LA-UR-18-25080

Approved for public release; distribution is unlimited.

Title: The Darwin Cluster

Author(s): Garrett, Charles Kristopher

Intended for: Lecture for the Parallel Computing Summer Research Internship

Issued: 2018-06-11

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

The Darwin Cluster

Kris Garrett

June 2018

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

What is Darwin?

What Darwin Is

From our website: **darwin.lanl.gov**

Darwin is an **ASC funded test bed cluster** that also allows external users to use the idle resources within. Darwin is configured as a non-standard HPC production stack, and as such, has some interesting quirks to it.

Darwin is a **very heterogeneous cluster**. Where average clusters are designed as homogeneous to make management and expectations simple, Darwin was designed as a test bed with **many types of nodes** available for running. With this heterogeneity comes its own run time experience. We provide nodes that contain x86 architectures of many flavors and also Power PC and ARM architectures. We have nodes with terabytes of memory and nodes with all kinds of GPUs.

What Darwin Is Not

- **Darwin is not a production space**
  - If you are looking for a place to run your large, production ready job, you have come to the wrong place
- **Darwin is not a storage space**
  - No backups of user data are done on Darwin
  - You have been warned!
- **Darwin is not your computer**
  - Darwin has many users besides you
  - Be courteous!

Darwin the Frankencluster

- **Intel CPUs**
  - Sandy Bridge
  - Ivy Bridge
  - Haswell
  - Broadwell
  - Sky Lake
  - KNL
- **IBM CPUs**
  - Power 8+
  - Power 9
- **ARM Cavium CPUs**
- **AMD EPYC CPUs**
- **NVIDIA GPUs**
  - Quadro K6000
  - GTX Titan X
  - Titan V
  - Tesla K20
  - Tesla K40
  - Tesla K80
  - Pascal P100
  - Volta V100

Darwin the Frankencluster

- **Network**
  - Ethernet everywhere
  - Pockets of infiniband

- **Other Node Types**
  - Up to 3 TB memory
  - NVMe
  - SATA SSDs

- **New Compilers**
  - OpenMPI 3.1.0
  - GCC 8.1.0
  - Intel 18.0.2
  - PGI 18.4
  - CUDA 9.1/9.2
  - XL 16.1.0

- **Slurm Workload Manager**

How to use Darwin

Logging in to Darwin

- **Must be on the Yellow or OCE network**
  - If outside LANL use VPN or ssl-portal

- **ssh darwin-fe**
  - Puts you on a front end node: darwin-fe[1,2]
  - Can login to darwin-fe3 for a high memory front end node

- **For X forwarding**
  - On a Mac: ssh –Y darwin-fe
  - Otherwise: ssh –X darwin-fe

Finding a Node

- **Darwin is split into partitions**
  - Most common partitions
    - general (default)
    - power8+
    - power9
    - arm
    - scaling

