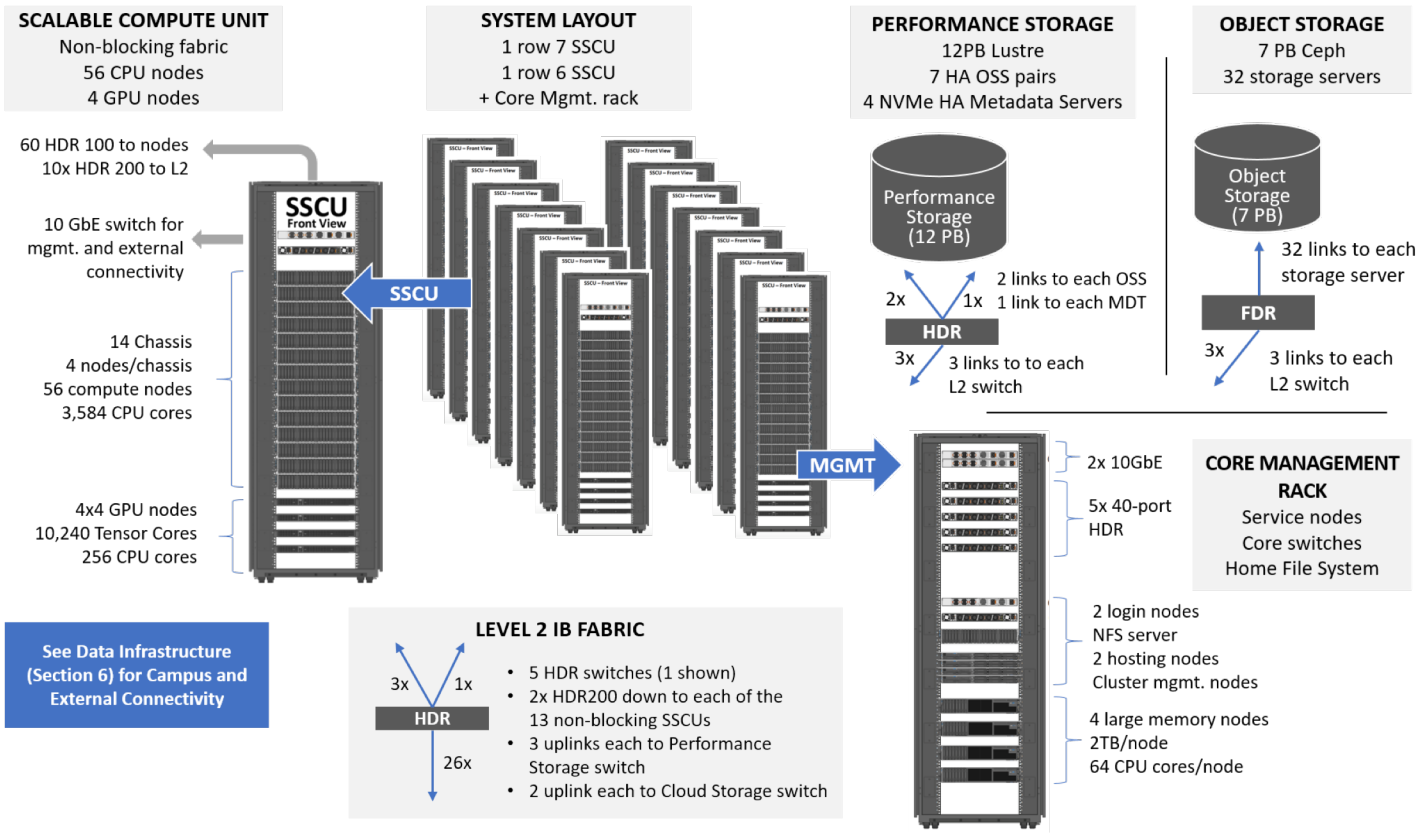
HPC and Advanced CI vs. non-HPC

- Most HPC systems are built using the same technology that goes into your non-HPC systems, with the main differences being scale and hardware grade
  - **Scale**: HPC systems can comprise O(100)-O(10,000) compute nodes; associated file systems are O(1)-O(100) petabytes
  - **Grade**: HPC systems are built using enterprise hardware, which meets higher standards for performance and reliability than consumer grade
- As a result, code that was developed on your local system using standard languages (C/C++, Java, Python) and libraries (TensorFlow, PyTorch) can be run on HPC systems
- Running at scale may require than you learn new parallel programming techniques or tools

