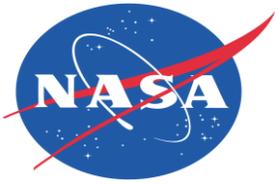




# Migrating Your Code to SLES 11 SP3 on Discover: Lessons Learned at the NCCS

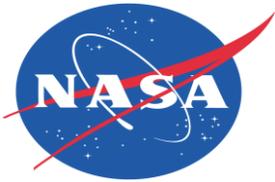
The Discover team  
NASA Center for Climate Simulation  
Monday 31 August 2015



# Why did we upgrade to SP3?



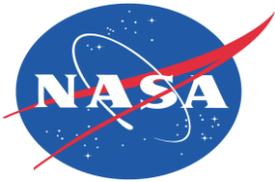
- Vendor support for SP1 is ending!
- New Haswell nodes require SP3.
- Drivers for new InfiniBand Host Channel Adaptor (IB HCA) cards require SP3.
- Upgraded drivers for Intel Phi coprocessor cards require updated OpenFabrics Enterprise Distribution (OFED) and SP3.
- Drivers and Compute Unified Device Architecture (CUDA) 6.x for new nVidia K40 GPUs require SP3.



# Overview: Compute nodes



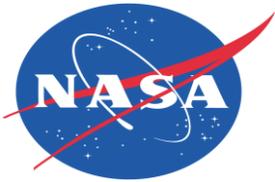
- **SCU8** (borg01 [wx] NNN, SP1, InfiniBand Fabric 2)
  - 480 Sandy Bridge nodes x 16 cores/node = 7,680 cores, 32 GB/node (2 GB/core)
- **SCU9** (borg01 yNNN, SP1, InfiniBand Fabric 2)
  - 240 Sandy Bridge nodes x 16 cores/node = 3,840 cores, 64 GB/node (4 GB/core)
- **SCU9** (borg01 zNNN, SP3, InfiniBand Fabric 3)
  - 240 Sandy Bridge nodes x 16 cores/node = 3,840 cores, 64 GB/node (4 GB/core)
- **SCU10** (borg [jklmn] NNN, SP3, InfiniBand Fabric 3)
  - 1,080 Haswell nodes x 28 cores/node = 30,240 cores, 128 GB/node (4.5 GB/core)
- **SCU11** (borg [opq] NNN, SP3, InfiniBand Fabric 1)
  - 612 Haswell nodes x 28 cores/node = 17,136 cores, 128 GB/node (4.5 GB/core)
- **SCU12** (borg [rst] NNN, SP3, InfiniBand Fabric 1)
  - 612 Haswell nodes x 28 cores/node = 17,136 cores, 128 GB/node (4.5 GB/core)
- **Total:** 3,264 nodes with 79,872 cores (not counting accelerators)



# Overview: Interactive nodes



- **SP1 login nodes** (`discover|discover-sp1`)
  - `discover[25-28]` (IB Fabric 2)
  - 4 Sandy Bridge nodes x 16 cores/node = 64 cores, 64 GB/node (4 GB/core)
- **SP3 login nodes** (`discover-sp3`)
  - `discover[05-08]` (IB Fabric 3)
  - `discover[15-18]` (IB Fabric 1)
  - 8 Haswell nodes x 28 cores/node = 224 cores, 128 GB/node (4.5 GB/core)
- **Large-memory interactive nodes**
  - `dali[09-17]` (SP1, IB Fabric 2)
  - 9 Westmere nodes x 12 cores/node = 108 cores, 192 GB/node (16 GB/core)



# Overview: Accelerators



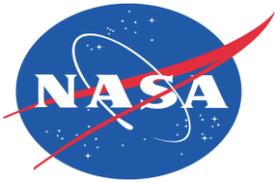
- nVidia GPUs
  - `--constraint=k10:borg01w040`
  - `--constraint=k20:borg01w[038-039]`
  - `--constraint=k40:borg01z[205-240]`
- Intel Phi cards
  - Offload mode (SCU8): `borg01w[001-239], borg01x[001-239]`
  - Native mode (native partition): `borg01w[001-018]`
- See the NCCS Primer for accelerator information:  
<http://www.nccs.nasa.gov/primer/>



# Overview: InfiniBand fabrics



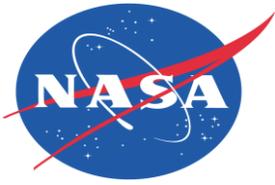
- Fabric 1 (SP3)
  - FDR (Fourteen Data Rate = 54 Gbps)
  - SCU11 + SCU12
- Fabric 2 (SP1)
  - QDR (Quad Data Rate = 32 Gbps)
  - SCU8 + SCU9 (240 nodes)
- Fabric 3 (SP3)
  - FDR (Fourteen Data Rate = 54 Gbps)
  - SCU9 (240 nodes) + SCU10
- Jobs are allocated across **only 1** fabric.