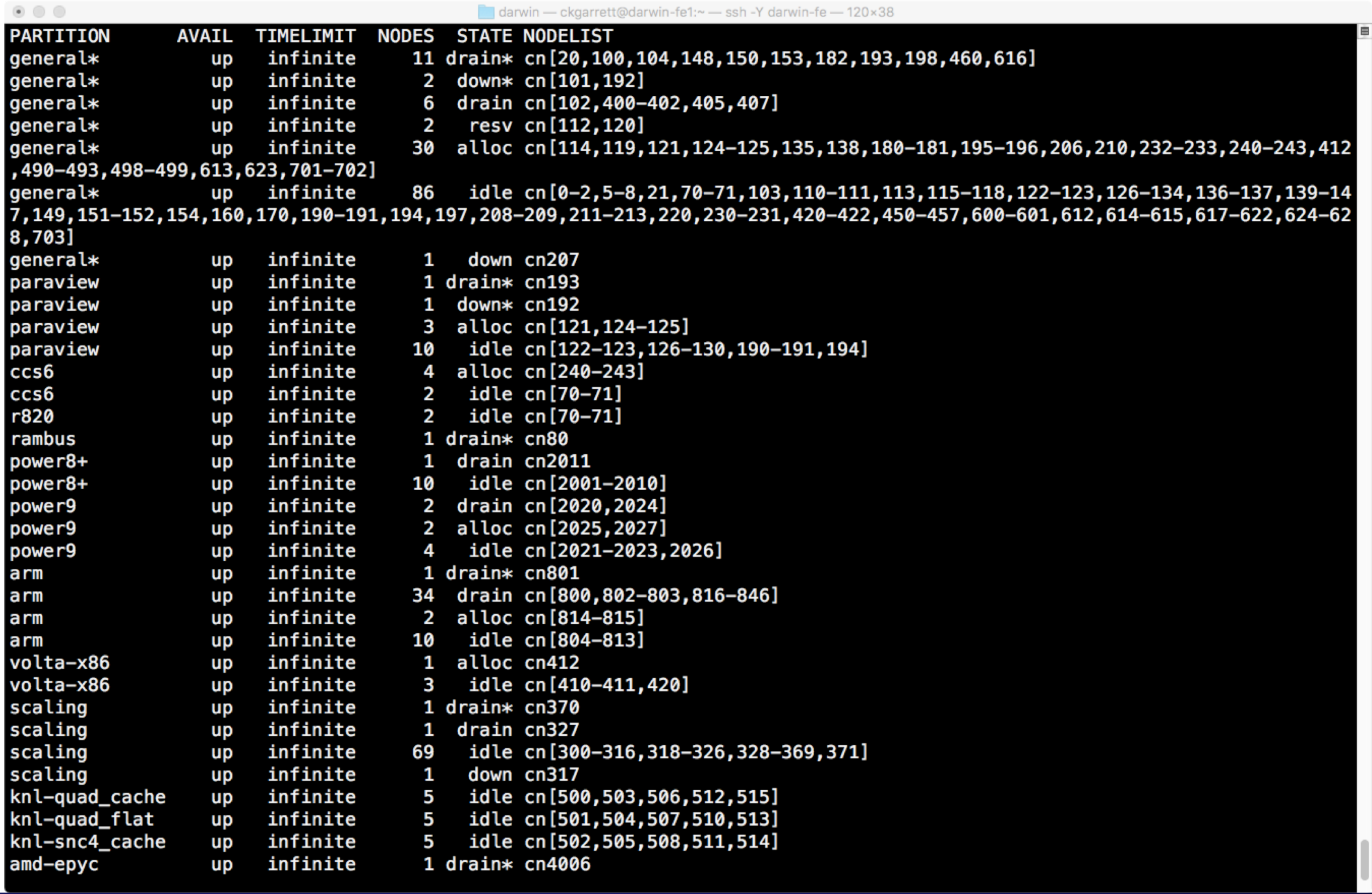
- **Use constraints for selection within the *general* partition**