Expanse – SDSC's primary HPC resources



Expanse architecture



Similarities and differences: HPC vs. desktop

Expanse standard compute nodes

System Component	Configuration
Compute Nodes	
CPU Type	*AMD EPYC 7742*
Nodes	728
Sockets	2
Cores/socket	64
Clock speed	2.25 GHz
Flop speed	4608 GFlop/s
Memory capacity	* 256 GB DDR4 DRAM
Local Storage	1TB Intel P4510 NVMe PCIe SSD
Max CPU Memory bandwidth	409.5 GB/s

CPUs mostly built by vendors you know (e.g., Intel and AMD)

Larger core counts (typical laptop CPU has 4-8 cores)

Clock speeds about the same

A lot more memory (vs. 8-16 GB)

Comparable local storage

Similarities and differences: HPC vs. desktop

Expanse GPU nodes

GPU Nodes	
GPU Type	NVIDIA V100 SMX2
Nodes	52
GPUs/node	4
CPU Type	Xeon Gold 6248
Cores/socket	20
Sockets	2
Clock speed	2.5 GHz
Flop speed	34.4 TFlop/s
Memory capacity	*384 GB DDR4 DRAM
Local Storage	1.6TB Samsung PM1745b NVMe PCIe SSD
Max CPU Memory bandwidth	281.6 GB/s

Much higher end GPU than you can probably afford

Lots of memory

Comparable local storage

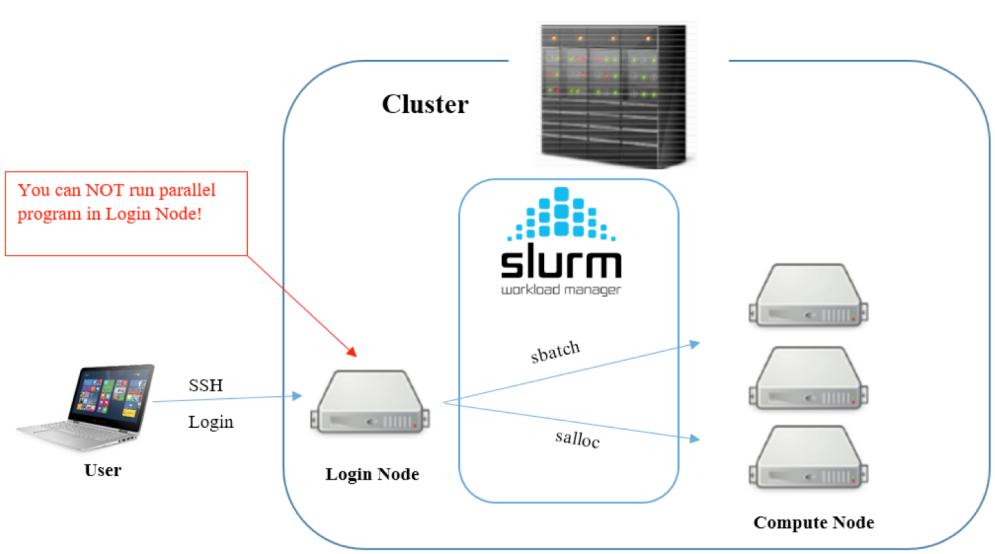
Using HPC and what you need to know

You'll be running on a shared resource and the rules are a little different than when working on your own dedicated resource

- Jobs are run using a batch scheduler
- Multiple file systems are available and your behavior can affect others
- Containers are supported, but there are some restrictions
- Your usage of the system will be limited by the size of your allocation
- User support does a lot to make your life easier

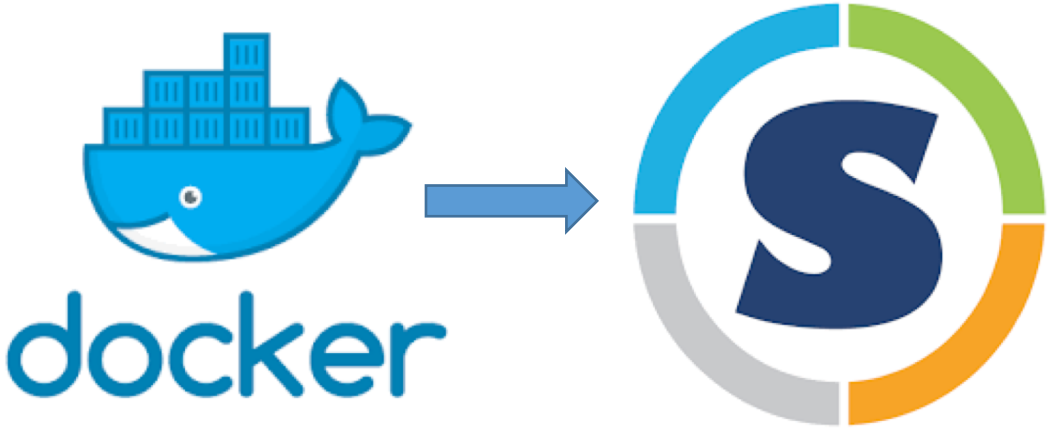
HPC difference #1 – you'll need to go through a scheduler

