Fair Share Scheduler for OpenNebula implementation and performance tests

CHEP 2018 Sofia, Bulgaria

INDIGO - DataCloud

INFN Istituto Nazionale di Fisica Nucleare SEZIONE DI TORINO

Stefano Bagnasco, Sara Vallero and Valentina Zaccolo

Why scheduling fairly?

Large public cloud (Public)

- approximately infinite resources
- tenants are charged *a posteriori*

➢ applications scale in/out freely

Small scientific datacentre (Private)

- saturated regime
- tenants are charged *a priori*

➢ advanced resource allocation needed

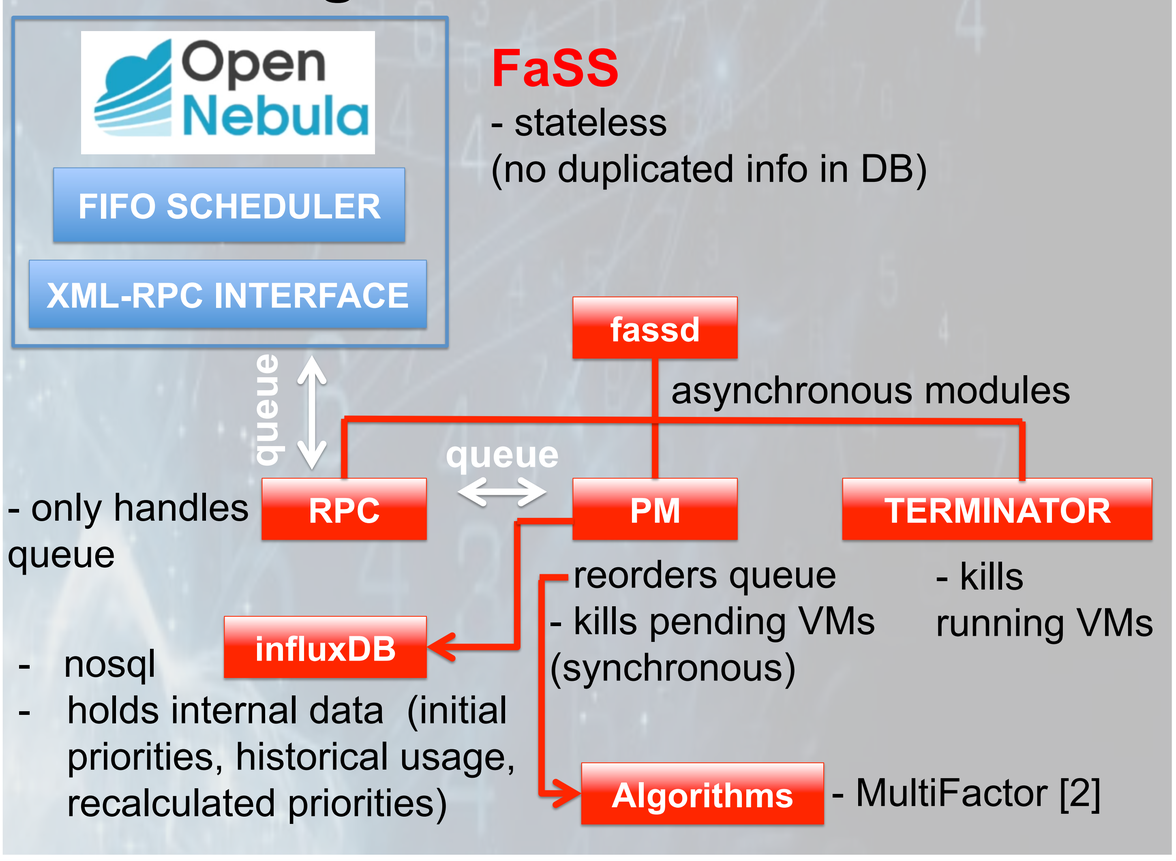
Default OpenNebula (ONE) scheduler is **FIFO** and based only on **static resources partitioning** among the projects → not suitable for scientific DC

FairShare Scheduling (FaSS) service [1]

Task priorities assigned according to
- ➢ initial weight
- ➢ historical resource usage

- **v1.0** released with ElectricIndigo in April 2017 with ONE patch
- **from v1.2** ONE 5.4 compatibility

