Scheduler Optimization for Current Generation Cray Systems

CUG 2017

Morris Jette (SchedMD) jette@schedmd.com
Douglas Jacobsen (NERSC) dmjacobsen@lbl.gov
David Paul (NERSC) dpaul@lbl.gov

SchedMD

NERSC

Outline

- Knights Landing
- DataWarp
- Slurm scheduling algorithms
- NERSC environment

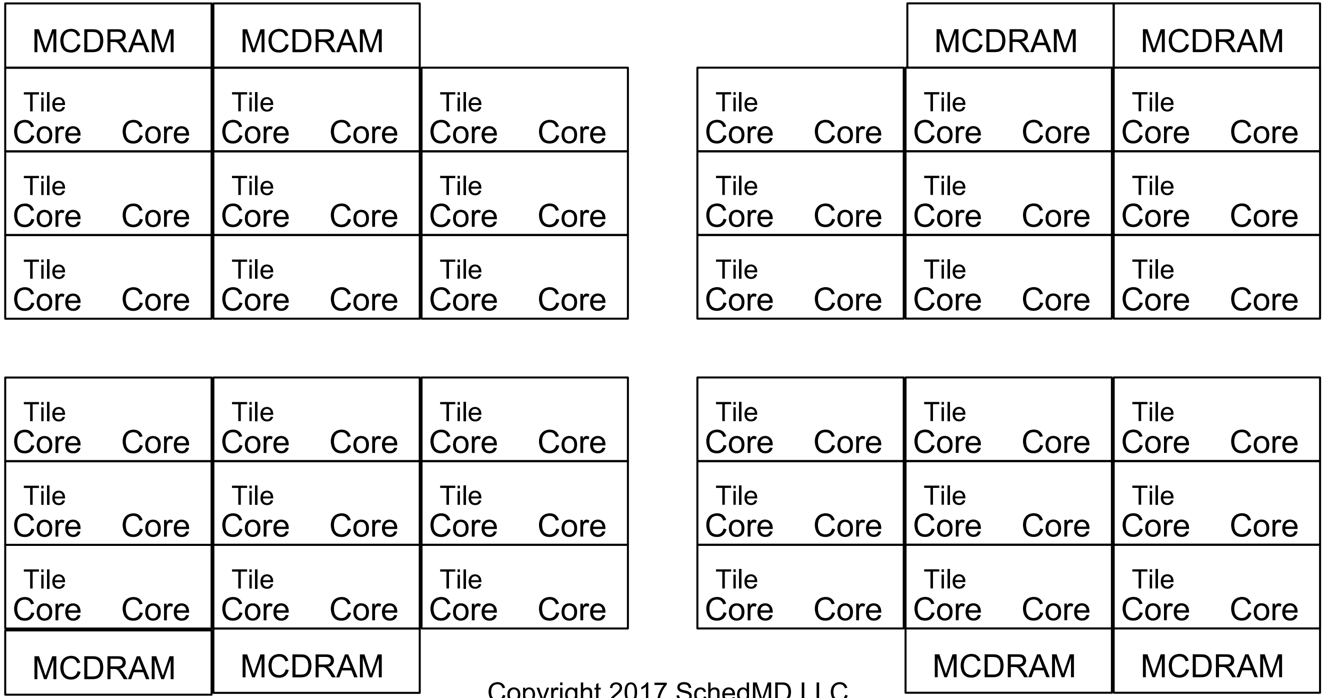
Intel Knights Landing (KNL)

- Up to 72 Airmont (Atom) cores with four threads per core
  - Arranged in 2-D mesh interconnect
- Up to 384 GB of "far" DDR4 RAM
- 8 – 16 GB of stacked "near" 3D MCDRAM (Multi-Channel DRAM), a version of high bandwidth memory (on package memory)
- Can be used as co-processor or self-boot (stand-alone processor)

KNL Modes

- Multiple NUMA modes
  - All to all, hemisphere, quadrant, sub-NUMA cluster 2, sub-NUMA cluster 4
  - Count of NUMA on node changes with mode change
- Multiple MCDRAM modes
  - Cache, flat (combined with primary memory), equal
  - Amount of high bandwidth memory changes with mode change
- Changing NUMA and/or MCDRAM mode requires node reboot

KNL SNC4 NUMA Mode
72-core Example



Slurm Node Features

- Used to establish node characteristics for scheduling purposes
- Split into two fields:
  - Available features: NUMA and MCDRAM modes which can be made available with a node reboot
  - Active features: Current NUMA and MCDRAM modes, possibly modified when computed node is booted

```
NodeName=nid00001
ActiveFeatures=quad,flat
AvailableFeatures=a2a,hemi,quad,snc2,snc4,cache,split,flat
```

Slurm Entities on a Node

- Sockets, Cores and Threads
- Slurm considers each NUMA as a socket
- The socket and cores-per-socket counts on a node can change when NUMA mode changes
  - Total core count constant (sockets x cores-per-socket = constant)