Finding a Node

sinfo

Useful for finding partition names

Quick snapshot of up/down nodes



```
PARTITION      AVAIL  TIMELIMIT  NODES  STATE NODELIST
general*          up   infinite     11 drain* cn[20,100,104,148,150,153,182,193,198,460,616]
general*          up   infinite      2  down* cn[101,192]
general*          up   infinite      6  drain cn[102,400-402,405,407]
general*          up   infinite      2   resv cn[112,120]
general*          up   infinite     30  alloc cn[114,119,121,124-125,135,138,180-181,195-196,206,210,232-233,240-243,412
,490-493,498-499,613,623,701-702]
general*          up   infinite     86   idle cn[0-2,5-8,21,70-71,103,110-111,113,115-118,122-123,126-134,136-137,139-14
7,149,151-152,154,160,170,190-191,194,197,208-209,211-213,220,230-231,420-422,450-457,600-601,612,614-615,617-622,624-62
8,703]
general*          up   infinite      1   down cn207
paraview          up   infinite      1 drain* cn193
paraview          up   infinite      1  down* cn192
paraview          up   infinite      3  alloc cn[121,124-125]
paraview          up   infinite     10   idle cn[122-123,126-130,190-191,194]
ccs6              up   infinite      4  alloc cn[240-243]
ccs6              up   infinite      2   idle cn[70-71]
r820              up   infinite      2   idle cn[70-71]
rambus            up   infinite      1 drain* cn80
power8+           up   infinite      1  drain cn2011
power8+           up   infinite     10   idle cn[2001-2010]
power9            up   infinite      2  drain cn[2020,2024]
power9            up   infinite      2  alloc cn[2025,2027]
power9            up   infinite      4   idle cn[2021-2023,2026]
arm               up   infinite      1 drain* cn801
arm               up   infinite     34  drain cn[800,802-803,816-846]
arm               up   infinite      2  alloc cn[814-815]
arm               up   infinite     10   idle cn[804-813]
volta-x86         up   infinite      1  alloc cn412
volta-x86         up   infinite      3   idle cn[410-411,420]
scaling           up   infinite      1 drain* cn370
scaling           up   infinite      1  drain cn327
scaling           up   infinite     69   idle cn[300-316,318-326,328-369,371]
scaling           up   infinite      1   down cn317
knl-quad_cache    up   infinite      5   idle cn[500,503,506,512,515]
knl-quad_flat     up   infinite      5   idle cn[501,504,507,510,513]
knl-snc4_cache    up   infinite      5   idle cn[502,505,508,511,514]
amd-epyc          up   infinite      1 drain* cn4006
```

