DE LA RECHERCHE À L'INDUSTRIE

CEA Site Report

www.cea.fr

SLURM User Group Meeting 2012 | Matthieu Hautreux <matthieu.hautreux@cea.fr>

Agenda

Supercomputing Projects

SLURM usage

SLURM related work

Supercomputing Projects

Supercomputing Projects

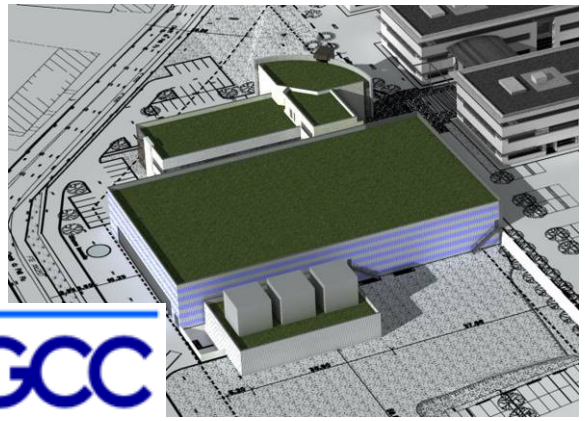
TERA

- Project started in 1998
  - Part of the Simulation Project for French Nuclear Deterence
- **Tera-100** supercomputer
  - Installed in 2010
  - 1 PF/s
  - Owned an operated by **CEA**
- Hosted at **CEA** Defense computing center

Supercomputing Projects

PRACE (PaRtnership for Advanced Computing in Europe)

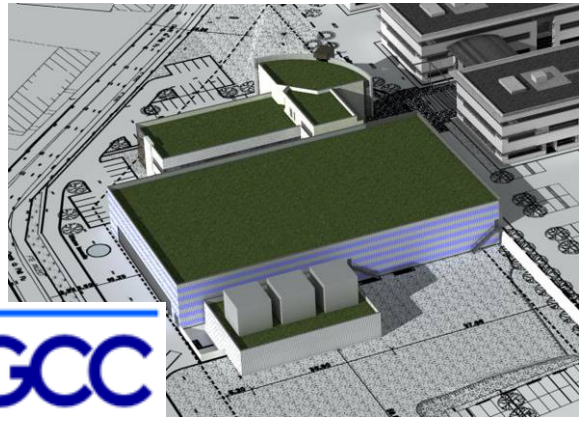
- Project Started in 2007
- **Curie** Supercomputer
  - First French Tier-0 supercomputer for the **PRACE** project
    - 2 stages installation in 2010-2011
    - 1.6 PF/s
  - Owned by **GENCI** (Grand Equipement National pour le Calcul Intensif)
  - Operated by **CEA**
- Hosted at the **TGCC** « Très Grand Centre de calcul du CEA »
  - CEA computing facility



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



Supercomputing Projects

TERA+

- CEA R&D Plateform
  - Autonomous computing center
  - Evaluation and validation of HW and SW prototypes
- Next evolution stage and focus point
  - R&D phase of T1K
  - Will help to define the main concepts of the next generation systems at CEA
    - Including SLURM related studies

SLURM Usage

SLURM Usage

Footprint

- All major clusters introduced since 2009 and operated by CEA
  - Tera+ : fortoy
  - Tera : Tera-100
  - 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 fortoy
  - Provided by SchedMD LLC
- Community version with community support for other small scale clusters

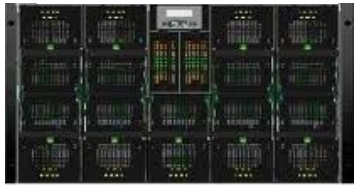
SLURM Usage

Hardware specificities

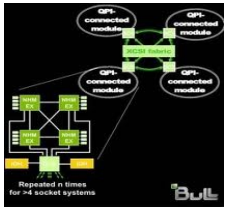
- Bull hardware
  - Bullx S6010 nodes
    - 4 Nehalem sockets
      - 32 cores


