A FairShare Scheduling Service for OpenNebula

Vallero Sara - Bagnasco Stefano

CHEP 2016 - October 10/14 - San Francisco (CA)
Requirements

**FUNCTIONAL**

- 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
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

**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