- The place where you login to the system is **not** where you will run your compute/data-intensive jobs
- Resources are allocated to your job through a scheduler, such as Slurm. Even when you're not explicitly writing batch scripts (e.g., Jupyter notebooks, interactive jobs, science gateways), this is being done behind the scenes



HPC difference #2 – you'll need to use containers differently

- Containers are great and you'll hear a lot more about them later today
- But you can't use Docker on most HPC machines because of security holes
- Fortunately, Docker containers that do not run services can easily be converted to secure Singularity containers



HPC difference #3 – play nice on the shared file systems

- Your home directory is NFS mounted. It is limited in size (100 GB) and should not be used for I/O intensive jobs
- Use Lustre file systems to read/write big data. But avoid writing large number of small files since this negatively impacts all users
- Take advantage of node-local scratch for temporary storage. Be sure to copy results to Lustre when you're done
- If you really need to work with lots of small files, consider storing as an archive (collection.tar) and untar into SSD
- More on file systems coming later today

HPC difference #4 – allocations vs. dedicated usage

- HPC systems are in heavy demand and the amount of usage is determined through a competitive allocations process. On most systems, awards are made separately for CPU time, GPU time and storage.
- Your allocation is debited for the resources you tie up, not the resources you use. To help you make the most of your allocation, many systems (including *Expanse*) support shared use of nodes.

HPC difference #5 – there's a lot of support

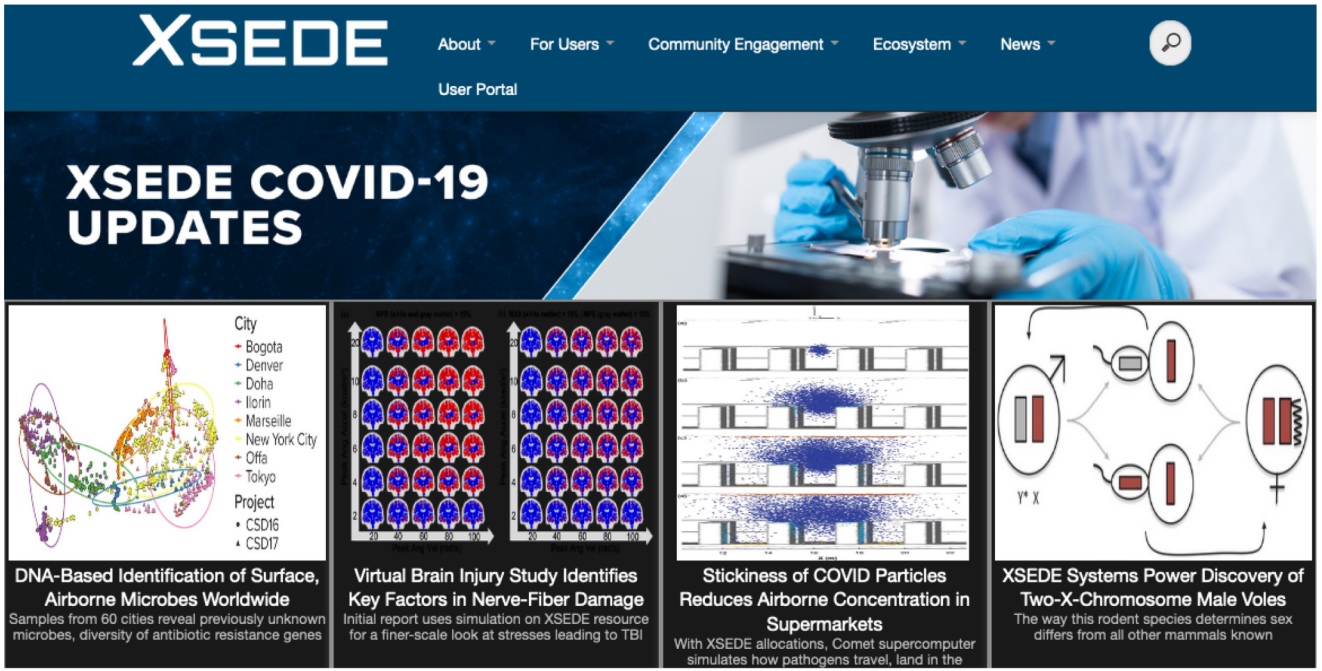
HPC systems are professionally administered

