

OpenMP Hackathon – Ascent Overview

Suzanne Parete-Koon
OLCF HPC Engineer

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

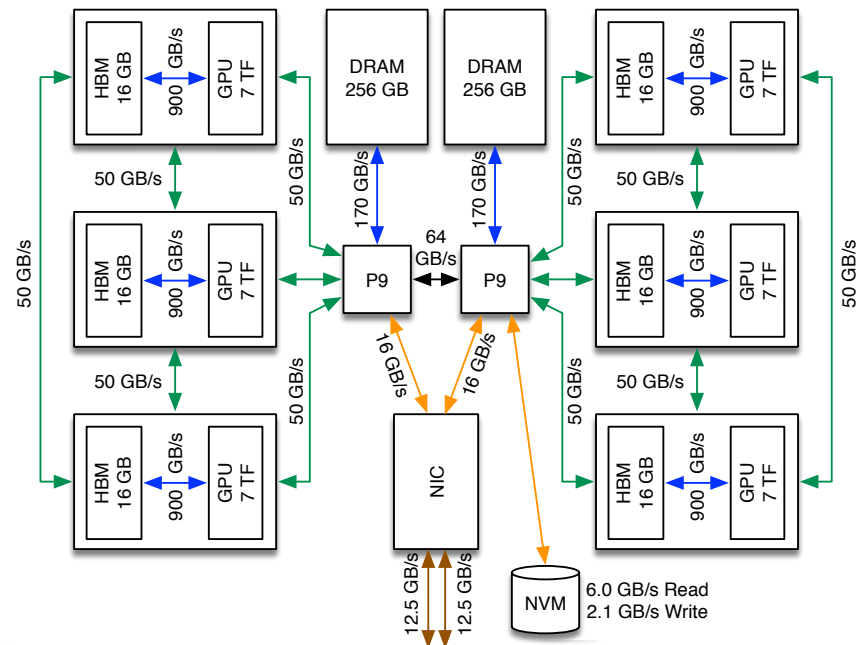
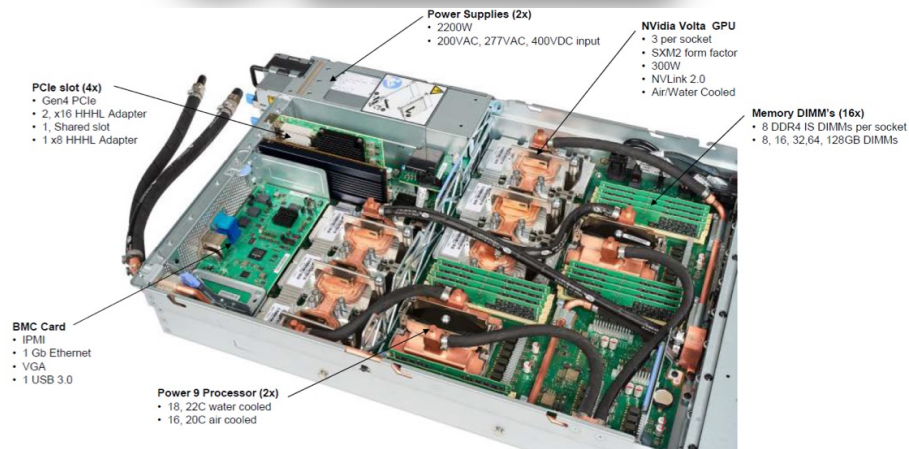
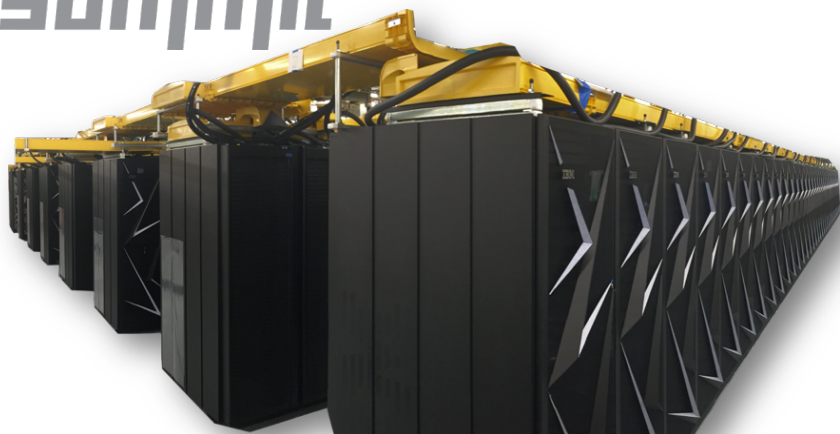
Login to Ascent

