Slurm at CEA

status and evolutions
Slurm at CEA
Agenda

Supercomputing projects

Slurm usage and configuration specificities

Planned work, research and evolutions
Slurm at CEA

Supercomputing projects
TERA

- Project started in 1998
  - Part of the Simulation Project for French Nuclear Deterence

- Tera-100 supercomputer
  - Installed in 2010
  - 1,25 PF/s
  - Owned and operated by CEA

- Hosted at CEA Defense computing center
PRACE
(PaRtnership for Advanced Computing in Europe)

- Project Started in 2007

- **Curie** Supercomputer
  - First French Tier-0 supercomputer for the PRACE project
    - 2 stages installation in 2011-2012
    - 1.6 PF/s
  - Owned by **GENCI**
    (Grand Equipement National pour le Calcul Intensif)
  - Operated by **CEA**

- Hosted at the **TGCC** « Très Grand Centre de calcul du CEA »
  - CEA computing facility
Slurm at CEA

Supercomputing projects

CCRT

(Computing Center for Research and Technology)

- French Industrial and research partners shared computing center
  - Hosted by CEA/DAM/DIF since 2003

- **Airain** Supercomputer
  - CCRT-C machine
    - 3rd phase of the CCRT project, installed in 2012
    - 200 TF/s
  - New HTC requirements (genomic studies)
    - Large number of small jobs
    - Job arrays
  - Operated by CEA

- Hosted at the **TGCC « Très Grand Centre de calcul du CEA »**
  - CEA computing facility
TERA+

- Evaluation and validation of HW and SW prototypes

- CEA PRACE prototypes
  - Connected to the PRACE infrastructure, Grid services and community

- CEA R&D Plateform
  - Autonomous computing center reflecting the production systems

- Next evolution stage and focus point
  - R&D phase of T1K
  - Will help to define and validate the main concepts of the next generation systems
    - Including SLURM related studies
Slurm at CEA

Slurm usage and configuration specificities
Slurm at CEA
Slurm usage and configuration specificities

Slurm footprint

- All major clusters introduced since 2009 and operated by CEA
  - Tera+: forttoy, inti
  - Tera: Tera-100, Visualization cluster
  - PRACE: curie
  - CCRT: airain

Support

- SLURM supported by supercomputer vendor for large machines of the TERA/PRACE/CCRT projects
  - One single vendor for now: BULL

- Level 3 support on the R&D cluster forttoy
  - Provided by SchedMD LLC

- Community version with community support for other small scale clusters
Configuration specificities

- Core/Memory level allocation
  - Flexible enough as it allows node level allocations too
  - Best-fit allocation across sockets
  - Task/cgroup for confinement/affinity

- Tree topology description
  - Optimize the number of leaf switches used by a job

- Multifactor Scheduling logic
  - QoS support
  - Fairshare support

- Backfill scheduling
Configuration specificities

- Same ideas and principles across the different machines
  - Fairshare scheduling not used on the Tera project
    - but planned
  - Fairshare logic adapted for TGCC use cases
    - In-house development (see details in the next slides)
  - Kerberised FS (NFS) accessed using slurm/auks on some machines
    - In production on small clusters but need some improvements for large clusters

- SLURM versions in production
  - Bull flavors of slurm-2.4.4 (plus local patches when necessary)
  - Backports of dev branch patches when necessary

- Wrapped in a CEA set of scripts and commands called « Bridge »
  - Automate per machine/user/project configuration
  - Simplify the integration of external tools and debuggers
  - Abstract the underlying resource manager / Batch system
    - Mostly the same interface as when we were using LSF/RMS
  - Transparent requests enhancement
    - Ex : automatic request of exclusive nodes when requested cores > threshold
**Improved hierarchical fairshare priority factor**

- Need to manage complex account hierarchies
  - Ex: a share of Curie is allocated to our industrial partners
  - They want to split their shares between their internal divisions or projects
  - Up to 4 levels of hierarchy in total

- Slurm hierarchical fairshare doesn't handle this well
  - Priority values become lower as the tree becomes deeper
    - Low priorities overall
    - Unfair when the tree is not balanced

- The ticket-based algorithm introduced in Slurm 2.5 doesn't address our use-case
  - Priorities fluctuate depending on the queue state (troubling for users)
  - Unfair depending on the distribution of active accounts

Actual usage does not converge towards allocated shares if all accounts use the machine greedily
Improved hierarchical fairshare priority factor

- Developed an improved version of the classic fairshare algorithm
  - Fair priority factors for unbalanced trees
  - Able to use the entire range of priority factors if needed
  - More info in dedicated slides...