- User services maintains a comprehensive software stack of compilers, math libraries, deep learning frameworks and applications.
- Before you try to build your own application, see if there is already a module or container available. Not only will this save you effort, but the performance may be better since apps are built using best choice of compilers, flags and dependent libraries.
- Hardware is very thoroughly tested before deployment and regularly monitored to ensure it's functioning optimally.

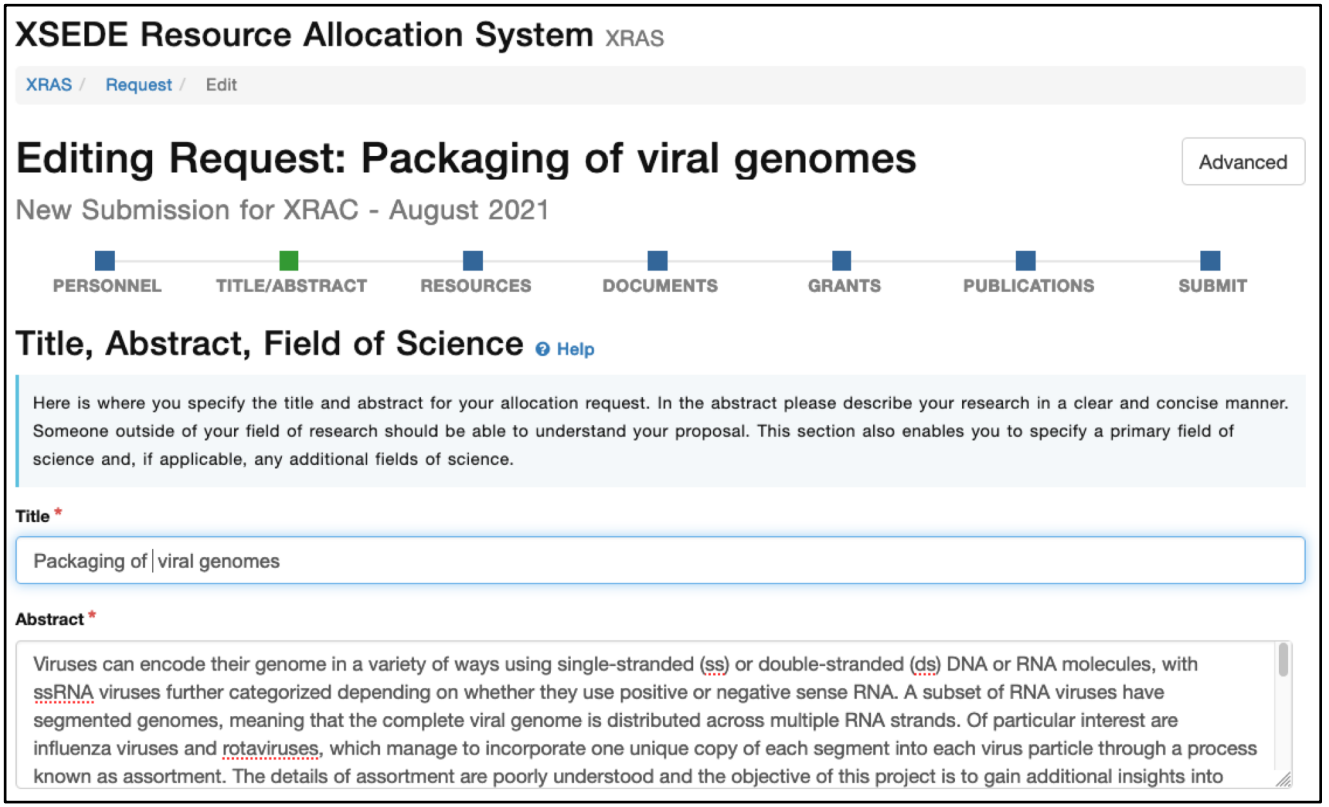
Wide variety of national allocated resources