# Overview: Partitions (most common)



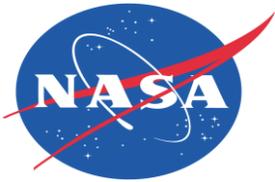
- Used to group resources for management
- Equivalent to PBS queues
- `compute` – Default; nearly all compute work is done here
- `datamove` – For data move jobs on Dirac
- `native` – For native-mode Phi use (`borg01w[001-018]`)
- Retired partitions: `debug`, `sp3`
  - Remove these from your scripts!



# Overview: QoS – Quality of Service



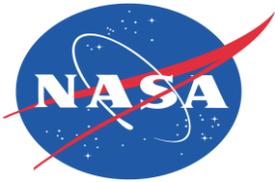
- Used to compute job priority and scheduling
- `allnccs` – The default QoS
- `debug` – Short (1 hour maximum) debugging runs
- `long` – 24 hour time limit
- `serial` – Only one job at a time will run



# SP1 vs SP3



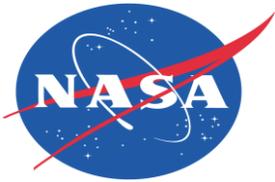
- Linux kernel upgraded - 2.6 vs 3.0
  - Still not the most recent, but vendor-supported
  - CMA (Cross-Memory Attach) in SP3 kernel
- Swap space: 8 GB vs NONE AT ALL!
- Many system libraries upgraded, e.g.
  - `/lib/libc.so`: 2.11.1 vs 2.11.3
  - `/usr/lib/libstdc++.so`: 6.0.10 vs 6.0.17
- Many system tools upgraded, e.g.
  - python: 2.6 vs 2.6.9
  - git: 1.6.0.2 vs 1.7.12.4



# Sandy Bridge vs Haswell



- Cores: 16 vs 28
- Memory: 32 or 64 GB vs 128 GB
- (SP1 or SP3) vs ONLY SP3
- Instruction sets: AVX vs AVX2 (wider vector registers)
- Sandy Bridge nodes are IBM nodes
- Haswell nodes are SGI nodes
  - MPT should only be used on SGI nodes.



# Choose your weapons carefully...



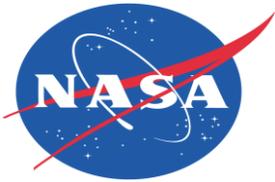
- **All** Haswell nodes run SP3.
- **For now**, some Sandy Bridge nodes are SP1, some SP3.
- **All** Sandy Bridge nodes **will** eventually be SP3.
- For now, specify SP1/SP3 and node type independently.
- To specify node OS:
  - `--constraint=sp1` - Only gets Sandy Bridge
  - `--constraint=sp3` - Gets Sandy Bridge **or** Haswell
- To specify node hardware:
  - `--constraint=sand`
  - `--constraint=hasw`
- To specify SP1 *and* Sandy Bridge:
  - `--constraint='sp1&sand'`



# General strategy for moving to SP3



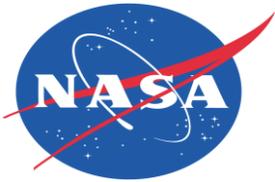
- Define a baseline of Sandy Bridge/SP1 results.
- Test your existing code on Sandy Bridge/SP3.
  - A recompile may be needed.
- Test your existing code on Haswell/SP3.
  - A recompile is **highly recommended**.
- Zero-diff results may not be possible.
- Tweak your code if desired.
  - Different optimizations available on Haswell.
- Modify job scripts for different cores and memory per node.



# Common compile-time issues & fixes



- Use the most recent compiler possible.
  - Intel 13.0.1.11 with `-xCORE-AVX2` will **fail** on Haswell.
  - Intel 15.x or later is **highly** recommended;  $\leq 14.x$  predates Haswell.
- PGI 15.x optimizes for build machine by default (`sand != hasw`).
  - Run on the build architecture, or...
  - ... use compiler options to remove platform-specific optimizations.
- All dependent libraries should be rebuilt for SP3...
  - ... **especially** MPI and thread libraries.
  - New IB HCA on Haswell nodes require MVAPICH rebuilds.
- Using `-xCORE-AVX2` may give different results on Sandy Bridge and Haswell unless `-no-fma` (no Fused Multiply-Add) is also used.
- When building older versions of GEOS, use the rebuilt version of baselibs, e.g.
  - `set basedir = /discover/swdev/mathomp4/Baselibs/GMAO-Baselibs-3_2_0-Intel_11.0.083-IMPI_4.1.2.040-gcc_4.3.4-SP3`