High-level architecture



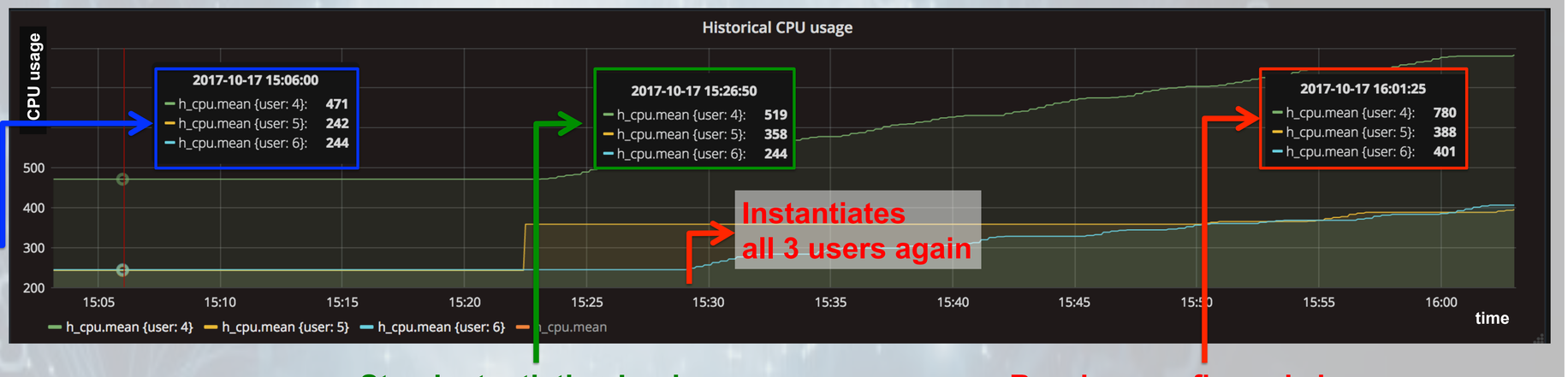
FaSS
- stateless
(no duplicated info in DB)

Performances

Test with three users, initial shares configured in /etc/fass/shares.conf

Huey	Dewey	Louie
uid=4	uid=5	uid=6
gid=1	gid=1	gid=1
share=50	share=25	share=25

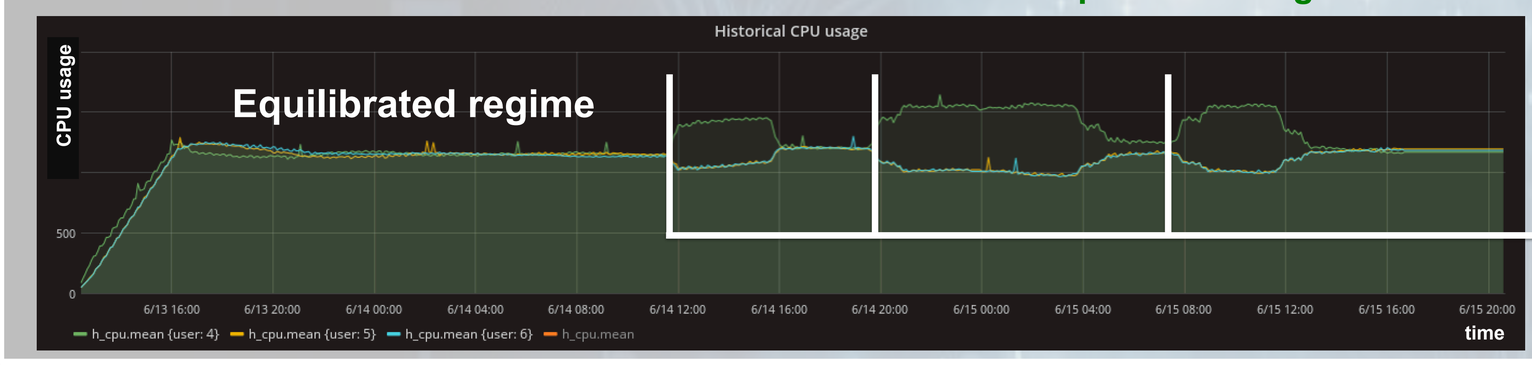
Periodically instantiating VMs for all 3 users



Stop instantiating Louie

Reaches configured shares

Stress test



- ➢ Run for 1 week, ~22k VMs
- stable performance

- ➢ Adding periodical stress sources (0 – 300 VMs to random users)
- favours Huey
- after stress → goes back to equilibrium

Installation

1. Prerequisites:
   - install ONE (versions above 5.4 operate with FaSS above v1.2, previous versions of ONE need FaSS v1.1 or before)
   - install InfluxDB and create fassdb
2. Install FaSS as root user:
   yum localinstall one-fass-service-v1.4-1.4.x86_64.rpm
3. Adjust the configuration file of the ONE scheduler, to allow it to point at the FaSS endpoint in /etc/one/sched.conf

Usage

1. Edit the initial shares for every user in /tmp/one-fass/etc/shares.conf
2. Start FaSS: systemctl start fass

Additional features

Keep your Cloud infrastructure clean with FaSS [3]
- Set VMs to be dynamic and to be terminated after a specific Time-to-Live (TTL)
- Instead of terminating, VMs can be powered-off, suspended or rebooted
- TTL settable per user.

Outlook

- Testing new algorithms
- Integrating into the production infrastructure

Stay tuned!

References:
[1] https://github.com/indigo-dc/one-fass
[2] https://slurm.schedmd.com/priority_multifactor.html
[3] https://opennebula.org/fass-fair-share-scheduler-for-opennebula/

INDIGO - DataCloud receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement RIA 653549