- The NSF funds a number of large resources that are available to the U.S. educational and research communities and allocated through the XSEDE program. These include CPU, GPU and specialized AI hardware along with storage and networking.
- NSF also funds XSEDE to support these resources (more in coming slides)
- Students are not eligible to serve as PI on XSEDE awards, but can still get access through faculty advisor or mentor. Exception made for NSF Graduate Student Fellows and Honorable Mention recipients.
- Allocation types
  - Startup – small awards for new users or projects with modest CI needs
  - Education – support for educational and training activities
  - Research – larger awards for supporting mature projects; review 4 times/year

XSEDE's Mission: Substantially enhance the productivity of a growing community of scholars, researchers, and engineers through access to advanced digital services that support open research; and coordinate and add significant value to the leading cyberinfrastructure resources funded by the NSF and other agencies.



XSEDE allocations process



Resources allocated through XSEDE

Compute

Bridge2 GPU Artificial Intelligence
PSC Bridges-2 GPU-AI

Bridges-2
PSC Bridges-2 Regular Memory

Bridges-2
PSC Bridges-2 Extreme Memory

Bridges-2 GPU
PSC Bridges-2 GPU

Expanse
SDSC Dell Cluster with AMD Rome HDR IB

Expanse GPU
SDSC Dell Cluster with NVIDIA V100 GPUs NVLINK and HDR IB

Jetstream
IU/TACC

KyRIC
Kentucky Research Informatics Cloud

OSG
Open Science Grid

Stampede2
TACC Dell/Intel Knights Landing, Skylake System

Storage

Jetstream Storage
IU/TACC Storage

Ocean
PSC Bridges-2 Storage

OSN
Open Storage Network

Ranch
TACC Long-term tape Archival Storage

SDSC Expanse Projects Storage

Advanced Services

SGCI
The Science Gateways Community Institute

XSEDE Extended Collaborative Support

Other

XSEDE
Public Data Collections

Innovative / coming soon: Voyager and Neocortex (specialized AI); Anvil and Delta (CPU & GPU); Ookami (ARM)

https://www.xsede.org/ecosystem/resources

XSEDE Training

Training classes

SEARCH:

START DATE	END DATE	CLASS NAME	REGISTERED
08/10/2021	08/11/2021	XSEDE HPC Workshop: Big Data and Machine Learning - August 10-11, 2021	REGISTER
07/18/2021	07/30/2021	IHPCSS 2021 Hands-on Training	REGISTER
07/12/2021	07/13/2021	Advanced Computing for Social Change (ACSC) Curriculum Development Workshop for AUCC	REGISTER
06/29/2021	06/30/2021	Workshop: Introduction to Computational Thinking Across the Curriculum: 3) Agent-based Modeling	REGISTER
06/22/2021	06/23/2021	Workshop: Introduction to Computational Thinking Across the Curriculum: 2) Systems Modeling	Registration closed

https://www.xsede.org/for-users/training

XSEDE Campus Champions

In addition to the staff at the supercomputer centers and XSEDE, there's a community of Campus Champions to provide support.

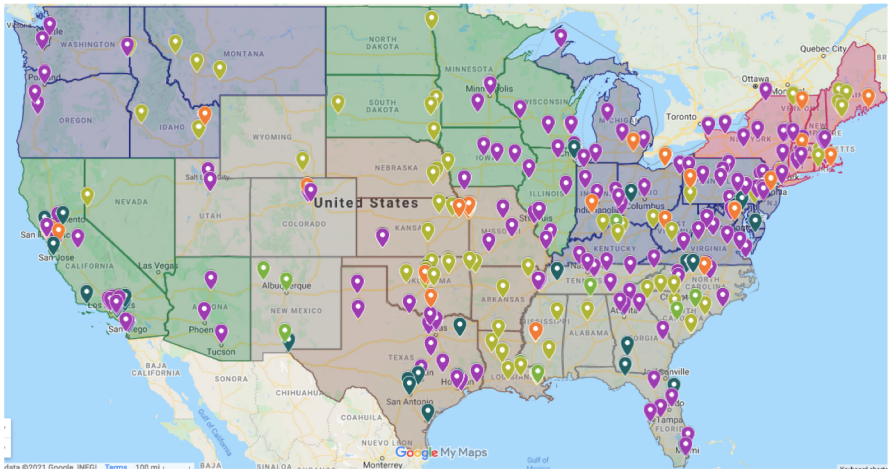
A Campus Champion helps their institution's researchers, educators and scholars with their computing-intensive and data-intensive research, education or scholarship.

