Outline

- SLURM Usage, Configuration and other Specificites
- Ongoing studies
- Interesting topics
SLURM Usage, Configuration and other Specificities
SLURM Usage at CEA

- **TERA+ CEA R&D project**
  - R&D platform to assess HW/SW technologies for next machines

- **TERA Project**
  - TERA100, a petaflop machine
  - First large scale system to use SLURM at CEA

- **PRACE (PartnerRship for Advanced Computing in Europe) Project**
  - CEA in-kind prototypes
  - TGCC Petaflopic machine (Curie)

- **Most of the installed clusters at CEA are using SLURM since 2007**
  - Sharing the same configuration principles

- **Basic Submission/Execution/Monitoring commands wrapped**
  - Using an inhouse product (**bridge**)
  - To mask resource manager specificities and ease migration
  - To tweak and adapt behaviors automatically
    - Based on the compiler, the initial request, ...
Configuration : Scheduling strategies

- **Allocation granularity** *(slurmctld)*

  - Core and memory allocation *(select/cons_res – CR_Core_Memory)*
    - Exclusive allocations of both memory and cores inside nodes
    - \( \text{MaxMemPerCore} = \frac{\text{Node Memory}}{\text{Cores Per Node}} \)
    - ✔ Help to reduce locality effect and account usage coherently
    - ✔ But requires homogeneous nodes with slurm < 2.3 (no support of partition specific values of MaxMemPerCore before 2.3)

  - Exclusive allocation of nodes on demand *(--exclusive in SLURM)*
    - ✔ Better for large tightly coupled jobs
    - ✔ Can be automatically set based on a configurable threshold with **Bridge**
Configuration : Scheduling strategies

- **Topology awareness and resources selection** *(slurmctld)*

  - **Inter-node** *topo/tree* to represent pruned tree IB topology
    - Best fit selection of switches
    - Best-fit selection of nodes in the switches

  - **Intra-node** topology with *sockets/cores/threads* description
    - Best-fit selection of cores inside sockets
    - Block allocation by default
    - No NUMA support in SLURM
    - ✔ On the CEA ongoing studies list
Configuration: Scheduling strategies

- **Scheduling logic** *(slurmctld)*

  - **Multifactor priorities** logic (priority/multifactor - QOS/Age/Fairshare)
    - **QOS** for interactive highly prioritized jobs and limits management
      - ✔ Orthogonal to the partition concept
      - ✔ Partition used to gather homogeneous HW
    - **Age** (~FCFS) prioritization (TERA) / **FairShare+Age** (TGCC)
      - ✔ Inside a QOS priority range

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Priority Range</th>
<th>Limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highest</td>
<td>Interactive Debugging</td>
<td>100 000 – 110 000</td>
<td># jobs ; # submissions ; MaxTime</td>
</tr>
<tr>
<td>High</td>
<td>Non-regression tests</td>
<td>70 000 – 80 000</td>
<td># jobs ; # submissions ; MaxTime</td>
</tr>
<tr>
<td>Normal</td>
<td>Interactive, Batch, Metascheduled</td>
<td>40 000 – 50 000</td>
<td># jobs ; # submissions ; MaxTime</td>
</tr>
</tbody>
</table>
Configuration : Scheduling strategies

- Scheduling logic (**slurmctld**)

  - **Backfilling** logic (sched/backfill)
    - Particularly interesting for TERA workload
    - ✓ adaptative execution time using app level checkpoint/restart
    - Reduces starvation of big jobs while optimizing throughput
    - Should help to have users describing execution time correctly on TGCC

![Diagram showing scheduling logic](image-url)
Configuration : Resources constraints and affinity

**Cores (slurmd – task/affinity TaskPluginParam=Cpusets,Cores)**

- Allocated cores containers for jobs
  - Prevent users from using unallocated cores on nodes
- Automatic binding to cores for best efficiency of jobs
  - Using cpusets (except for salloc/mpirun executions)
  - Using a block distribution by default (-m block:block by default)
- Cgroups support in dev (task/cgroup)
  - CEA/Bull dev for SLURM
  - Currently available in slurm-2.3

**Memory (slurmd – jobacct_gather/linux Frequency>0 )**

- Memory usage collected regularly
  - Configurable interval to reduce noise (60s)
- Jobs killed if memory limit exceeded due to RSS usage
  - Does not really fit the requirement
- Cgroups support in dev (task/cgroup)
  - RSS+Swap usage could be took into account
  - Cgroup memory support can be cost effective
  - Promising solution but not used in production
Configuration : Accounting and Users management

- « Cluster centric » database (*slurmdbd*)
  - Accounting data including useful resources consumption information
    - Metascheduler fed using this data (TERA)
    - Accounting digest generated and included at the end of each batch job
  - Users and accounts definition
    - Synchronized from external sources (LDAP, Metascheduler, ...)
      - ✔ In-house scripts based on *sacctmgr* cmdline
  -Limits and QOS definition
  - MySQL DB backend
Specificities : MPI Integration

