Centre de Calcul
de l'Institut National de Physique Nucléaire
et de Physique des Particules

Local Batch System Migration @ CC-IN2P3

February 2023

Outline

- Migration
- Billing
- Energy Sobriety
- Infrastructure
- Context
- Alerting / Monitoring
- Work in progress

Local Cluster Migration

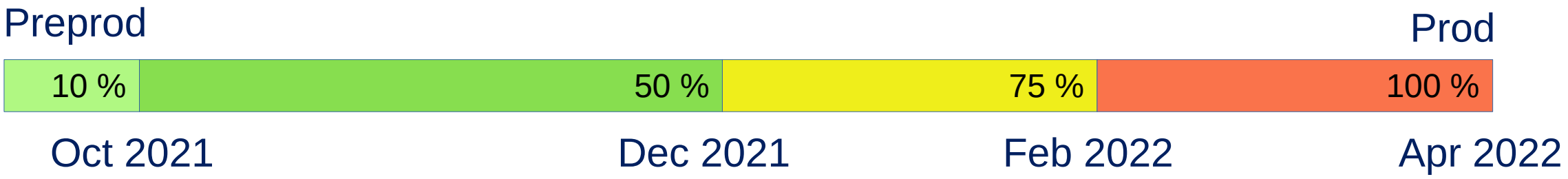
- Why leaving Grid Engine
  - Mostly for costs reason
  - Grid Engine license ran until 31th March 2022
  - Small bugs, and tickets with no answer...
- Why Slurm?
  - Slurm is free (optional chargeable support)
  - Slurm is well known and robust
  - There are similarities between Grid Engine and Slurm

    → quick adaptation for our users

Local Cluster Migration

First tests with Slurm: June 2021

Hardware resources in Slurm Cluster:



From UGE to Slurm

- Commands and logic are similar
- Slurm deals with real memory only (no virtual memory)
- GPU mangement is different
  - Slurm: GPUs are considered as a resource per se
  - UGE: GPUs are associated to CPUs
- Slurm has its own mechanisms
  - Jobs / Steps / Tasks
  - No queue, but partitions and QoS

Slurm HPC and HTC

- HPC context:
  - Resources are homogeneous
  - Things run faster when they’re not spread over whole cluster
  - Few groups and users
- Slurm was created for HPC, but works well in HTC context
- Some configuration tweaks to do

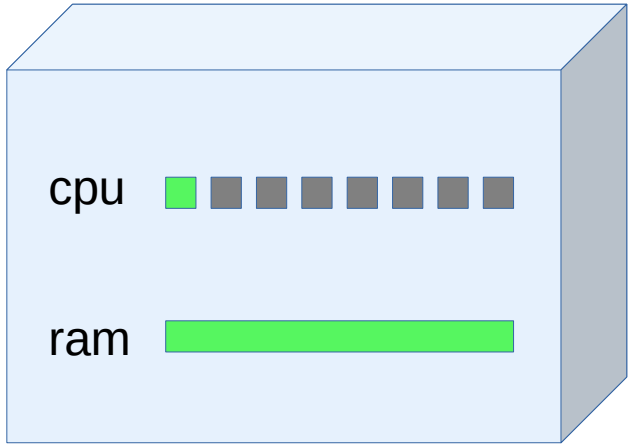
HTC settings for Slurm

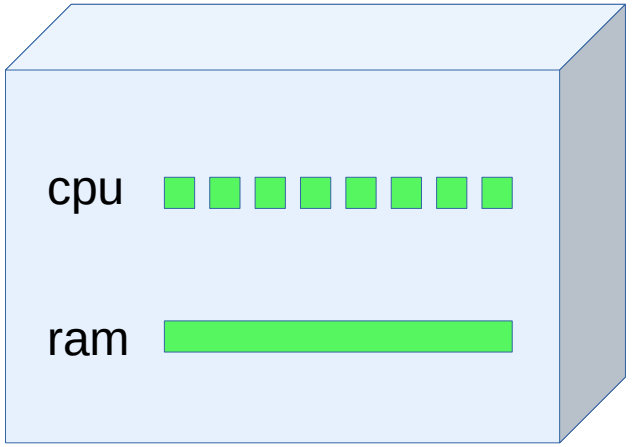
- Groups and users have to be created inside Slurm
  - Mapping needed between CC groups and users, and Slurm ones
- Add a running CPUs limit per group
  - Fairshare only is not sufficient when job fluctuation is important
  - CPU limits prevent user from taking too much resources
- Force time declaration at submission
  - Backfilling is difficult when all jobs have a big default time
- Use LLN (Least Loaded Node) option
  - Spreads job over whole cluster
  - Prevent nodes from overflowing

Local Cluster Billing

Until now, hs06 (cpu) was the only thing taken into account for billing

But what if a job requests huge amount of memory?





In both cases whole node is allocated but on one hand, bill is 1 cpu, and on the other hand, bill is 8 cpus. Is it fair?