Login to Ascent with:

```
ssh UserID@login1.ascent.olcf.ornl.gov
```

Use the password that you picked when you set up your account. (not your OLCF passcode FOB)

If you have difficulty accessing Ascent, post in the *systems__ascent* slack channel.



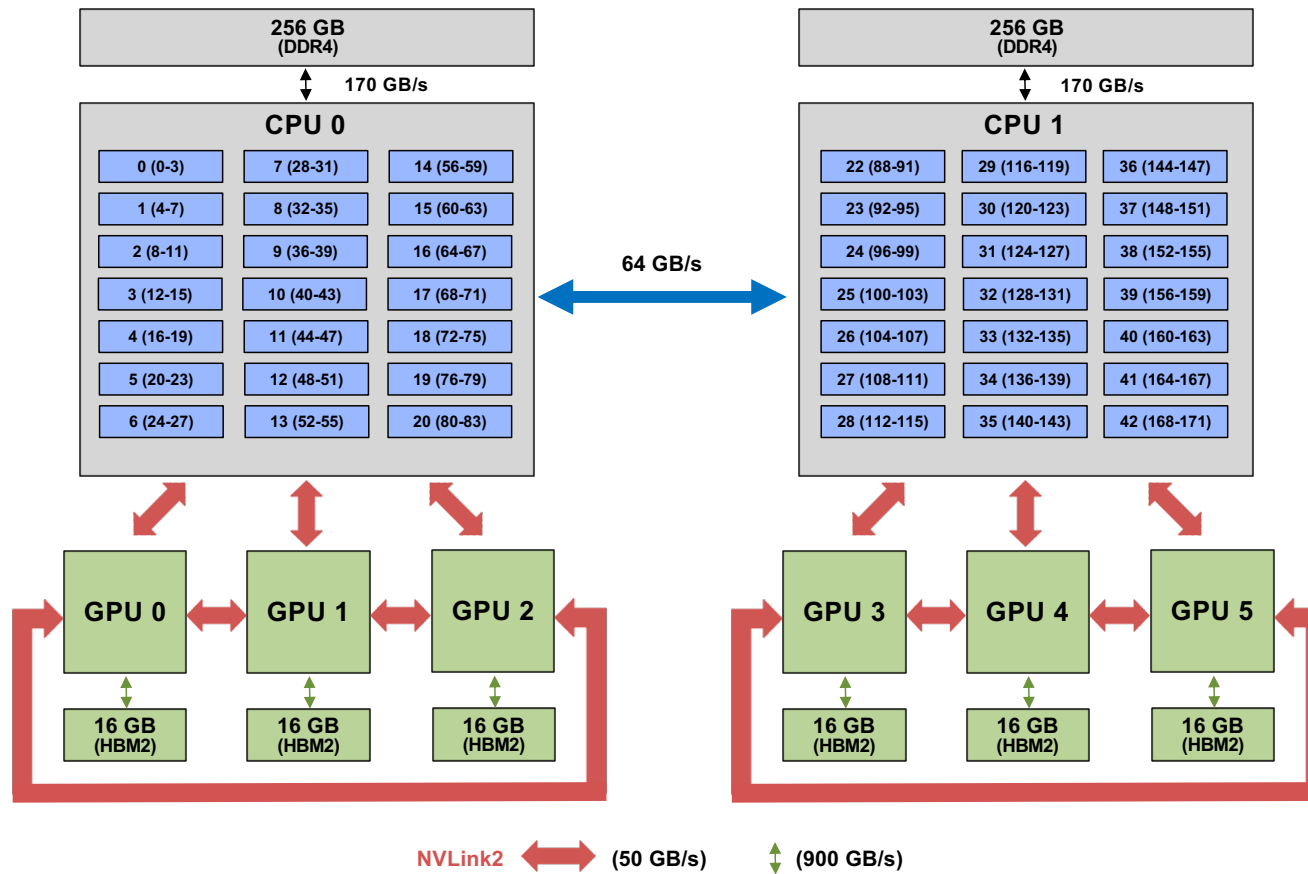
TF	42 TF (6x7 TF)	↔	HBM/DRAM Bus (aggregate B/W)
HBM	96 GB (6x16 GB)	↔	NVLink
DRAM	512 GB (2x16x16 GB)	↔	X-Bus (SMP)
NET	25 GB/s (2x12.5 GB/s)	↔	PCIe Gen4
MMsg/s	83	↔	EDR IB

