

Attendees:

Local: Stefan, Ulrich, Maarten, Miguel,

Remote: Manfred, Jan Just,

Apologies: Tim, Dirk, Victor

Action Items

===

- Plan for classical batch systems
 - Finish up on mjf.py tool and create a rpm for easy deployment on sites
 - > first implementation done, packaging is pending
 - Invite sites who currently have machine features installed to also install above rpm
 - > tested successfully at GRIDKA
 - Finish up on a SAM probe to retrieve and test machine features and let it run through WLCG nagios (e.g. some non-critical profile)
 - > SAM probe is still under development, especially moving to the "mjf.py" tool / interface needs to be done
 - Collect the information from SAM probes on the "test sites" and display it on a separate web page
 - > depends on SAM probe to be finished
 - Check the gathered information and gain confidence in its validity
 - > depends on SAM probe
 - Invite VOs to also use the information provided on "test sites"
 - > not yet started, the idea is to have e.g. CERN and T1 sites as a playground for both batch and virtualised infrastructures and via a fast development cycle provide fixes
 - Ask for wider deployment of machine features on physical batch systems WLCG wide
 - > pending
 - Understand the power number provided on virtualized WNs
 - > not yet started, Manfred: Hepix working group understands how to provide sensible information, this needs to be also broadcasted to sites which are providing virtualized WNs and also to IaaS service providers
 - Find out which IaaS systems are currently being tested / used within WLCG
 - > On an email conversation on this topic Victor provided an EGI twiki which shows that the major IaaS systems at the moment used in production and validation are openstack and opennebula. We shall concentrate on those first. see also <https://wiki.egi.eu/wiki/Fedcloud-tf:Testbed#Endpoints> and <https://wiki.egi.eu/wiki/Fedcloud-tf:ResourceProviders>
 - Get in contact with Condor team to ask about a development of providing machine/job features information
 - > Dirk was in contact with the Condor team who confirms its not a big problem to implement the features. Some parts of features don't have a one to one equivalent but sensible values can be provided. It was suggested to make an official request to the condor team on this topic.

Status of physical batch systems

===

- 'mjf.py' was successfully tested at GRIDKA by Manfred
- Interface enhancement for mjf.py, "give me the remaining time (real time seconds) I can run", i.e. minimum of shutdown time and remaining time within the queue, this info shall also be provided for virtual environments -> ACTION
- next step is writing the spec file for packaging and uploading the rpm to the WLCG repository at CERN

Status of virtual infrastructures

===

- Started with testing mjf.py on the CERN/openstack infrastructure. ie. we can "generate" machine features through the nova client as already tested by Tim (see also <https://twiki.cern.ch/twiki/bin/view/AgileInfrastructure/SchedulingMetadata>). The test was redone generating machine features values for a test machine and "mjf.py" was enhanced to consume the information via the magic IP and return it through the same interface.
- Meeting with Ulrich and Tim later today to discuss on how to provide the information in an automatic way on the openstack service side.
- In contact with PIC who is running opennebula and start extending machine features also on this platform
- In order not to overload the service side we shall have a cache implemented with "mjf.py", ie. within a specific time range only the cached values are returned, only after the time has expired the information will be updated. The service may provide a caching time otherwise we fall back onto a hardcoded time within mjf.py -> ACTION

Aob

===

- Decided to setup a central repository which will host
 - the client implementation "mjf.py"
 - implementations of the different service flavors for real batch systems and IaaS systems, unless the implementations will be provided within the upstream projects (e.g. condor could be an example).

Currently mjf.py is hosted in github.com, we'll check whether to leave the repository there or to move to a central CERN service.

- Ask all implementations of services already existing to be committed unless they are provided within the concerned system
- The next meeting will be called on the mailing list (it will be again a Tuesday 2pm in 28-r-15)