Finding a Node

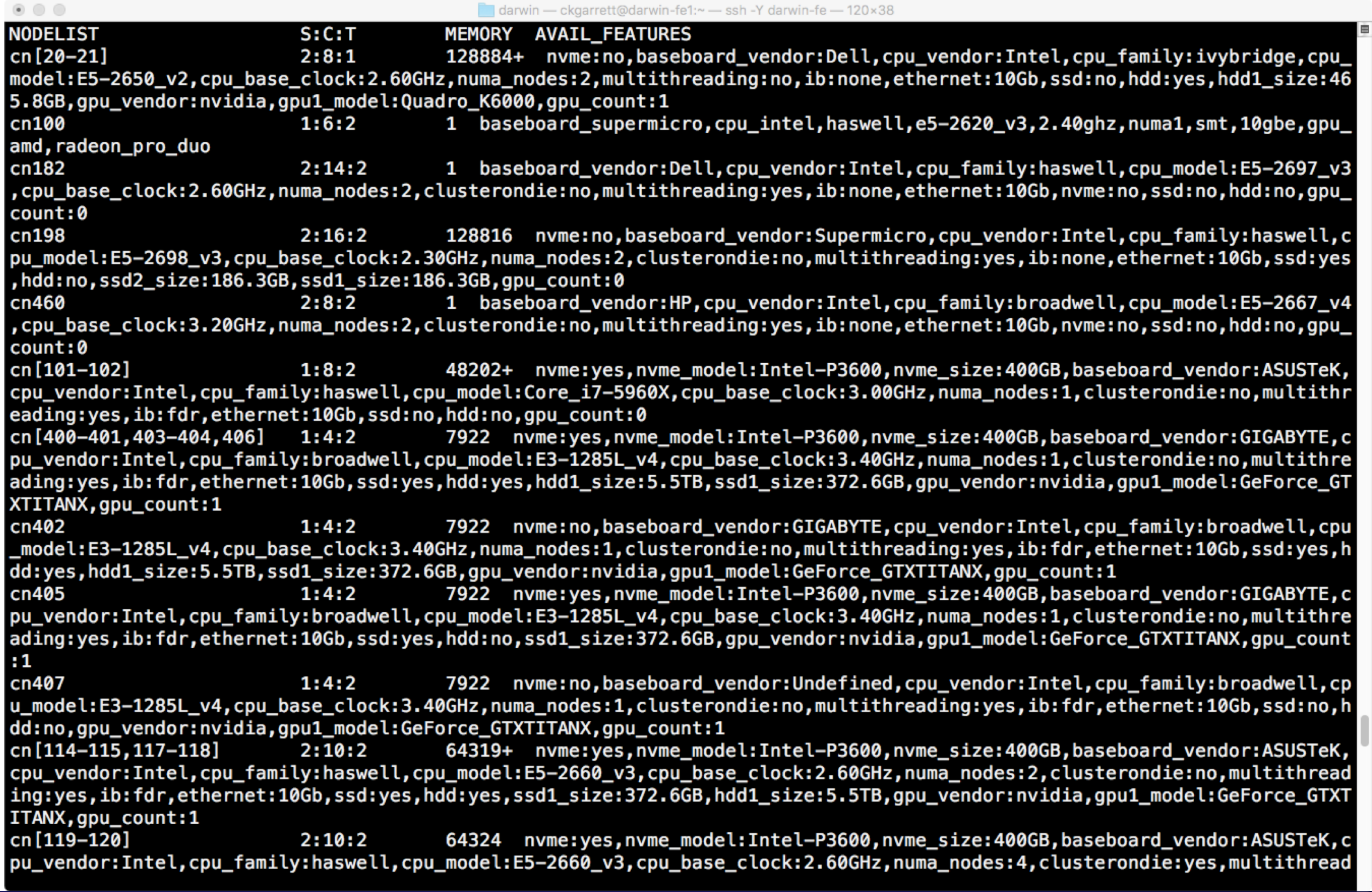
sinfo_s

Find nodes with specific hardware

Use with grep to pare down information

`sinfo_s | grep nvidia`

sinfo_f
similar but shows up/down/alloc



```
NODELIST                 S:C:T        MEMORY  AVAIL_FEATURES
cn[20-21]                2:8:1        128884+  nvme:no,baseboard_vendor:Dell,cpu_vendor:Intel,cpu_family:ivybridge,cpu_
model:E5-2650_v2,cpu_base_clock:2.60GHz,numa_nodes:2,multithreading:no,ib:none,ethernet:10Gb,ssd:no,hdd:yes,hdd1_size:46
5.8GB,gpu_vendor:nvidia,gpu1_model:Quadro_K6000,gpu_count:1
cn100                    1:6:2        1  baseboard_supermicro,cpu_intel,haswell,e5-2620_v3,2.40ghz,numa1,smt,10gbe,gpu_
amd,radeon_pro_duo
cn182                    2:14:2       1  baseboard_vendor:Dell,cpu_vendor:Intel,cpu_family:haswell,cpu_model:E5-2697_v3
,cpu_base_clock:2.60GHz,numa_nodes:2,clusterondie:no,multithreading:yes,ib:none,ethernet:10Gb,nvme:no,ssd:no,hdd:no,gpu_
count:0
cn198                    2:16:2       128816  nvme:no,baseboard_vendor:Supermicro,cpu_vendor:Intel,cpu_family:haswell,c
pu_model:E5-2698_v3,cpu_base_clock:2.30GHz,numa_nodes:2,clusterondie:no,multithreading:yes,ib:none,ethernet:10Gb,ssd:yes
,hdd:no,ssd2_size:186.3GB,ssd1_size:186.3GB,gpu_count:0
cn460                    2:8:2        1  baseboard_vendor:HP,cpu_vendor:Intel,cpu_family:broadwell,cpu_model:E5-2667_v4
,cpu_base_clock:3.20GHz,numa_nodes:2,clusterondie:no,multithreading:yes,ib:none,ethernet:10Gb,nvme:no,ssd:no,hdd:no,gpu_
count:0
cn[101-102]              1:8:2        48202+  nvme:yes,nvme_model:Intel-P3600,nvme_size:400GB,baseboard_vendor:ASUSTeK,
cpu_vendor:Intel,cpu_family:haswell,cpu_model:Core_i7-5960X,cpu_base_clock:3.00GHz,numa_nodes:1,clusterondie:no,multithr
eading:yes,ib:fdr,ethernet:10Gb,ssd:no,hdd:no,gpu_count:0
cn[400-401,403-404,406]  1:4:2        7922  nvme:yes,nvme_model:Intel-P3600,nvme_size:400GB,baseboard_vendor:GIGABYTE,c
pu_vendor:Intel,cpu_family:broadwell,cpu_model:E3-1285L_v4,cpu_base_clock:3.40GHz,numa_nodes:1,clusterondie:no,multithre
ading:yes,ib:fdr,ethernet:10Gb,ssd:yes,hdd:yes,hdd1_size:5.5TB,ssd1_size:372.6GB,gpu_vendor:nvidia,gpu1_model:GeForce_GT
XTITANX,gpu_count:1
cn402                    1:4:2        7922  nvme:no,baseboard_vendor:GIGABYTE,cpu_vendor:Intel,cpu_family:broadwell,cpu