NodeName=knl123 ActiveFeatures=a2a,flat Sockets=1 Cores=64 Threads=4	↔	NodeName=knl123 ActiveFeatures=snc4,flat Sockets=4 Cores=16 Threads=4

Node Features: Scheduling

- User specifies required modes on job command line
  - Only AND operation supported, no OR, XOR, counts, etc.
- Job will be allocated nodes already in desired mode if possible
- Nodes will be rebooted only if needed
  - Boot time can consume minutes, avoid when possible

```
sbatch -C a2a,flat -n 72 -N1 my.bash
```

Node Features: Scheduling

- Slurm configuration parameter identifies expected node reboot time to optimize scheduling when comparing overhead of node reboot against waiting for running jobs to complete
- Nodes can only be rebooted when no active jobs
  - Could prove problematic to schedule if resource allocations not at node level (e.g. different cores allocated to different jobs)
- The job is billed for all resources from the time of allocation
  - Boot time is charged against job in fairshare and sacct
  - Boot time not counted against the job time limit

Slurm Node Features Plugin

- Provides mechanism to get and modify a node's MCDRAM and NUMA configuration plus boot the node
- Configuration file with administrative options
  - Users permitted to change node configuration
  - Modes available (may be a subset of those supported by the hardware)
- Two plugins available
  - knl_cray for Cray systems
  - knl_generic for generic clusters

knl_cray Plugin

- Cray's *capmc* and *cnselect* commands used to:
  - Read current MCDRAM and NUMA mode
  - Change MCDRAM and NUMA mode
  - Reboot nodes
  - Test node status
- All operations performed by *slurmctld* daemon on a service node (ctl1)

knl_cray Plugin

- If node mode change or boot fails, the *capmc* command currently does not identify the failing node
  - The job allocated those nodes will be requeued and held
  - Nodes previously allocated to the job can be used in subsequent resource allocations until the bad node(s) can be identified

Zonesort

- KNL application performance can benefit greatly if free pages are sorted at job start and/or periodically through job lifetime
  - Slurm's "--mem_bind=sort" option will run **zonesort** on the NUMA allocated to the job at its start

Caveats

- Slurm currently only supports homogeneous NUMA
    - 68-core KNL in sub-NUMA cluster 4 (SNC4) mode not supported
        - Results in unbalanced NUMA domains of [16, 16, 18, 18] cores
        - Slurm requires all NUMA domains to have same core count
- Consider CoreSpecCount configuration to minimize OS jitter
    - Reserves specific cores for system use (off limits to user)
    - Linux kernel can keep several cores busy under load

DataWarp Scheduling

- Slurm allocates and deallocates both persistent and job-specific DataWarp resources
- DataWarp resource limits can be configured by Slurm user, account, and/or Quality Of Service (QOS)
- DataWarp resources can be reserved using Slurm advanced reservation mechanism

DataWarp Scheduling Algorithm (1 of 2)

- Slurm estimates when pending jobs will be able to start
- Pending jobs expected to start soonest are allocated DataWarp resources (subject to limits and reservations) and files staged-in as requested
- Job-specific DataWarp resources allocated to pending jobs can be revoked for workload changes (i.e. higher priority jobs)
- Persistent DataWarp allocations must be explicitly deleted after creation, even if workload changes

DataWarp Scheduling Algorithm (2 of 2)

- Pending jobs will not be allocated compute resources until after DataWarp resource allocation and file stage-in completes
- DataWarp file stage-out starts after completion of job execution
- Slurm job record persists until after stage-out completes and job-specific DataWarp resource allocation delete

DataWarp Error Handling

- DataWarp errors are logged in the Slurm job record and the job is placed into HELD state
- User and/or system administrator responsible to investigate and respond to the failure (i.e. cancel or release the job)

DataWarp @ NERSC

- Cori - (NERSC-8) has 288 DW-Servers providing ~1.8PiB of SSD storage
- Tightly integrated with the WLM and directly connected to the Aries HSN
- Non-Recurring Engineering (NRE) contracts with Cray & SchedMD for *Phased* functionality
- Easy-to-use batch script commands for; allocation, configuration, staging and teardown
- Three pools configured with differing characteristics: wlm_pool, sm_pool, dev_pool
- Proven to be very performant

NERSC DataWarp Usage

Job Instances

- Duration equals life of job
- Single-user, single-job
- Quotas implemented
- Staging ~12TiB/.5Mil files is ~30 minutes

Persistent Instances

- Long lasting- specific removal
- Multi-job, multi-user (Posix permissions)
- Counted against *owner's* quotas
- Well suited for very large read-only data sets shared by multiple users