https://www.xsede.org/community-engagement/campus-champions
https://www.xsede.org/web/site/community-engagement/campus-champions/current

XSEDE Campus Champions

Currently 723 Champions at 338 institutions

Institution	Campus Champions	EPSCoR	MSI
AIHEC (American Indian Higher Education Consortium)	Russell Hofmann		
Alabama A & M University	Damian Clarke, Raziq Yaqub, Georgiana Wright (student)	✔	✔
Albany State University	Olabisi Ojo		✔
Arizona State University	Michael Simeone (domain) , Sean Dudley, Johnathan Lee, Lee Reynolds, William Dizon, Ian Shaeffer, Dalena Hardy, Gil Speyer, Richard Gould, Chris Kurtz, Jason Yalim, Philip Tarrant, Douglas Jennewein, Marisa Brazil, Rebecca Belshe, Eric Tannehill, Zachary Jetson, Natalie Mason (student)		
Arkansas State University	Hai Jiang	✔	
Austin Peay State University	Justin Oelgoetz		
Bates College	Kai Evenson	✔	
Baylor College of Medicine	Pavel Sumazin , Hua-Sheng Chiu, Hyunjae Ryan Kim		
Baylor University	Mike Hutcheson, Carl Bell, Brian Sitton		
Bentley University	Jason Wells		



+ Alaska, Hawaii, Puerto Rico, USVI and Guam

https://www.xsede.org/community-engagement/campus-champions
https://www.xsede.org/web/site/community-engagement/campus-champions/current

A very brief introduction to getting hardware information

- You may be asked to report details of your hardware in a manuscript, presentation, proposal or request for computer time
- You'll know what you're running on and can answer questions like
  - Is the login node the same as the compute nodes?
  - How does one machine compare to another?
- It will give you a way of estimating performance or at least bounds on performance relative to another system. All else being equal, jobs will run at least as fast on hardware with
  - Faster CPU clock speeds
  - Larger caches
  - Faster local drives

CPU info - lscpu

There are multiple ways to get information about your CPU, but the easiest is probably the lscpu command

```
[sinkovit@login01 ~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                64
On-line CPU(s) list:   0-63
Thread(s) per core:    1
Core(s) per socket:    64
Socket(s):             1
NUMA node(s):          4
Vendor ID:             AuthenticAMD
CPU family:            23
Model:                 49
Model name:            AMD EPYC 7742 64-Core Processor
Stepping:              0
CPU MHz:               3149.843
BogoMIPS:              4491.80
Virtualization:        AMD-V
L1d cache:             32K
L1i cache:             32K
L2 cache:              512K
L3 cache:              16384K
NUMA node0 CPU(s):     0-15
NUMA node1 CPU(s):     16-31
NUMA node2 CPU(s):     32-47
NUMA node3 CPU(s):     48-63
Flags:                 fpu vme de pse tsc
```

