Managing Diversity in Complex Workloads in a Complex Environment

Slurm User Group 2017
Nicholas P. Cardo, CSCS
September 25, 2017

Special Thanks to Miguel Gila and Mark Klein for their contributions to this work and presentation.
Chocolate, Cheese, and HPC
Topics To Cover

- Piz Daint
- Customer Requirements
- Policies
- Implementation Details
- Metrics
- What’s Next?
Introducing Piz Daint

- 5320 Hybrid Nodes (Cray XC50)
  - 1 x Intel(R) Xeon(R) CPU E5-2690 v3
  - 1x NVIDIA Tesla P100-PCIE-16GB
  - 64 GB Memory

- 1430 Multicore Nodes (Cray XC40)
  - 2 x Intel(R) Xeon(R) CPU E5-2695 v4
  - 64 – 128 GB Memory

- Slurm Version 17.02.7

#3 Top500 June 2017 List
Customer Type #1

- 25 dedicated multicore nodes
  - Restricted to only these nodes
- High throughput type workload
- Hyperthreading enabled, and used
- Nodes are dedicated to this customer
- Custom filesystems
  - Use of specialized external filesystem
  - Use of Cray DataWarp
  - Use of local Scratch space
- Nodes configured to run multiple jobs per node

This work has been supported by the Swiss National Science Foundation
Customer Type #2

- 270 multicore node equivalent hours
  - No GPUs
  - Cannot exceed
- Calculated over 3 months
- Customer managed share distribution
- No node sharing
Customer Type #3

- Conglomeration of experiments
- 1500 – 2000 users
- Access granted through a peer review process
- Can exceed granted hours
  - Limited to idle time only
  - No commitment for service
Customer Type #4

- Actually multiple customers
- All governed by the same rules
- On-Demand type purchased packages
- Limited to purchased hours
  - Cannot exceed
  - Can purchase more
- Quick turnaround
Customer type #5

- Actually multiple customers
- Cannot exceed their allocated hours
Policies
Project Type Identification

- cp – Contractual Partner (customers #2 & #5)
- prod – Production (customers #3)
- 2go – CSCS2GO (customers #4)

What about Customer Type #1?
Project Requirements

- Any “cp” project cannot exceed their allocated hours.
- Any “prod” project may exceed their allocated hours, but with no expectation of service. (consume idle cycles)
- Any “2go” project should bump to the head of the list and start as soon as possible.
- Each project can be restricted to only GPU nodes, only Multicore nodes, or anywhere in the system.
Other Requirements

- Fast turnaround for small debug jobs.
- Fast turnaround for pre/post-processing jobs.
- Prevent jobs from being submitted if:
  - User’s SCRATCH inode quota has been exceeded.
  - User’s “account” has exceeded their allocated time, with exception.
- “prod” projects can submit jobs with no expectation of service only if they’ve exceeded their allocated hours.
- Simple “queue” structure.
- Maximum job size when requesting GPU or Multicore nodes. (can be different)
- Multicore nodes have two different memory sizes.
- Handle thousands of jobs/steps.
- Jobs cannot span both GPU and multicore nodes, unless arranged.
- Data Transfer Capability without allocating compute nodes.
Implementation Details

- The UI model should hide the implementation model's complexity from the user.
- The UI model should be consistent with the implementation model.
Slurm Partitions with Restrictions

- **debug**
  - 8 Multicore + 8 GPU nodes, MaxTime=00:30:00, MaxNodes=4, PriorityJobFactor=30, PriorityTier=30
- **prepost**
  - Multicore only, MaxTime=00:30:00, MaxNodes=1, PriorityJobFactor=20, PriorityTier=20
- **normal**
  - MaxNodes=2400, MaxTime=1-00:00:00, PriorityJobFactor=10, PriorityTier=20
- **large**
  - MaxNodes=4400, MaxTime=12:00:00, PriorityJobFactor=20, PriorityTier=20
- **low**
  - MaxNodes=2400, MaxTime=06:00:00, PriorityJobFactor=0, PriorityTier=10
- **xfer**
  - MaxNodes=1, MaxTime=1-00:00:00, LLN=YES, PriorityJobFactor=20, PriorityTier=30
- **wlcg**
  - MaxNodes=1, MaxTime=5-00:00:00, LLN=YES, MaxCPUsPerNode=68, PriorityJobFactor=10, PriorityTier=30, +QOS
- **2go**
  - MaxNodes=2400, MaxTime=1-00:00:00, PriorityJobFactor=30, PriorityTier=20
- **total**
  - MaxNodes=6200, MaxTime=08:00:00, PriorityJobFactor=30, PriorityTier=40
Priority Tiers

- **PriorityTier=40**  *restricted access*
  - total  PriorityJobFactor=30

- **PriorityTier=30**  *dedicated nodes*
  - debug  PriorityJobFactor=30
  - xfer  PriorityJobFactor=20
  - wlcg  PriorityJobFactor=10

- **PriorityTier=20**  *normal production workload*
  - 2go  PriorityJobFactor=30
  - prepost  PriorityJobFactor=20
  - large  PriorityJobFactor=20
  - normal  PriorityJobFactor=10

- **PriorityTier=10**  *no expectation of service*
  - low  PriorityJobFactor=0
Priority Calculation Parameters

- PriorityParameters = (null)
- PriorityDecayHalfLife = 40-00:00:00
- PriorityCalcPeriod = 00:05:00
- PriorityFavorSmall = No
- PriorityFlags =
- PriorityMaxAge = 14-00:00:00
- PriorityUsageResetPeriod = QUARTERLY
- PriorityType = priority/multifactor
- PriorityWeightAge = 172800
- PriorityWeightFairShare = 259200
- PriorityWeightJobSize = 0
- PriorityWeightPartition = 172800
- PriorityWeightQOS = 500000
- PriorityWeightTRES = (null)

Make sure your Weights are large enough to create a large spread of priorities.
Site Policy Enforcement

- Enforce site custom policies
- Handles complex policy decisions
- Middleware between Site and Slurm
How It Works

job_submit.lua

mem cache

ckproject

exit

Slurm

CSCS DB

Job Details

Return Code

single blocking thread
How It Works

job_submit.lua → ckproject

Job Details
Return Code

exit

Slurm

CACHING is AWESOME!!!

SINGLE BLOCKING THREAD

CSGS DB
Policies Enforced

- Is project allowed to use the specified partition?
- Does the job exceed the maximum node count for that type of nodes?
- Has the user exceeded their SCRATCH inode quota?
- Has the “Account” exceeded their quota?
- Is the ”Account” expired?
- Is the “Account” permitted to use the GPU nodes?
- Is the “Account” permitted to use the Multicore nodes?
- Is the “Account” permitted to use the “low” partition?
- Is the user batch disabled?
Jobs Submitted per Day

September 2017

Slurm User Group 2017
Average Queue Wait Time (September 2017)

Production wallclock time limit is 24:00:00.

time_start – time_eligible = Queue Wait
Under Consideration

- **Data Cache**
  - Cache the output from command strings.
  - Reduces hits on the database.

- **ckproject performance improvements**
  - Reduce hits on the database.
  - General code speedup.

- Restrict the number of queued/running jobs per user per partition
General Comments

- Be careful to balance the amount of work being done during a scheduling cycle and a backfill cycle.

- Still have user related issues:
  1. Loops...
     - `watch -n 0.5 squeue | grep username`
  2. Exponential growth
     - `sbatch --array=1-20`
     - `sbatch --nodes=100`
     - ...
     - `sbatch thesamefile`
  3. Interactive background tasks in a loop...
     - `srun file &`
Thank you for your attention.