_model:E3-1285L_v4,cpu_base_clock:3.40GHz,numa_nodes:1,clusterondie:no,multithreading:yes,ib:fdr,ethernet:10Gb,ssd:yes,h
dd:yes,hdd1_size:5.5TB,ssd1_size:372.6GB,gpu_vendor:nvidia,gpu1_model:GeForce_GTXTITANX,gpu_count:1
cn405                    1:4:2        7922  nvme:yes,nvme_model:Intel-P3600,nvme_size:400GB,baseboard_vendor:GIGABYTE,c
pu_vendor:Intel,cpu_family:broadwell,cpu_model:E3-1285L_v4,cpu_base_clock:3.40GHz,numa_nodes:1,clusterondie:no,multithre
ading:yes,ib:fdr,ethernet:10Gb,ssd:yes,hdd:no,ssd1_size:372.6GB,gpu_vendor:nvidia,gpu1_model:GeForce_GTXTITANX,gpu_count
:1
cn407                    1:4:2        7922  nvme:no,baseboard_vendor:Undefined,cpu_vendor:Intel,cpu_family:broadwell,cp
u_model:E3-1285L_v4,cpu_base_clock:3.40GHz,numa_nodes:1,clusterondie:no,multithreading:yes,ib:fdr,ethernet:10Gb,ssd:no,h
dd:no,gpu_vendor:nvidia,gpu1_model:GeForce_GTXTITANX,gpu_count:1
cn[114-115,117-118]      2:10:2       64319+  nvme:yes,nvme_model:Intel-P3600,nvme_size:400GB,baseboard_vendor:ASUSTeK,
cpu_vendor:Intel,cpu_family:haswell,cpu_model:E5-2660_v3,cpu_base_clock:2.60GHz,numa_nodes:2,clusterondie:no,multithread
ing:yes,ib:fdr,ethernet:10Gb,ssd:yes,hdd:yes,ssd1_size:372.6GB,hdd1_size:5.5TB,gpu_vendor:nvidia,gpu1_model:GeForce_GTXT
ITANX,gpu_count:1
cn[119-120]              2:10:2       64324  nvme:yes,nvme_model:Intel-P3600,nvme_size:400GB,baseboard_vendor:ASUSTeK,c
pu_vendor:Intel,cpu_family:haswell,cpu_model:E5-2660_v3,cpu_base_clock:2.60GHz,numa_nodes:4,clusterondie:yes,multithread
```

Getting a Node

- **salloc -N 1**
  - Gets an x86 node in the general partition
- **salloc -N 1 --constraint=“cpu_family:haswell”**
  - Gets a node in the general partition with a XEON Haswell processor
  - Find the constraint name with sinfo_s or sinfo_f
  - cn[190-194]   2:16:2   128822+ nvme:yes,nvme_model:Intel-P3600,nvme_size:400GB,baseboard_vendor:Supermicro,cpu_vendor:Intel,**cpu_family:haswell**,cpu_model:E5 2698_v3,cpu_base_clock:2.30GHz,numa_nodes:4,clusterondie:yes,multithreading:yes,ib:none,ethernet:10Gb,ssd:no,hdd:no,gpu_count:0
  - Remember to grep for what you’re looking for
- **salloc -w cn212**
  - Allocate a specific node

Getting a Node

- **To get a node in another partition**
  - salloc –p power9
  - Most partitions have one node type: but check!