HBM & DRAM speeds are aggregate (Read+Write).
All other speeds (X-Bus, NVLink, PCIe, IB) are bi-directional.

Node Overview

Summit Node

(2) IBM Power9 + (6) NVIDIA Volta V100



Available File Systems / Storage Areas on Ascent

NFS Directories – This is where you might want to keep source code and build your application.

NOTE: These directories are read-only from the compute nodes!

[/ccsopen/home/userid](#)

- Your personal home directory

[/ccsopen/proj/gen163](#)

- Can be accessed by all participants of this event
 - You should create a directory here with your team name to collaborate (source code, scripts, etc.)
-

GPFS Directories (parallel file system) – This is where you should write data when running on Ascent's compute nodes.

[/gpfs/wolf/gen163/scratch/userid](#)

- Your personal GPFS scratch directory

[/gpfs/wolf/gen163/proj-shared](#)

- Can be accessed by all participants of the event
 - You should create a directory here with your team name to collaborate (data written from compute nodes)
-

OpenMP Offloading : Summary Table

Compiler			Module	Offloading Flags	Useful Flags	Useful Environment variables (verbose)
C	C++	Fortran				
xlc	C++	xf	xl/16.1.1-10	-qomp=omp -qoffload		
nvc	nvc++	nvfortran	nvhpc/21.7	-mp=gpu -gpu=cc70	-Minfo=accel -Minfo=mp -Minfo=loop	NVCOMPILER_ACC_NOTIFY
clang	clang++	flang	llvm/12.0.0	-fopenmp \ -fopenmptargets=nvptx64- nvidia-cuda \ -Xopenmp-target \ -march=sm_70		LIBOMPTARGET_INFO=-1
gcc	g++	gfortran	gcc/10.2.0	-fopenmp	-foffload= "-lm -latomic"	GOMP_DEBUG=1

- **The cuda module needs to be loaded for the LLVM clang compiler to target GPU offloading**
- Spectrum MPI provides compiler wrappers that automatically choose the proper compiler.
- To see the available versions: \$ module -t avail XL
- https://docs.olcf.ornl.gov/systems/summit_user_guide.html#compiling