- Running on our clusters at TGCC and CCRT
  - Real usage is now closer to shares
  - Partners can subdivide their shares if needed
  - Good feedback from our users

- Will be contributed upstream if the community is interested
  - Small patch (approximately 100 lines of code)
  - Could replace the current non-ticket based algorithm or live alongside it
Slurm at CEA

Planned work, research and evolutions
Different areas of interest identified

- Power aware scheduling
  - Flexibility, adaptability and efficiency in power supply and consumption

- Hierarchical communication architecture
  - Enhance messages aggregation and better isolate independent sections

- Centralized RM architecture
  - Merge multiple clusters to manage shared resources within the RM
    - Global FS, licences, ...
  - Replace the In-House scheduling logic in Tera-100 Meta-Scheduler

- Scheduling and accounting
  - Add features to the fairshare scheduler
Power aware scheduling

Main goals

- Optimize the amount of power required to operate a set of resources
- Cap the amount of power available to the proposed resources on demand
- Optimize the global performances in a limited/capped amount of power

Envisioned means

- Get access to physical power supply details through a dedicated layout and leverage that information in the resource manager
- Evaluate the power requirements of jobs based on their resource requirements
  - Requested frequency (DVFS), number of cores and mem per node, accelerators, ...
  - Power considered as a « backstage » (indirect) resource
- Schedule in respect of the amount of available power and the power supply chain
  - Respect intermediate amount in the power supply chain

Envisioned evolutions

- Extend the concept to other « backstage » resources like cooling

Work in progress in the Perfcloud project

- POC expected by the end of 2013
Hierarchical communication architecture

Main goals

- Differentiate components and roles in a hierarchical way
  - Controllers | Gateways ... | Compute nodes
- Optimize the communication paths between compute nodes and controllers
  - Reduce the amount of processed RPC on the controllers
    - Aggregation and/or concatenation of messages in compound requests
  - Leverage known network details
  - Reduce the noise on the compute nodes (limited gateway role)

Envisioned means

- Get valuable informations through the layouts framework and leverage that information in the resource manager
  - Components description and architecture layout
  - Administrative network topology layout
- Enhanced reversed tree communication using gateways
  - Aggregation and/or concatenation of messages in a new compound message

Work planned for T1K R&D

- POC expected by the end of 2014
Centralized RM architecture

- Main goals
  - Optimize the sharing of global resources among clusters in the compute center
    - Ex: licenses, global storage bandwidth,...
  - Get access to a centralized scheduling logic
    - Global management and fairshare of all the resources
    - Automatic rerouting of jobs to available resources

- Envisioned means
  - Move Fairshare logic from our in-house Meta-Scheduler to the RM of the clusters
    - Metascheduler no longer have enough topological details to take smart decision
  - Merge clusters RM into one single centralized RM
    - Centralizing the fairshare logic
    - Manage all the resources including the globally shared resources
  - Delegate some scheduling decisions to sub-domains
    - Delegation of scheduling of steps, accounting, ...

- Study planned for T1K research and beyond
Scheduling and accounting

- Fairshare scheduling lacks some important features
  - Administrative control over the accounted cpu usage
    - The accounted usage can only be reset
    - Removing a job from the accounting is not possible (job refund)
    - The half-decay period cannot be changed

- Per partition/resource accounting and shares
  - Our projects/partners have different shares for each partition
    - Standard, large and GPU nodes
  - Have to rely on separate accounts for each partition
    - Hidden to users thanks to « Bridge »
  - Simulated independant fairshare trees for each partition
Thank you for your attention

Questions?