

JINR experience and some thoughts on joining resources simplification

Mikalai KUTOUSKI

Laboratory of Information Technologies
Joint Institute for Nuclear Research
Dubna, Moscow region, Russia

Ways of joining resources simplification (1/2)

- 1) Provide “smarter”/simpler tools for grid services installation and configuration (e.g. “wizard”-style like “Next->Next->...->Done”)
 - for example, to simplify StoRM SE deployment in the framework of collaboration in BESIII experiment a special tool/script was developed
 - provide pre-deployed services in VM/CT images
 - distribute grid services as [snappy](#) apps

Ways of joining resources simplification (2/2)

2) Minimize a set of services need to be deployed on each site

- host most of grid services including CE at just few places (i.e. at organizations what have enough expertise in middleware deployment, hardware, electricity, networks bandwidth, etc)
- other sites should just host computing worker nodes (unclear in case of storage)
- computing worker nodes can be dynamically deployed in the local private clouds (e.g. the same way as JINR and BITP cloud were integrated into WLCG infrastructure)

Two more options for 2nd way

- (1) Integrate each local private clouds into WLCG but open questions are:
- Will it remind at end the current WLCG with 200 grid sites?
 - What is the difference between EGI Federated Cloud and that model?
- (2) Integrate local private clouds (e.g. on the national basis?) with each other following the model “cloud bursting”
- a driver to integrate OpenNebula and OpenStack-based clouds with each other using OCCl protocol.
 - OpenNebula-OpenStack scenario
 - Limitation: no real time resources monitoring because of incomplete OCCl standard implementation
 - OpenNebula-OpenNebula scenario
 - No such limitation mentioned above(worked around by using native OpenNebula XML-RPC)
 - In such “cloud bursting” model one can join a set of local private clouds (e.g. on the national basis similar to NGIs?) into one national cloud and integrate just that “joint” cloud into WLCG (hence reduce number of entities/endpoints to interact with).

Challenge

- One needs to evaluate what is simpler to deploy and maintain:
 - Grid site with a set (even minimal) of grid services
 - Local private cloud

Third way

3) Outsource grid services/cloud deployment and maintenance:

- Local admin just performs «low»-level operations (mounts hardware and supply its remote accessibility via IPMI)
- Remote expert does the rest: deploy grid services/cloud

Combinations

- 1st and 2nd way:
 - A tool like a OpenStack Fuel (it simplifies OpenStack-based clouds deployment)
 - implement OpenStack local private clouds deployment using Fuel with some typical predefined profiles/configurations
- As soon as full control over server via IPMI will be implemented in the Fuel the 3rd way will become even simpler: remote admin (i.e. 3rd way) will be able to deploy private cloud (2nd way) using simple tool (1st way).

References

- Snappy Ubuntu Core:
<https://developer.ubuntu.com/en/snappy/#tour>
- OpenStack Ironic (bare-metal provisioning):
<https://wiki.openstack.org/wiki/Ironic>