UiO : Universitetet i Oslo

Batch systems: SLURM (@UiO)

Dmytro Karpenko (UiO/NeIC)
Heavily based on materials from Bjørn-Helge Mevik (UiO)

Slurm entities

* User

* Account

* Partition

* QoS

No queues!

Slurm jobs

* Job

* Job step

* Task

Slurm exclusively for ATLAS

Make a separate partition and you're good to go!

Map jobs to different user accounts and give them different priorities.

Slurm @ UiO

Heterogenous resources:

- 684 "standard" nodes (16 cores, 64 GiB RAM)
- 20 accelerator nodes (GPU or Xeon Phi)
- 8 hugemem nodes (32 cores, 1 TiB RAM)

Heterogenous projects:

- Some have a limit on #cpus, others a limit on #cpu hours
- A mix of inexperienced and expert users

Heterogenous load:

- A lot of single-cpu (or few cpus) jobs
- Some parallel jobs up to ~ 1000 cores
- Many jobs needing lots of RAM
- Some jobs needing whole nodes

Typically 1000-1500 jobs running, ~ 1000 pending

ATLAS @ Slurm @ UiO

- ATLAS jobs run into several accounts, but they are just ones from many other.
- Those accounts have limits.
- But those accounts never get penalized over usage!
- And they have slightly higher priorities.

UiO: Setup

- Hand out cores and RAM with ConsRes
- Job cpu and RAM limits: cgroups
- Project cpu and RAM limits: QoS limits
- Project cpu hour limits: Gold (via prolog/epilog)
- Fair usage: Fairshare priorities

Single-core and multicore

The nodes have features: n or c

ATLAS jobs that require 8+ cores are assigned to n-nodes

Others ATLAS jobs get assigned to c nodes.

Less stress for the backfiller/scheduler.

ATLAS jobs accounts

- ATLAS jobs mapped to different users according to proxies (this is ARC, not Slurm)
- Username defines in which account the job will run (also ARC)
- Accounts:
  - analysis/production – low priority
  - configuration – high priority
  - all other – normal priority
  - lowpri (through separate submitter, less important jobs)