UNIVERSITY OF BIRMINGHAM

Advanced and Future Topics

HEPSYSMAN Workshop, 14th January, 2016

Mark Slater, Birmingham University

Ganga Service

The GangaService was developed to fill more production roles with Ganga or when long term submissions, etc. need to be run

It behaves as you would expect a service to behave. It can:

- Run Ganga as a daemon
- Access a running Ganga instance using a simple API

A typical use of the GangaService is shown:

```
from GangaService.Lib.ServiceAPI.ServiceAPI import
GangaService

gs = GangaService()
gs.gangadir = "/home/mws/gangadir-server"
gs.prerun = "source ~/dirac_ui/bashrc"
gs.gangacmd = "/cvmfs/ganga.cern.ch/runGanga.sh"

print gs.sendCmd("""
j = Job()
j.submit()
""")

gs.killServer()
```

Import the API

Setup how you want the Ganga service to run

Send the commands as if you were in IPython. The returned string is stoud/err

Plugin Development

One of the main advantages with Ganga is that you can relatively easily develop plugins to handle the specifics of certain elements of your use case

Any plugin type can be added, however the most useful are:

Applications

These usually provide a set of parameters that map to the specific program to be run and perform some checks to make sure they are sensible

Splitters

These can be dependant on how you want to split up a master job into smaller subjobs, e.g. by input files or parameters and can therefore provide a simpler interface than the generic splitter provided

There wasn't really time to cover this in the workshop but a step-by-step guide will added to the Tutorial very soon!

Future Credential System

The current credential system is having a compete overhaul in the near future to provide:

- Multiple local and backend credentials handled (not just grid and AFS)
- Easy runtime management of required credentials
- Credential dependent submission/monitoring

This is in beta and should be available in the next couple of months

```
# examine startup creds
for c in CredentialStore.getCredentials():
     print c.identity()
     print c.timeleft()

# create a new credential
c = VomsProxy( role = "..", proxy_path = ".." )
c.renew()


# submit job with a particular cred
j = Job()
j.credentials = [ AFSToken(), c ]
j.submit()


# Define default credentials in .gangarc:
DefaultCredentials = { {'AFSToken' : {'Backend':'LCG','Application':'Athena'},
 {'UID/Type' : {'Backend':'Westgrid', 'Application':'Athena'} }
```

Pure Python Usage

In the next 6 months, we are hoping to provide a seamless python interface to Ganga, i.e. allow you to use Ganga and access all your jobs, etc. from a plain python session

In doing this, we will also greatly speed up Ganga startup time and make the whole code base significantly more efficient!

In addition, we will soon have Ganga available on PyPI and so you will be able to easily install Ganga by simply doing:

Useful Links

The main Ganga web page is still hosted at CERN:

https://ganga.web.cern.ch/ganga/

The Ganga github page hosts the project including the Tutorial Wiki and Issue tracker:

https://github.com/ganga-devs/ganga

As well as directly posting new issues here you can get help at:

project-ganga<AT>cern.ch

OR

project-ganga-developers<AT>cern.ch

Or emailing me: mws@hep.ph.bham.ac.uk