CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

MC jobs with SLURM at CSCS

April 15th 2014

Miguel Gila, CSCS

miguel.gila@cscs.ch

grid@cscs.ch

CSCS-LCG2

- **Swiss Tier-2 for ATLAS, CMS and LHCb**

- **Currently providing ~27kHS06**
  - ~2.7k jobs running at all times
  - ~1k jobs in queue avg. / seen max of ~4k

- **Running SLURM 2.6.2 since October 2013**

- **Multiple node configuration**
  - 10x AMD Interlagos (32 core) *– to be decommissioned*
  - 67x Intel Sandy Bridge (32 core)
  - 16x Intel Ivy Bridge (40 core)

  - HT enabled => one job slot per "core"
  - Using Infiniband QDR/FDR + shared filesystem (GPFS) => MPI possible (but not configured)

MC jobs

- Running ATLAS MC jobs since Jan 2014
  - First job started on 2014-01-01T18:35:40
- EMI-3 ARC and CREAM CEs configured, but MC jobs landing only on ARC
- We make no distinction between jobs: MC and SC run under the same users, partitions and conditions ➔ this makes fair-share calculations easier and more accurate
  - ATLAS, CMS and LHCb (+ops, hone, dteam) allowed to run MC jobs
- ATLAS is the only VO running MC
  - So far completed ~6800 ATLAS jobs (1.6% of ATLAS)
  - In terms of CPU time, MC jobs account for 259.066h (18.5% of ATLAS)

The middleware

- **Initial issues with the middleware:**
  - ARC-CE would reserve 8x cores per job (=64 instead of 8)
  - CREAM-CE would simply ignore MC jobs

- **Modified the submission scripts by hand for SLURM and MC** (thanks to ICM's initial port)
  - ARC: *submit-SLURM-job*
  - CREAM: *slurm_submit.sh* + *slurm_local_submit_attributes.sh*

- **Problems now solved,** anything **newer** than the following versions should support MC without issues:
  - emi-cream-ce-1.2.2-2.el6.noarch
  - nordugrid-arc-3.0.3-1.el6.x86_64

- **Currently ATLAS MC jobs only land on ARC-CE**

SLURM configuration

- **SLURM supports multicore jobs by default**

slurm.conf

```
SelectType=select/cons_res              # consumable resources
SelectTypeParameters=CR_CPU_Memory      # consumable resources are CPU and MEM
SchedulerType=sched/backfill            # backfill is enabled

MaxTasksPerNode=40                      # Max is 128tasks per node, but for WLCG we want ~one per CPU

NodeName=DEFAULT        RealMemory=64359  CPUs=32 State=UNKNOWN    # keep it simple: no Sockets, SocketsPerBoard…
NodeName=wn[80-95]      RealMemory=128894 CPUs=40 State=UNKNOWN    # just number of CPUs = job slots

PartitionName=DEFAULT Nodes=wn[01-48],wn50,wn[52-95] Default=YES Priority=10     DefMemPerCPU=2000 Shared=NO
PartitionName=atlas   Priority=10   MaxTime=96:00:00 AllowGroups=atlas,nordugrid MaxMemPerCPU=4000 Default=NO
```

- **One partition (queue) per VO** (atlas uses 2, atlas + atlashimem)
  – All nodes belong to all partitions

- **Testing is easy:**

```
$ srun -N1 --ntasks-per-node=2 -p other hostname
```

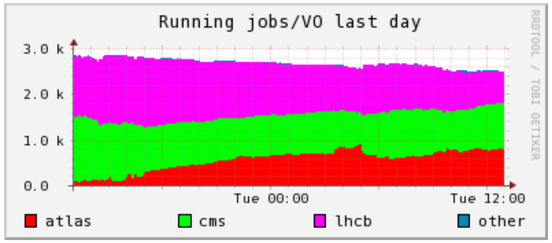
```
$ cat simple_script.jdl
[
SMPgranularity = 2;
#WholeNodes = True;
#HostNumber = 1;
#CpuNumber = 2;
Executable="simple_script.sh";
InputSandbox = {"simple_script.sh"};
StdOutput = "stdout.out";
StdError = "stderr.out";
OutputSandbox = {"stdout.out", "stderr.out"};
OutputSandboxBaseDestURI = "gsiftp://localhost";
]
```