Local Cluster Billing

- Slurm is able to apply a ratio between cpu and ram
- For example with a ratio of 1 CPU/3G of RAM:
  - If 2 cpus are requested, Slurm will allocate 2 CPUs and 6G of RAM
  - If 2 cpus are requested, Slurm will allocate 2 CPUs and 6G of RAM
  - If 1 cpu and 9G RAM are requested, Slurm will allocate 3 CPUs and 9G of RAM
- Advantage: billing with cpu is easier
- Disadvantages: user incomprehension and a lot of wasted resources

Local Cluster Billing

After 9 months of production, we now have data and statistics

We decided to remove ratio and to move billing a little further (not with Slurm)

Billing will be calculated with this fomula:

$$\max(\text{allocCPUs} \times \alpha_{cpu}, \text{allocMEM} \times \alpha_{mem}) \times T \times \text{HS06}$$

$$\alpha_{cpu} = 1$$

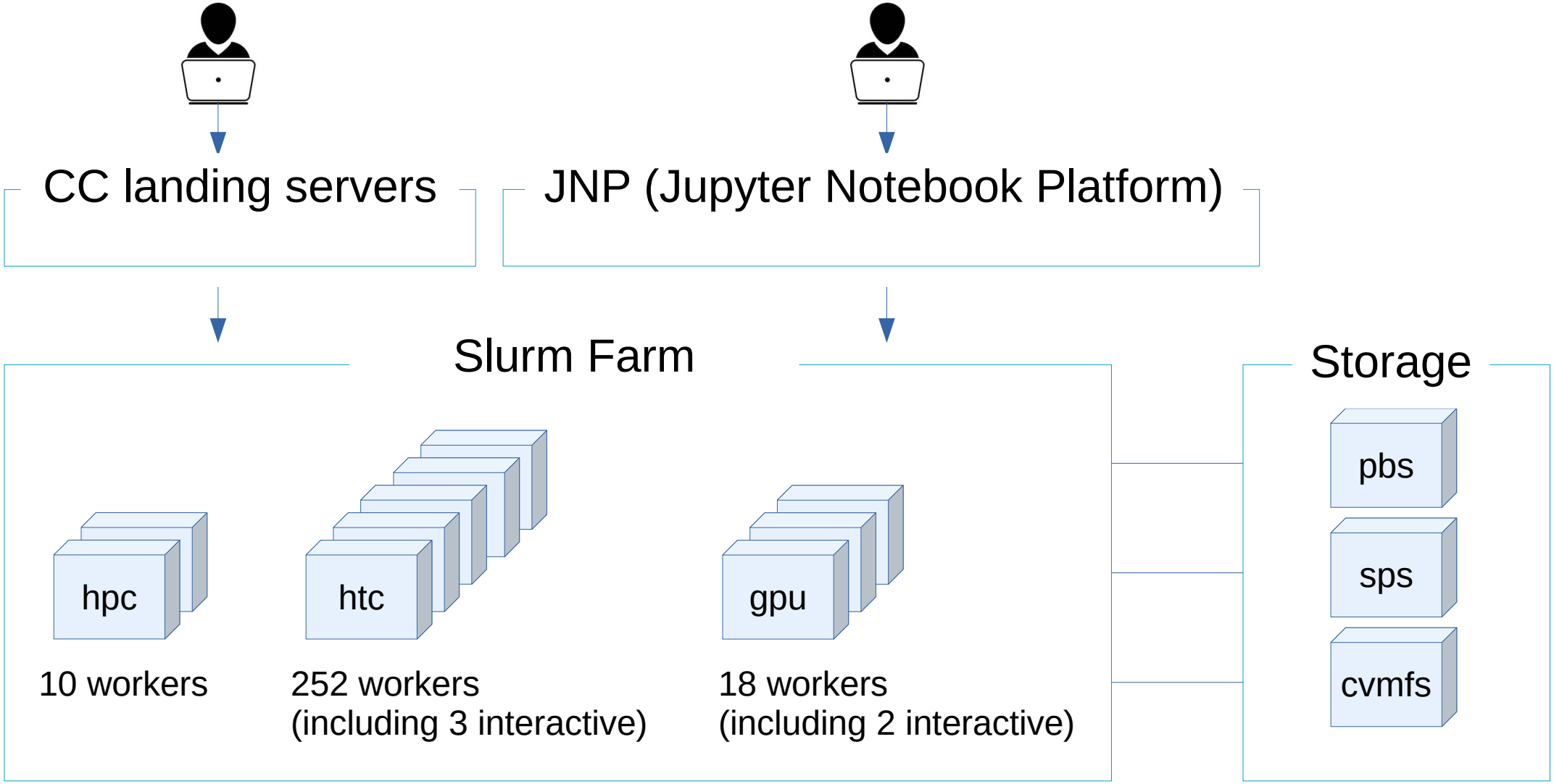
$$\alpha_{mem} = 1/3$$

Local Cluster Billing and Energy Savings

- Removing ratio will make jobs more efficient
  - Lots of resources savings…
  - So we are also removing 33% of the computing power

- Users have to declare CPU, RAM and time
- Efficiency tracking
  - The most efficient jobs are, the less energy we need for for same computational power
  - Statistics tools (BBQ)
  - We will try to track the most inefficient users

Local Cluster Infrastructure



Local Cluster Infrastructure

	Single-core/GPU	Multi-cores/ GPUs	Multi-nodes	Hardware	Workers
HTC	Yes	Yes	No	64 cores 192G RAM	290
HPC	Yes but no point	Yes	Yes	32 cores 128G RAM	16
GPU	Yes	Yes	No	4 V100 / 20 cores 192G RAM	18

- Total (2022) 21K cores / 72 V100 / 8 K80
- Total (2023) 16K cores / 72 V100

Local Cluster Context

- Groups: 150
- Users: 4500
- # HTC jobs / month: 1.5M
- # HPC jobs / month: 4000
- # GPU jobs / month: 8000
- Up to 12K simultaneous running jobs

Alerting / Monitoring

Alerting

- Nagios

Monitoring

- CC Sampler (home made web app based on grafana for time series visualization)
- BBQ (home made python web app for instant visualization)

Work in progress

- Submission through Grid (ARC CE)
- Fine tuning resource sharing (limits, fairshare, etc.)
- Explore REST Api
- Explore ways of saving energy
- Automation anywhere everywhere

Questions?

bertrand.rigaud@cc.in2p3.fr

guillaume.cochard@cc.in2p3.fr

Thank you for your attention.