- **OpenMPI based implementations**

  - **SLURM support in OpenMPI**
    - Historical approach
    - Salloc/mpirun mode
      - Uses srun to launch one *orted* daemon per node
    - Do not fully inherit SLURM launcher capacities and scalability
      - Still require a first step to init out-of-band communication paths
    - Problems to understand complex core level allocations
      - For hybrid MPI/OpenMP (-c option no managed by mpirun)
      - For adaptative multi-steps allocations
  
  - **OpenMPI support in SLURM**
    - Reserved ports for out-of-band OpenMPI communications in advance
      - Speed up comm paths init
    - Requires an recent OpenMPI version
    - Each process execution managed by SLURM
      - Better handle affinity for hybrid jobs
    - Partial debugging available with Totalview
    - Default mode for TERA
      - Using BullxMPI, Bull MPI layer based on OpenMPI
Specificities : Addons

- **SLURM Spank Framework (CEA Dev)**
  - Kerberos support using *spank-auks*
    - Requires a working AUKS infrastructure (http://sourceforge.net/projects/auks/)
  - X11 support with OpenSSH using *spank-x11*
    - Both interactive and batch mode
    - Requires SSO or equivalent (stackable on top of *spank-auks*)
  - Kernel scheduling policy selection using *spank-setsched*
    - Helps to use an optimized policy if/when necessary
  - OOM-Killer score adjustment of tasks using *spank-oom-adj*
    - Used to declare user tasks launched by SLURM as best candidates

- **Sanity checks : (slurm – HealthCheckProgram=...)**
  - Periodic sanity checks
    - Hard disks
    - IB links
    - Lustre FS access
  - Automatically drain faulty nodes (proactive action to app crash)
    - First event that trigger the diagnose/repair/test workflow
Ongoing studies and Feedback
Ongoing studies and feedback

- **Scalability in number of jobs (TERA100)**
  - About 10K jobs can be submitted and started in < 120s
    - 10 clients
  - About 10K jobs can be submitted in pending state in <60s
    - 10 clients
    - With a modified defer mode
      - ✔️ Local patch to ensure no call to schedule() in batch submission when defer mode is activated (patch to be proposed)
  - Management of 10K jobs is ok
    - ☺️ No problem of management while the tests are running
    - ☻️ Thanks to Bjorn-Helge Mevik 's patch to speed up backfilling
      - ✔️ Unresponsivness for 20 minutes before that at the end of the 10K jobs

- **Job preemption using a « sudden death » approach**
  - Ensure a maximum wait time to access resources for specific QOS
  - Based on Grace Time (CEA/Bull dev for 2.3)
  - Evaluation not yet completed

- **Cgroups support for tasks compartmentalization**
  - Including cores, memory and devices support (accelerators)

- **GPU integration**
  - Exclusive allocation of nodes that have GPUs for now
Ongoing studies and feedback

- **Soft/Hard Memory limits**
  - Ensure job execution time shortening when soft mem limit is reached
    - Notion of grace time
  - Ensure job cancellation when hard mem limit is reached
    - Without additional delay
  - Partial implementation that is functional but not perfect
    - Would require more modifications in SLURM codes for a complete support
  - In production on TERA100
  - General interest for such a feature in the main branch of SLURM?
Interesting topics
Interesting topics

• **QOS advanced features**
  - QOS activation/desactivation
  - QOS time slots association
    ☛ Only allow QOS usage on specific time slot (like for reservation)

• **Heterogeneity management**
  - For job layouts
    ☛ Requesting multiple tasks with different resources requests per tasks
      ✔ 4 cores for the 2 first tasks, 2 cores for the others,...
  - For hardware resources allocation
    ☛ Requesting multiples nodes with different features on each
      ✔ 2 nodes with GPUs, 2 nodes with more memory, ...

• **Extended Job Accounting**
  - Add new fields in the accounting tables (generic resources, power consumption, ..)

• **Fairshare management**
  - Notion of time credit
    ☛ A user can use up to a certain amount of time and is blocked after that
  - Multiple time credit banks for different HW
    ☛ Users allowed to use up to 10K hours of basic nodes and up to 5K hours of GPU
    ☛ nodes/partitions mapped to specific time banks to automatically account execution time to the corresponding banks
Interesting topics

- **Preemption in suspend mode with no memory restriction**
  - Ensure on-demand access to the whole cluster if necessary
  - Currently restricted to jobs that fit available memory on nodes

- **Pruned hierarchical slurmdbdss**
  - Centralize users/limits/QOS/... definition on a single entity
  - Distribute accounting burden on clusters

- **Heterogeneous topologies support**
  - Unified way to manage compound topologies in SLURM

- **NUMA topology in intra-node resources selection**
  - BULL MESCA 16 sockets node will require it for best efficiency
Interesting topics

- **LDAP accounts sync automation**
  - Avoid in-house scripts, dynamic accounts addition/removal

- **Pool of spare nodes**
  - Automatically rerun canceled jobs due to node failures

- **Resources allocation tagging**
  - To let users describes which jobs can share resources by tag

- **Kerberos Authentication (not only kerberos support)**
  - Replace munge for enhanced security with untrusted hosts
Thank you for your attention

Questions ?