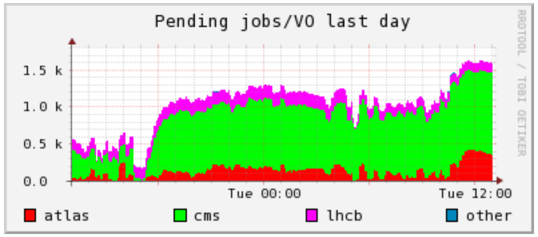
SLURM configuration

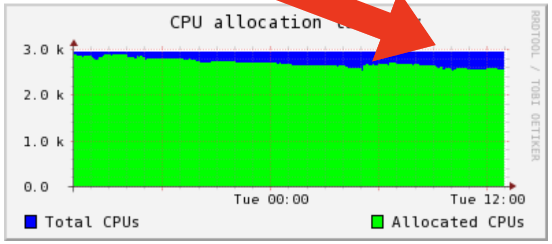
- **No implicit limit on the number of cores that can be requested by jobs.**
  - The CE needs to limit it to < # cores in a system
  - Nowadays 8 seems a safe assumption

- **Backfilling enabled,** but not very useful as jobs don't request time constraints ➔ they get the max configured and SLURM can't plan ahead
  - Avg. queue time ATLAS SC: 2h:32m:44s
  - Avg. queue time ATLAS MC: 11h:59m:39s

SLURM is draining nodes to make room for MC jobs ➔ node utilization goes down







Our evaluation

- **The good:**
  - Extremely easy to deploy: minimal or no changes on the scheduler configuration are required
  - Fair-share calculations take into account total number of cores per job
  - Middleware seems to be prepared for SLURM + MC jobs
- **The bad:**
  - Backfilling would work better if jobs would actually use time limits
  - Not a lot of scripts available to parse accounting DB ➔ a lot of work in-house to plot statistics (http://wiki.chipp.ch/twiki/bin/view/LCGTier2/PhoenixMonOverview)
- **The ugly:**
  - Some versions of SLURM have serious bugs (i.e. 2.6.2 would crash when reserving cores instead of full nodes ➔ ops is affected!!)

Questions?

Thank you for your attention.