# Common run-time issues & fixes



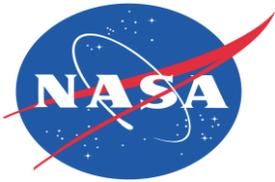
- Wait time is longer on all nodes while we upgrade Sandy Bridge nodes
  - Request the minimum resources to get your job done.
- SGI MPT
  - Only licensed for use on Haswell (SGI) nodes
  - Use `mpipexec_mpt` to launch tasks.
- Intel MPI can be slow with high Haswell node counts.
  - Many environment variables can be set to address this problem, e.g.
  - `setenv I_MPI_HYDRA_BOOTSTRAP slurm`
  - `export I_MPI_DAPL_UD_SEND_BUFFER_NUM=4096`
  - `export I_MPI_DAPL_UD_RECV_BUFFER_NUM=4096`
  - `export I_MPI_DAPL_UD_ACK_SEND_POOL_SIZE=4096`
  - `export I_MPI_DAPL_UD_ACK_RECV_POOL_SIZE=4096`
  - `export I_MPI_DAPL_UD_RNDV_EP_NUM=2`
  - `export I_MPI_DAPL_UD_REQ_EVD_SIZE=2000`
  - `export DAPL_UCM_REP_TIME=2000 # REQUEST timer, waiting for REPLY in millisecs`
  - `export DAPL_UCM_RTU_TIME=2000 # REPLY timer, waiting for RTU in millisecs`
  - `export DAPL_UCM_RETRY=7 # REQUEST and REPLY retries`
  - `export DAPL_ACK_RETRY=7 # IB RC Ack retry count`
  - `export DAPL_ACK_TIMER=20 # IB RC Ack retry timer`
  - `export DAPL_UCM_RETRY=10 # REQUEST and REPLY retries`
  - `export DAPL_ACK_RETRY=10 # IB RC Ack retry count`



# Common run-time issues & fixes (cont.)



- InfiniBand run time errors can often be fixed with environment variables:
  - `setenv I_MPI_DAPL_CHECK_MAX_RDMA_SIZE 1`
  - `setenv I_MPI_RDMA_CHECK_MAX_RDMA_SIZE 1`
  - Forces MPI message fragmentation.
  - Or bypass DAPL (Direct Access Programming Library) altogether after loading IMPI:
    - `setenv I_MPI_FABRICS shm:ofa`



# Common run-time issues & fixes (cont.)



- Use Intel MPI 4.1.2.040 or later for full SLURM support.
- Run-time MPI/fork/memory errors can often be fixed with The Magic Words™:
  - `limit stacksize unlimited`
  - `limit memorylocked unlimited`
  - Put this in your job script **and** your `.login/.profile/.cshrc`.
  - This solves a **lot** of run-time MPI problems, and has no negative effect, so just use it.
- Seemingly random job step aborts from SLURM can occur due to a run-time race condition, which can be addressed with:
  - `setenv HYDRA_LAUNCHER_EXTRA_ARGS="--input none"`



# Common script issues & fixes



- For Intel MPI, MVAPICH, and OpenMPI: The `--perhost` option for `mpirun` and `mpiexec_hydra` does not work; use `--ntasks-per-node`.
- Totalview does not work with SGI MPT.
  - `mpiexec_mpt` is a script here, but a binary at NAS.
- `ksh` does not properly export the user environment to SLURM jobs. Use `--export=all` with `sbatch` to ensure proper environment.



# Best practices



- Remove all references to Westmeres
  - SLURM: Remove `#SBATCH --constraint=west`
  - PBS (while NCCS wrappers are supported): Remove `#PBS proc=west`
- Only use the PBS wrappers if you must.
- Only use a core type if must, e.g. `#SBATCH --constraint=sand|hasw`
- Only use a partition if you must, e.g. `#SBATCH --partition=native`
- Only use a QoS if you must, e.g. `#SBATCH --qos=long`
- Let SLURM do the work for you, as much as possible.
  - Specify process or core count, not node count.
  - Specify tasks per node only if explicit undersubscription is needed.
  - Specify memory needed per core, if necessary, not per node.



## Best practices (cont.)



- Know your code!
  - Expected run time
  - Expected memory needed
- Port and test your code in stages.
- Recompile when possible to take better advantage of the improved hardware in Haswell nodes.
- Avoid severe node under/oversubscription – much more memory on Haswell nodes, but no swap space.
- Use The Magic Words™.
- SP3 login nodes may be used as Dali alternatives.
  - Similar memory, same filesystems and software available.



# What does the future hold?



- All Sandy Bridge nodes will migrate to SP3.
- The Sandy Bridge nodes will eventually be replaced.
- Eventually some nodes may be shared among small jobs.
  - This will require extensive testing prior to implementation.
  - User training will be provided prior to implementation.



# Open session: Your experiences