  - Bullx B510 thin nodes
    - 2 Sandy Bridge EP sockets
      - 16 cores


  - Bullx B505 blades
    - 2 Westmere / 2 NV M2090
      - 8 cores



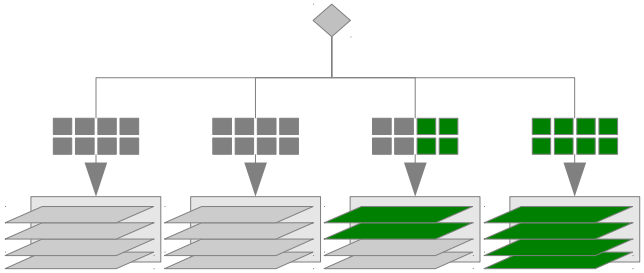

  - Bullx S6010 nodes with BCS (Bull Coherency Switch)
    - 4x 4 Nehalem sockets
      - 128 cores

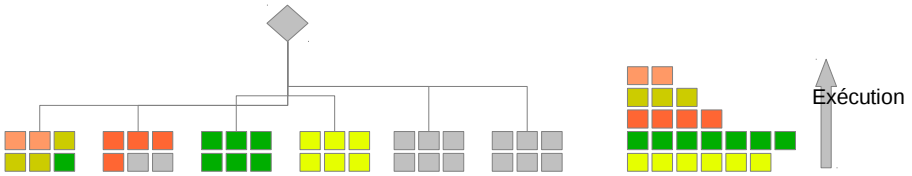



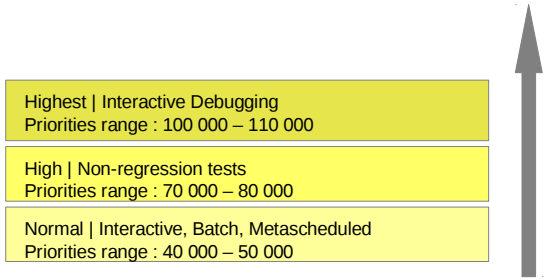
- Infiniband interconnects only
  - Tree/Pruned-Tree topologies

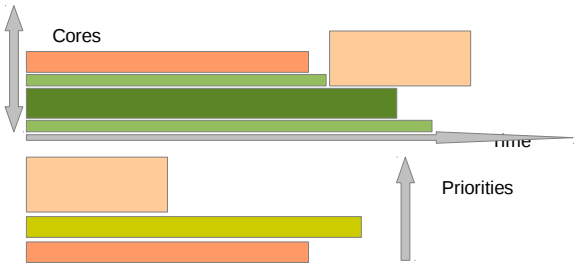
Configuration specificities

- Core/Memory level allocation
  - More flexible 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









SLURM Usage

Configuration specificities

- Large usage of advanced reservations
  - Especially on TGCC machines to ensure resources to Grand Challenges or training sessions
  - Also used to planify maintenance period
- SLURM Spank framework
  - Kerberos credential support
    - using Auks
  - X11 support through tunneled SSH
    - Using local dev slurm-spank-x11
  - OOM-Killer score adjustment
    - Using local dev spank-oom-adj

SLURM Usage

Configuration specificities

- Same ideas and principles across the different machines
  - The only difference is the Fairshare scheduling not used on the Tera project
- SLURM versions in production
  - Bull flavors of slurm-2.3.x and slurm-2.4.x
  - 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 ressource manager / Batch system
  - In the process of being released as an opensource project