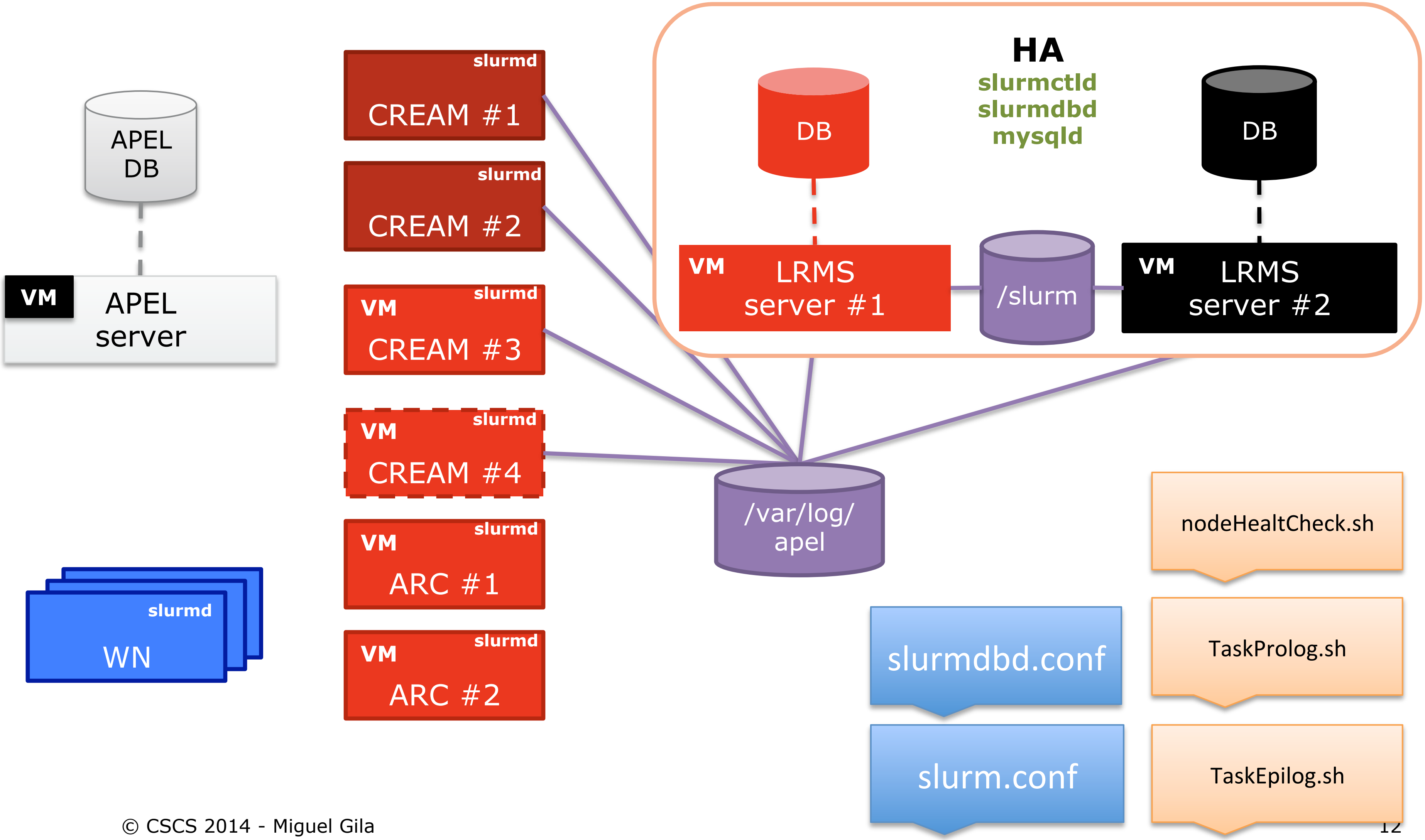
Backup slides

SLURM processes

- **slurmd:**
  - runs on the clients (WN, ARC and CREAM)
- **slurmctld:**
  - it is the scheduler itself
  - runs on the control nodes (can be HA)
- **slurmdbd:**
  - it connects slurmctld and the accounting DB
  - runs on any node (usually control nodes, can be HA)

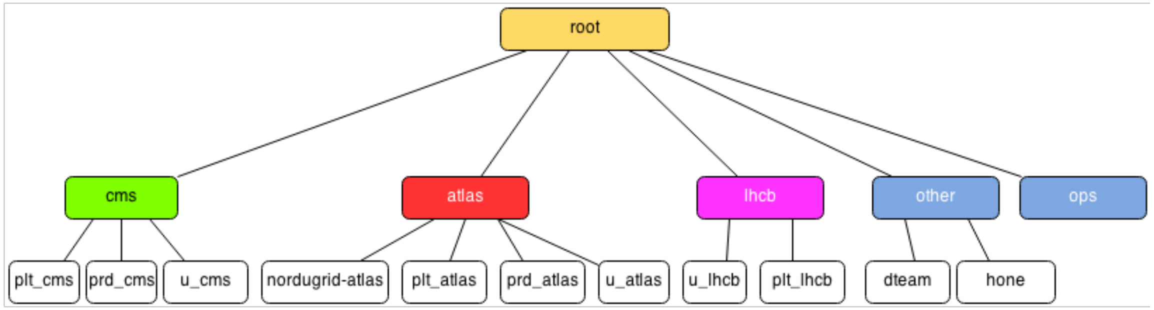
- **mysqld:**
  - runs anywhere (can be HA)

Our setup


APEL
DB
VM
APEL
server
slurmd
WN
slurmd
CREAM #1
slurmd
CREAM #2
slurmd
VM
CREAM #3
slurmd
VM
CREAM #4
slurmd
VM
ARC #1
slurmd
VM
ARC #2
HA
slurmctld
slurmdbd
mysqld
DB
DB
VM
LRMS
server #1
/slurm
VM
LRMS
server #2
/var/log/
apel
nodeHealtCheck.sh
TaskProlog.sh
TaskEpilog.sh
slurmdbd.conf
slurm.conf

Configuration details

- **7 partitions** (atlas, atlashimem, other, ops, lcgadmin, cms, lhcb)
- **All nodes are in all partitions/queues**
- **1 reservation for priority_jobs**
  - OPS + VO *sgm users
  - 2 nodes fully reserved (because of bug on slurm 2.6.2)
- TaskProlog.sh and TaskEpilog.sh empty
- nodeHealthCheck.sh runs on all nodes every 3 minutes and checks for basic system health. It drains the node if not all checks are successful
- Both SLURM control daemon nodes need to share /slurm for consistency
- Hierarchical accounting configuration



slurm.conf

```
ControlMachine=slurm1
BackupController=slurm2
[…]
SlurmdSpoolDir=/tmp/slurmd
TaskProlog=/etc/slurm/TaskProlog.sh
TaskEpilog=/etc/slurm/TaskEpilog.sh
AuthType=auth/munge
SchedulerType=sched/backfill
SelectType=select/cons_res
SelectTypeParameters=CR_CPU_Memory
TaskPlugin=task/none
ProctrackType=proctrack/linuxproc
DefaultStorageType=slurmdbd
AccountingStorageType=accounting_storage/slurmdbd
JobAcctGatherType=jobacct_gather/linux
JobCompType=jobcomp/script
JobCompLoc=/usr/share/apel/slurm_acc.sh
AccountingStorageEnforce=limits
HealthCheckInterval=180
HealthCheckProgram=/etc/slurm/nodeHealthCheck.sh
[…]
```

```
[…]
PriorityType=priority/multifactor
PriorityDecayHalfLife=07-12
PriorityFavorSmall=YES
PriorityMaxAge=4-0
PriorityWeightAge=1000
PriorityWeightFairshare=5000
PriorityWeightJobSize=1000
PriorityWeightPartition=10000
PriorityWeightQOS=1000
FastSchedule=1
PreemptType=preempt/none
[…]
```