- **X forwarding from a compute node**
  - salloc -N1 -C cpu_family:haswell --x11

Modules

- **To use software on Darwin, use modules**
  - module avail: see available modules
  - module list: see loaded modules
  - module load <module>: load a module
  - module unload <module>: unload a module
  - module purge: unload all loaded modules
  - Different partitions have different modules!

- **Regularly updated modules**
  - GCC, Intel, PGI, XL, CUDA, OpenMPI

- **Logging in to a compute node that is not x86 removes all modules!**

Front End vs Compute Node

- **What to do on front end nodes**
  - Edit files
  - Submit batch jobs
  - Run sinfo, sinfo_s, sinfo_f
  - Takeaway: don’t do much on a front end node!

- **What to do on compute nodes**
  - Run Matlab, parallel programs, or other intensive programs
  - Compile
    - Front end nodes do not have the same architecture as compute nodes
    - Using –march=native on a front end node will do the wrong thing

File Systems

- **No quota on disk space**
- **Not backed up!!!**
- **Two spaces**
  - /home/<moniker>
  - /scratch/users/<moniker>
- **/scratch**
  - Added recently because we had extra hardware
  - Less used currently so may perform a little better

Quotas

- **Time quotas (default time is 2 hours on all QOS)**

QOS	Normal	Long	Debug	Scavenge
Max wall clock	10 Hours	2 Days	4 Hours	7 Days
Max Node Count	32	16	32	None
Priority	Normal	Normal	High	Low

- **Slurm commands: -t <time>, --qos=<name>**

MPI

- **Load an openmpi module**
  - module load openmpi/3.1.0-gcc_7.3.0
  - This will load the associated compiler too

- **Use mpirun, not srun**
  - One node run: mpirun –n <num_processes> <application>
  - May want the extra flags: --mca btl ^openib
    - MPI compiled with infiniband support but many nodes don’t have infiniband
    - Causes OpenMPI to spew warnings
    - These flags stop the warnings

MPI

- **Use mpirun on multiple nodes**
  - mpirun -N 1 --hostfile /path/to/hostfile /path/to/mpi_hello_world

    ```
    Hello world from processor cn314, rank 0 out of 4 processors
    Hello world from processor cn320, rank 2 out of 4 processors
    Hello world from processor cn321, rank 3 out of 4 processors
    Hello world from processor cn332, rank 1 out of 4 processors
    ```
  - hostfile is a file with names of nodes: one name per line

    ```
    cn314
    cn320
    cn321
    cn332
    ```
  - Can get all node names from the environment variable: SLURM_NODELIST
  - Make sure all nodes are identical:
    - The *scaling* partition was made for this

Python

- **Use Anaconda**
  - module load anaconda/Anaconda3

- **For extra modules, create a local conda environment**
  - conda create -n MyCondaEnvName pip
  - source activate MyCondaEnvName

- **Install modules with pip**
  - pip install <PythonPackageName>
  - Example: pip install tensorflow

Administration

- **To get an account on Darwin**
  - Email darwin-admin@lanl.gov
  - Give us your name, z number, moniker, and a reason for using the cluster

- **For general questions/requests**
  - First, see if darwin.lanl.gov has the answer
  - Then, email darwin-admin@lanl.gov

- **There is also a user list**
  - darwin-users@lanl.gov
  - Don’t spam everyone!

The End