Autonomic Approach to the Runtime Management of HPC Cluster Resources

Quentin GUILLOTEAU (CTRL-A & DATAMOVE),
Olivier RICHARD (DATAMOVE), Eric RUTTEN (CTRL-A)

Friday 29th April, 2022

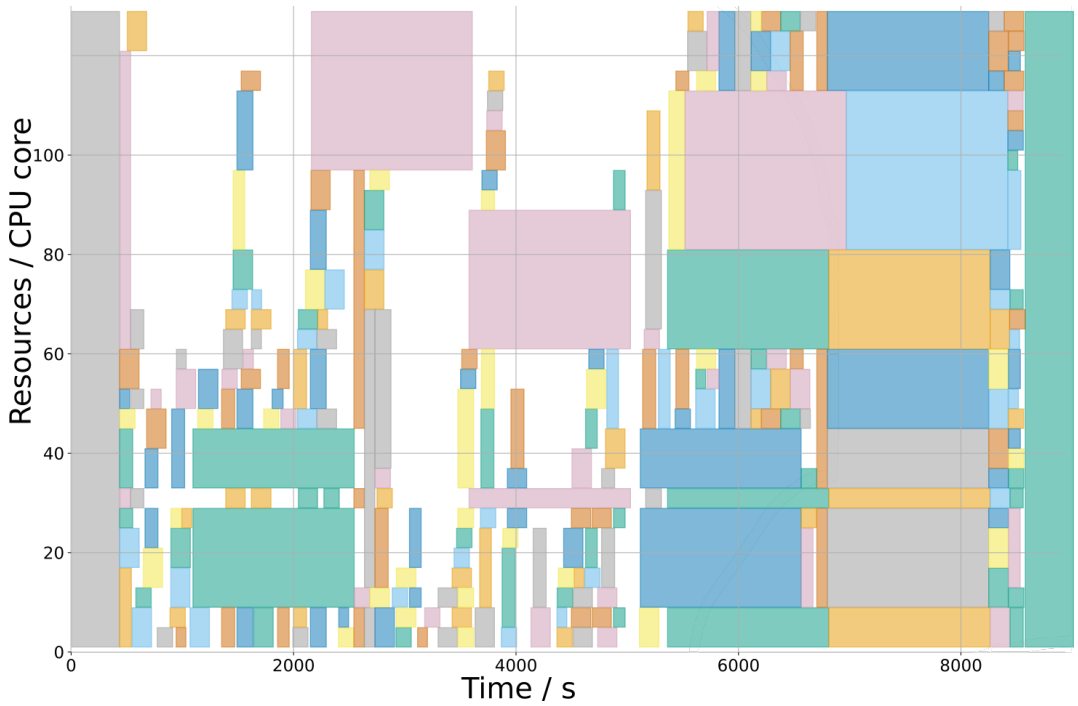
Is the Jar Full?

Jar: HPC Cluster

- Resources (CPU cores)
- Time (seconds)

Stones: HPC Jobs

- Static resource allocation
- Static time allocation



Idle Resources = Lost Computing Power

Sand: Bag-of-Tasks Applications

- Monte-Carlo, Parameter Sweep, etc: Lot of Small Indep. Jobs
- Lowest priority (Best-Effort): 2^{nd} Class Citizens (can be killed)

The Problems

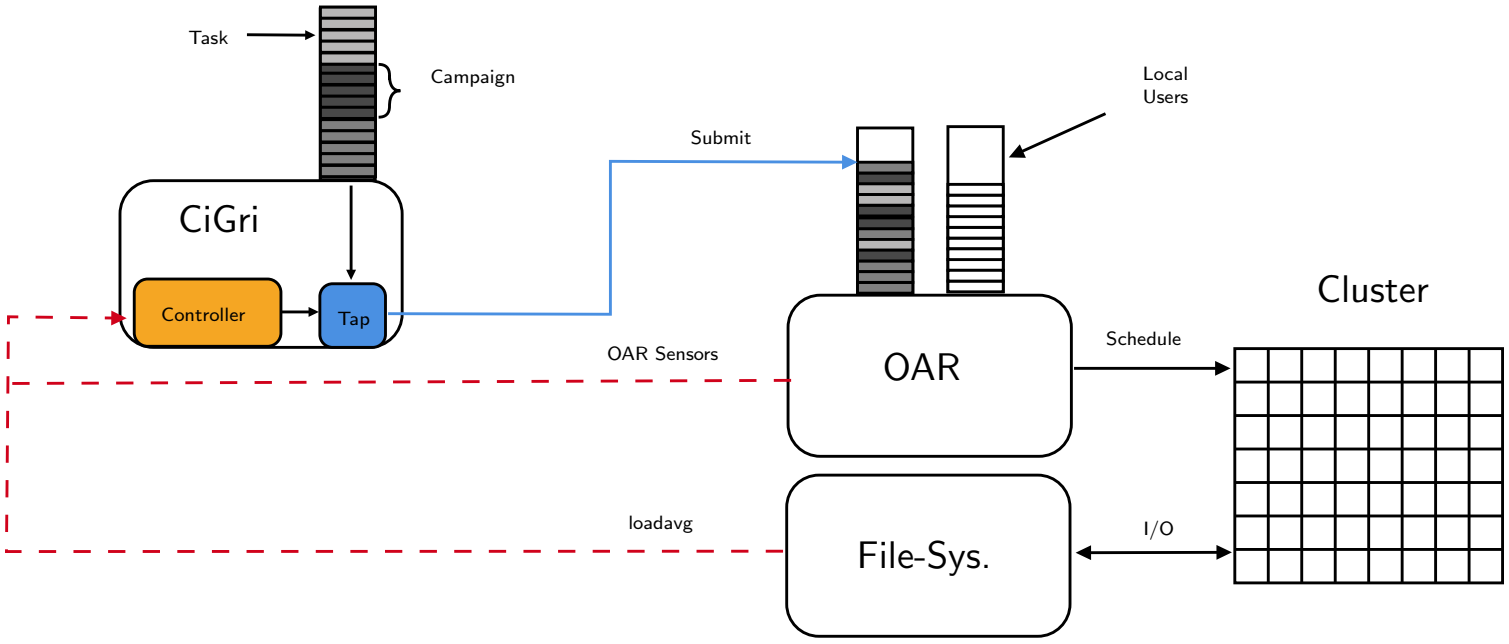
- Small jobs (*Sand*) can **perturb** the premium users (*Stones*)
  $\hookrightarrow$ Also use shared resources: Filesystem, Network, etc
- Great **Variability/Unpredictability** in job size, I/O, etc
  $\hookrightarrow$ Needs **online measurement** of the system to take decisions

My goal

Dynamically submit BoT jobs to **harvest the idle resources** while **ensuring QoS** for premium users.

How?

Use **Control Theory**[1] to regulate the BoT submissions based on **current load** of the cluster.



Example: Avoid the collapse of a Filesystem due to too many I/O

[1] Collaboration with Gipsa-lab