INDIGO - DataCloud
Better Software for Better Science

A FairShare Scheduling Service for OpenNebula

In a large Public Cloud, applications can scale out with only cost driven limitations. Whereas a small Scientific Datacenter normally works in a saturated regime and tenants are charged a priori by paying for a fraction of the total assets. In this case, an advanced resource allocation policy is required in order to optimise the datacenter occupation efficiency.

FUNCTIONAL REQUIREMENTS

- provide a **resource partitioning framework** to handle unallocated (dynamic) resources
- guarantee the coexistence of the dynamic and static partitioning models
- for dynamic resources, provide an allocation mechanism based on a **fair-share algorithm**
- define a new kind of **dynamic quota**
- provide a **queuing mechanism** for handling the requests that cannot be immediately fulfilled
- possibly apply the fair-share mechanism also to non standard resources (e.g. GPUs)

DESIGN

- not intrusive in the OpenNebula (ONE) core
- **self-contained module** interacting only with the ONE XML-RPC interface
- keep the **original scheduler** implementation for matching resources to requests
- **order the requests queue** according to priorities
- do **not add new states** to the VM life-cycle
- **synergy with the ONE** authentication, quotas, monitoring and accounting systems

A fraction of the available resources is statically assigned and partitioned among projects. Additional assets are partitioned dynamically following the effective request per project.
Efficient and fair access to dynamic resources must be granted to all projects. To this purpose, the FairShare scheduling Service implements an **algorithm** which **prioritises tasks according to** an **initial weight** and to the **historical resource usage** per project.

HIGH LEVEL ARCHITECTURE

Priority Manager:

- periodically calculates priorities for queued requests
- uses a set of pluggable algorithms

Database:

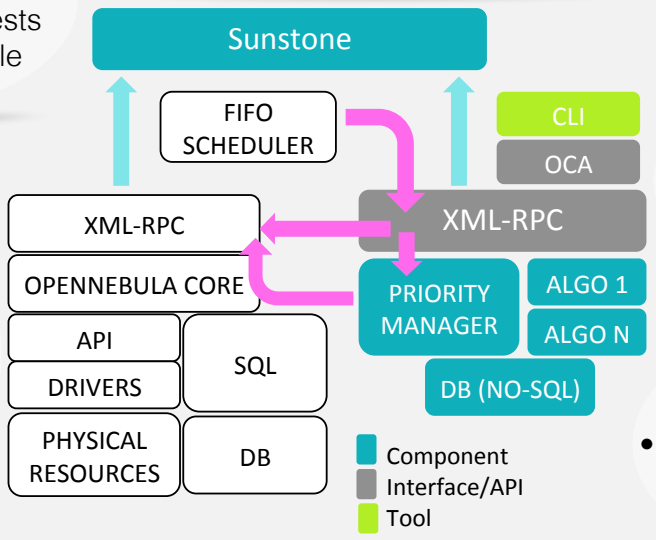
- noSQL (store large XML strings)
- holds the module's internal data:
  - initial priority values
  - historical information on resource usage
  - recalculated priority values
  - …

Sunstone:

- extend the original GUI to monitor and operate the new service

Algorithms:

- MultiFactor
- FairTree
- …



White boxes are the ONE original components.

Client:

- interface to the service core
- uses a set of bindings analogous to the ONE Cloud API (OCA)

XML-RPC server:

- catch scheduler calls
- provide re-ordered queue
- redirect unknown methods to the core ONE XML-RPC server

Contact: Vallero S. (sara.vallero@to.infn.it) and Bagnasco S. (stefano.bagnasco@to.infn.it) - **INFN Torino**

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