Feedback

- Sanity checks
  - Large number of checks to perform
    - Have a high impact on the nodes when ran
  - May kill a job because of an unused faulty resource
    - Degrade time-to-solution and robustness of the system
  - Current thoughts
    - Move Sanity checks to prolog/epilog and no longer use the periodic check
      - At least reduce the periodic tests to a vital minimum
    - Pros/Cons prolog vs epilog
      - Prolog :
        + ensure that all works well right before the job execution
        - delay the execution of the job, decrease the responsiveness
      - Epilog :
        + do not decrease responsiveness, only increase return-to-service time
        - issues that appears between epilog and next jobs are not took into account
    - Do checks in epilog and perform periodic check on idle or partially idle nodes to detect issues in advance (remove the main drawback of sanity checks in epilog)

Feedback

- Scalability
  - Concerning helper tasks running on compute nodes
    - i.e. : Prolog/Epilog/HealthCheck, Pam_slurm, Spank plugins
  - Need to contact the controller to get mandatory state information for their internal logic
    - i.e. : sinfo -n $(hostname) to get the state of the node in epilog, squeue to get the information concerning concurrent jobs on the node, ...
    - A large load is induced by simultaneous helper tasks
  - With thousands of nodes and hundreds of jobs, the controller is stuck too often
    - i.e. : Large number of threads only waiting to process sinfo requests
  - More states should be propagated from slurmctld to slurmds to avoid N->1 callbacks
    - Current workaround is to use « scontrol listpids » and try to guess what is happening on the node without disturbing the controller (other jobs, other jobs from the same user, ...)

Feedback

- Responsiveness
  - Loaded controllers process requests with a high level of concurrency
  - A few interactive user/admin RPCs overwhelmed by a large number of daemons messages coming from the compute nodes (see previous slide)
  - No distinction between « control flows » of user/admin requests and « data flows » of internal mechanisms
  - A kind of QOS would be great to separate the flows and provides different levels of QOS per RPC and per initiator

Feedback

- Internal communication tree
  - Generic tree with configurable width to contact all the involved nodes
  - No easy way to get the tree used to communicate between a particular nodeset
    - Mandatory to understand the root cause of a communication issue with hundreds/thousands of nodes
  - Logical tree not mapped to the physical underlying control network
    - May cause significant overhead to the physical network layer in some situation exp : aggregation of stacked ethernet switches federated by a single router

Feedback

- Memory consumption and monitoring
  - The most problematic issue on a day-to-day basis
  - Memory support in task/cgroup could help but ...
    - RHEL6 kernels suffer from SMP locality issues and degrades performances with memory cgroup
    - OOM-killer external actions are not easy to track and associate with the initiator
      - Need to pursue the proposition of Mark Grondona concerning cgroup event management

SLURM related work

SLURM related work

Studies

- Scalablity study performed with Yiannis Georgiou from Bull
  - Should be detailed during this user group by Yiannis
- 2 internships
  - SLURM layout framework (May-Jul 2012, but to be continued in 2013)
    - Evaluate the possibility to provide a generic framework to describe relations between nodes and other components in a supercomputer
      - Racking, power supply cables, cooling pipes, ethernet control network, ...
    - Evaluate the possibility to use this framework to enhance the communication and scheduling logic of SLURM
  - SLURM topology (Jun-Dec 2012)
    - Evaluate state-of-the-art interconnect topologies and their properties
    - Assess the direct eligibility of SLURM to manage the new ones or think about the way to manage them efficiently

SLURM related work

Troubleshooting and features

- Some patches delivered to Bull as our main support contact
  - Most/All of them integrated or corrected
    - Only get rid of the local soft/hard memory limits proposal of the last user group
- A few patches proposed directly to the community
  - We want to still say « Hello » to the community sometimes :)
- A few enhancements
  - Modification of the task/cgroup logic (with the help of M.Grondona)
  - Reorganization of slurmstepd logic to better handle secured FS

Thank you for your attention

Questions ?

Commissariat à l'énergie atomique et aux énergies alternatives
Centre DAM Ile-de-France | Bruyères-le-Châtel 91297 Arpajon Cedex
T. +33 (0)1 69 26 40 00 |
Etablissement public à caractère industriel et commercial | RCS Paris B 775 685 019

DAM/DIF
DSSI
SISR