Current Datawarp Status

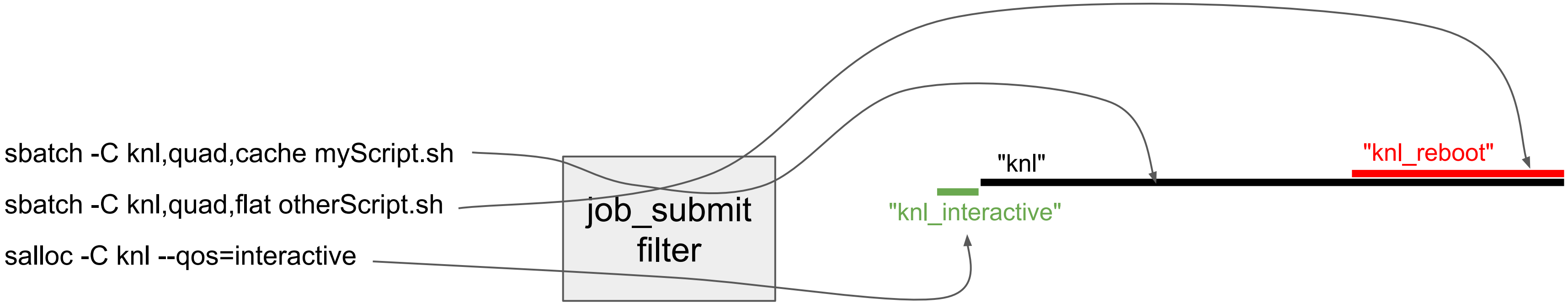
<u>Successes</u>:

- Performant
- Staging sans Compute
- Reliable hardware
- Failures isolated from rest of system
- Integration with WLM
- Improved centralized logging
- SSD Protection from *“bad user”* - read-only mode
- Rapid fix turnaround (esp. WLM)
- Continuing user adoption

***<u>Not</u>* <u>Successes</u>**:

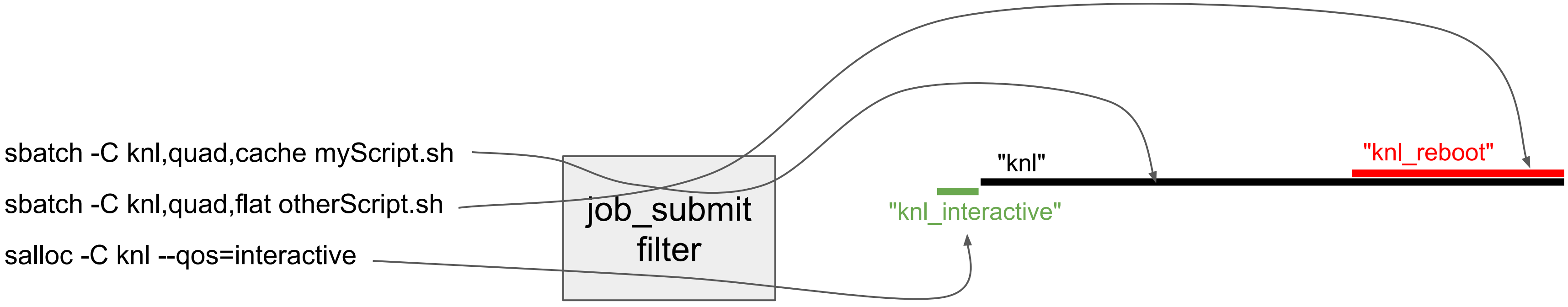
- S/W Updates = new bugs or regression
- Delays with *Phased* functionality
  - Caching modes affected most
  - Full Transparent Cache = UP05
- Create/Delete Persistent requires a Compute allocation
- Debugging failures can be daunting
- Staging of very large data sets (timeouts)

KNL Scheduling - NERSC



- Cori is a heterogeneous system, has both haswell and knl processors; users are required to request which architecture their job needs (-C haswell, -C knl) [why should haswell be "default"]
- knl mode changes are costly - time and potential job loss (if boot fails)
- Currently we attempt to minimize the scale and quantity of mode changes (until reliability improvements, cost reductions (boot time)
- -C knl, -C knl,quad cache routed to "knl", all other modes to "knl_reboot"
- Can dynamically reconfigure knl and knl_reboot partitions to meet any capability need without impacting queued jobs, low administrative cost

KNL Scheduling - NERSC



- Slurm 17.02 also includes two important scheduling optimization to reduce risk/cost of mode changes
  - delay-boot : jobs requiring a mode change are delayed by a configurable (user-overridable) time. This allows the system to prefer to run jobs on nodes already booted to a given mode. Wait time required to change the configuration of the system. Defaults to 48 hours at NERSC.
  - boot-time (knl_cray.conf): scheduler can now include a constant estimate of time required to boot nodes. This allows better future planning, and minimizes risk of jobs running into advanced reservations

Questions?

More information available online:

https://slurm.schedmd.com/documentation.html
https://slurm.schedmd.com/intel_knl.html
https://slurm.schedmd.com/burst_buffer.html
https://slurm.schedmd.com/slurm_ug_2011/SLURM.Cray.pdf
https://slurm.schedmd.com/SLUG16/KNL.pdf
https://slurm.schedmd.com/SLUG15/Burst_buffer.pdf