Memory info - /proc/meminfo

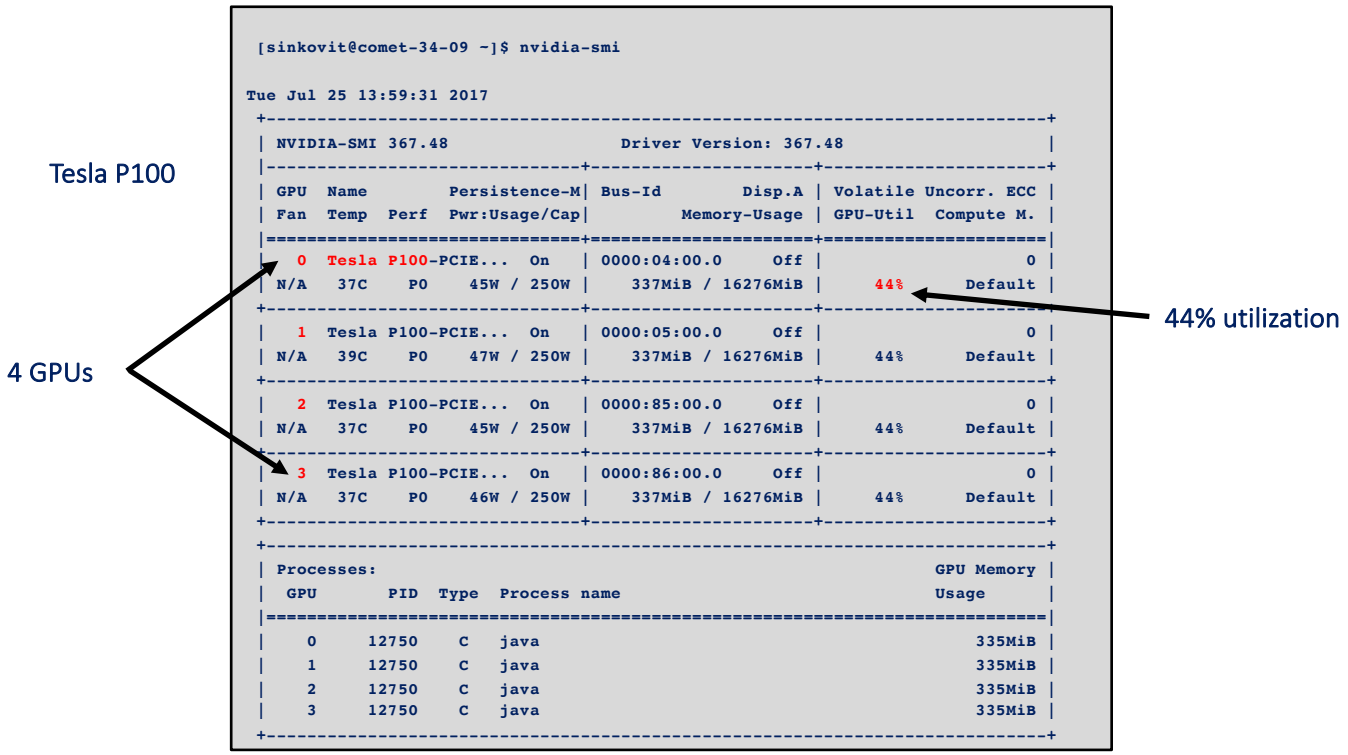
On Linux machines, the /proc/meminfo pseudo-file lists key memory specs. More information than you probably want, but at least one bit of useful data

```
[sinkovit@login01 ~]$ cat /proc/meminfo
MemTotal:       131206256 kB
MemFree:        34026548 kB
MemAvailable:   86309220 kB
Buffers:           78452 kB
Cached:         56145072 kB
SwapCached:        81344 kB
Active:         28791600 kB
Inactive:       47065492 kB
Active(anon):   21252040 kB
Inactive(anon):  5977528 kB
Active(file):    7539560 kB
...
...
```

GPU info – nvidia-smi

If you're using GPU nodes, you can use nvidia-smi (NVIDIA System Management Interface program) to get GPU information (type, count, etc.)

Note – this is from a Comet GPU node




```
[sinkovit@comet-34-09 ~]$ nvidia-smi

Tue Jul 25 13:59:31 2017
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 367.48                 Driver Version: 367.48                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  On   | 0000:04:00.0     Off |                    0 |
| N/A   37C    P0    45W / 250W |    337MiB / 16276MiB |     44%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla P100-PCIE...  On   | 0000:05:00.0     Off |                    0 |
| N/A   39C    P0    47W / 250W |    337MiB / 16276MiB |     44%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla P100-PCIE...  On   | 0000:85:00.0     Off |                    0 |
| N/A   37C    P0    45W / 250W |    337MiB / 16276MiB |     44%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla P100-PCIE...  On   | 0000:86:00.0     Off |                    0 |
| N/A   37C    P0    46W / 250W |    337MiB / 16276MiB |     44%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|    0     12750    C   java                                          335MiB |
|    1     12750    C   java                                          335MiB |
|    2     12750    C   java                                          335MiB |
|    3     12750    C   java                                          335MiB |
+-----------------------------------------------------------------------------+
```

Conclusions

- HPC systems differ from non-HPC systems primarily in scale and grade of the hardware. Nearly everything that you know carries over to HPC
- But there are some things that you need to be aware of when working on a shared resource regarding job submission, file systems, containers, etc.
- HPC resources come with a strong support network – user services, XSEDE, Campus Champions and others. You're not on your own like when running on your personal hardware

Note that we only briefly touched on many topics in this talk. Later presentations will cover the details that you need to know to run the exercises