jsrun – Basic Options

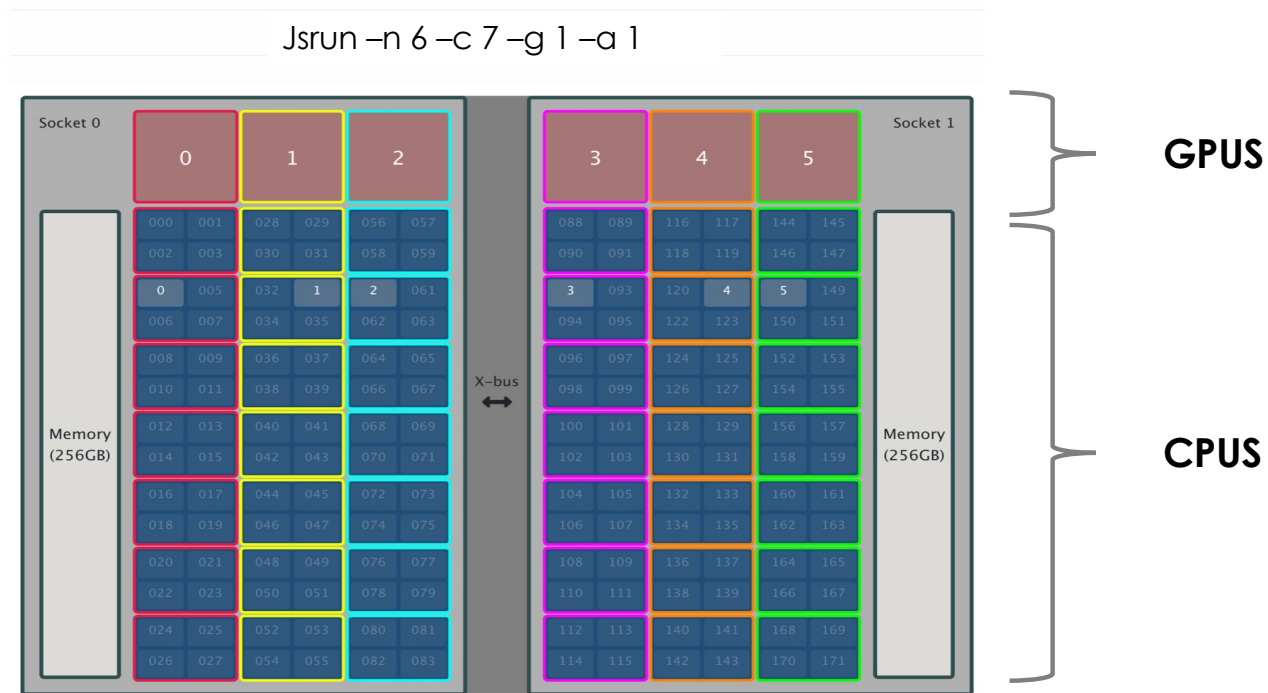
`jsrun [-n #resource sets] [CPU cores, GPUs, tasks in each resource set] program [program args]`

jsrun Flags		Description	Default Value
Long	Short		
<code>--nrs</code>	<code>-n</code>	Number of RS	All available physical cores
<code>--tasks_per_rs</code>	<code>-a</code>	Number of MPI tasks (ranks) per RS	N/A (total set instead [-p])
<code>--cpu_per_rs</code>	<code>-c</code>	Number of CPUs (physical cores) per RS	1
<code>--gpu_per_rs</code>	<code>-g</code>	Number of GPUs per RS	0
<code>--bind</code>	<code>-b</code>	Number of physical cores allocated per task	<code>packed:1</code>
<code>--rs_per_host</code>	<code>-r</code>	Number of RS per host (node)	N/A
<code>--latency_priority</code>	<code>-l</code>	Controls layout priorities	<code>gpu-cpu,cpu-mem,cpu-cpu</code>
<code>--launch_distribution</code>	<code>-d</code>	Order of tasks started on multiple RS	<code>packed</code>

See `man jsrun` for full list of options

jsrun – Basic Options

`jsrun [-n #resource sets] [CPU cores, GPUs, tasks in each resource set] program [program args]`



For more description see:

https://docs.olcf.ornl.gov/systems/summit_user_guide.html#resource-sets

Basic Batch Script

```
#!/bin/bash
# Begin LSF Directives
#BSUB -P GEN163
#BSUB -W 1:00
#BSUB -nnodes 1
#BSUB -alloc_flags gpumps
#BSUB -J RunSim123
#BSUB -o RunSim123.%J
#BSUB -e RunSim123.%J
```

```
date
jsrun -n 6 -c 7 -g 1 -a 1./a.out
```

Don't copy this text to start a batch script, use the ones in the link below.

To see more examples and BSUB Options:

https://docs.olcf.ornl.gov/systems/summit_user_guide.html#batch-scripts

jsrun Job Launcher – Tools & Documentation

hello_jsrun

- https://code.ornl.gov/t4p/Hello_jsrun
- Simple “Hello World”-type program used to test layout of resources on a Summit node using `jsrun`.

job-step-viewer

- <https://jobstepviewer.olcf.ornl.gov/>

“Job Launcher (jsrun)” section of the Summit User Guide

- https://docs.olcf.ornl.gov/systems/summit_user_guide.html#job-launcher-jsrun

Additional training materials to learn about `jsrun`

A (fairly) quick tutorial on using the `jsrun` job launcher on Summit/Ascent:

- https://github.com/olcf-tutorials/jsrun_quick_start_guide

Recent Presentation of “jsrun Basics”

- Slides: https://www.olcf.ornl.gov/wp-content/uploads/2019/12/jsrun_basics.pdf
- Recording: <https://vimeo.com/393782415>

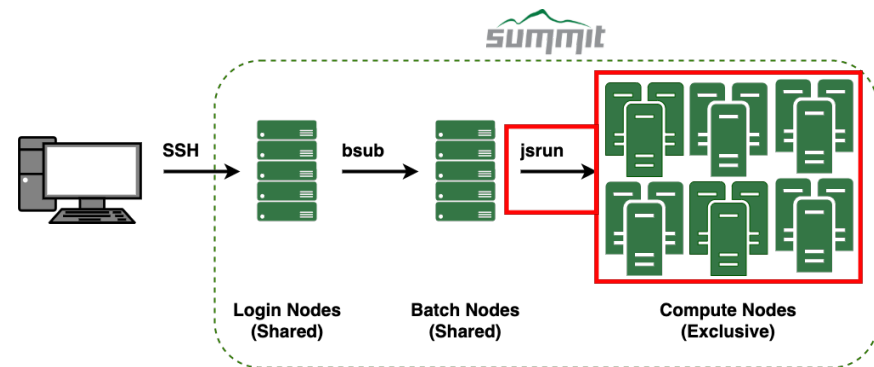
Must use `jsrun` to run on compute nodes

```
[UID@login1: ~]$ hostname  
login1
```

```
[UID@login1: ~]$ bsub -P GEN163 -nnodes 1 -W 60 -Is /bin/bash  
Job <15167> is submitted to default queue <batch>.  
<<Waiting for dispatch ...>>  
<<Starting on login1>>
```

```
[UID@login1: ~]$ hostname  
login1
```

```
[UID@login1: ~]$ jsrun -n1 hostname  
h49n16
```



The login nodes are shared among all participants (compiling, file editing, data analysis, etc.), so please **DO NOT RUN YOUR APPLICATIONS ON THE LOGIN NODES!!**

Some Useful Commands / Flags

jobstat

- Shows information about the jobs running on the system

--smpiargs="-gpu"

- `jsrun` flag that enables CUDA-Aware MPI
- If you are not familiar with CUDA-Aware MPI or GPUDirect, please see this tutorial: https://github.com/olcf-tutorials/MPI_ping_pong

-alloc_flag "gpumps smt1"

- `bsub` flag that allows you to start a CUDA MPS server or change the SMT mode of the physical CPU cores
- Multiple options are separated by a space-delimited list
- [https://docs.olcf.ornl.gov/systems/summit_user_guide.html?highlight=alloc_flag#com](https://docs.olcf.ornl.gov/systems/summit_user_guide.html?highlight=alloc_flag#common-bsub-options)

[mon-bsub-options](https://docs.olcf.ornl.gov/systems/summit_user_guide.html?highlight=alloc_flag#common-bsub-options)

Ascent Queue Policy

Number of Nodes	Max Walltime
1 – 2	2 hours
3 – 4	1 hour

There are a total of 16 schedulable compute nodes in Ascent, so please be respectful of others when requesting resources...

- Try to limit yourself to 1 compute node unless needed
- When you're finished with an allocation, please kill it (i.e., `exit` from within an interactive job or `bkill JOBID` for batch jobs).

Other Helpful Links

OLCF Summit User Guide

- https://docs.olcf.ornl.gov/systems/summit_user_guide.html
- NOTE: Ascent mounts different file systems than Summit, so please refer to info in these slides or the Training System (Ascent) section of the Summit User Guide for this information
 - https://docs.olcf.ornl.gov/systems/summit_user_guide.html#training-system-ascent
- NVIDIA's Nsight Profiling Tools
 - https://docs.olcf.ornl.gov/systems/summit_user_guide.html#profiling-gpu-code-with-nvidia-developer-tools

OLCF Training Archive

- Contains slides and recordings from previous OLCF training events.
- https://docs.olcf.ornl.gov/training/training_archive.html



Questions?