Expanding the user base beyond HEP for the Ganga distributed analysis user interface.

Robert Currie[1], Ulrik Egede[1], Alexander Richards[1], Mark Slater[2], Mark Smith[1], Matthew Williams[2]

1) Imperial College, 2) Birmingham University

Outline

Introduction / What is Ganga?

"Ganga is an easy-to-use frontend for job definition and management." github.com/ganga-devs/ganga

Reasons to use Ganga over doing things manually:

- Local storage of job management and configuration
- Automated file transfers
- Configure once run anywhere
- Support for multiple backends not just DIRAC
- There are features to create, progress and manage thousands of jobs automatically

Recent changes within Ganga

Ganga has gone through some large changes recently, many of which make it more appealing to smaller VOs.

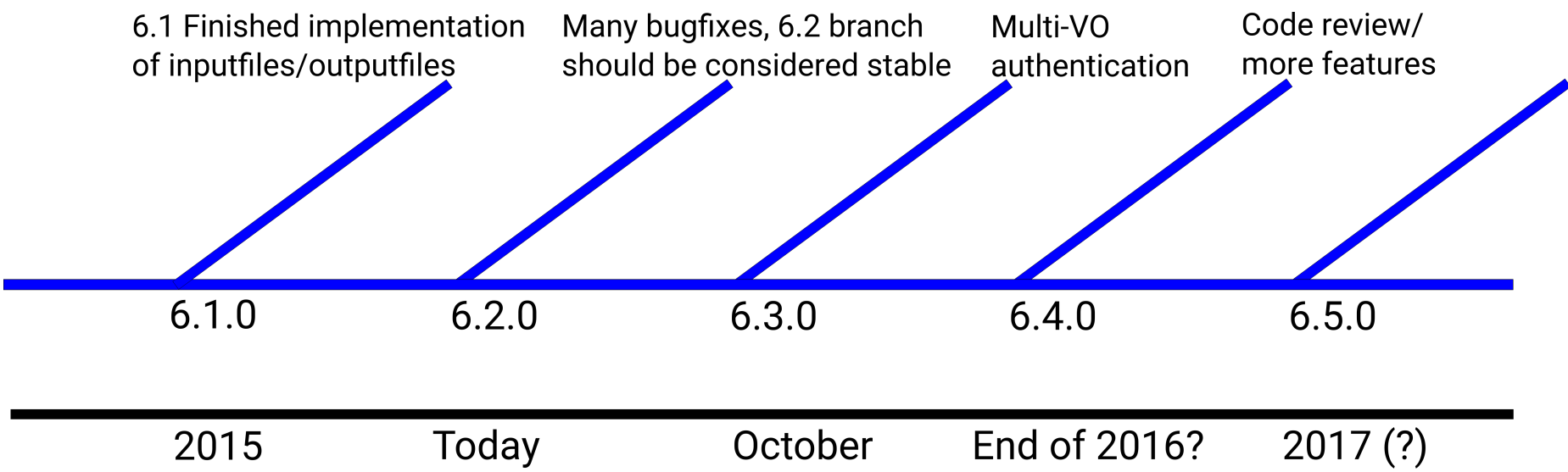
Some of the main changes are:

- Ganga is now available via “pip install” or can be run directly from CVMFS on CERN-VM.
- The out of the box configuration of Ganga is much friendlier to new users.
- Integrated testing is working to keep Ganga stable and reliable.
- Starting with Ganga 6.2.0 we’re moving to a more reliable release strategy.

Status of Ganga / Release Roadmap

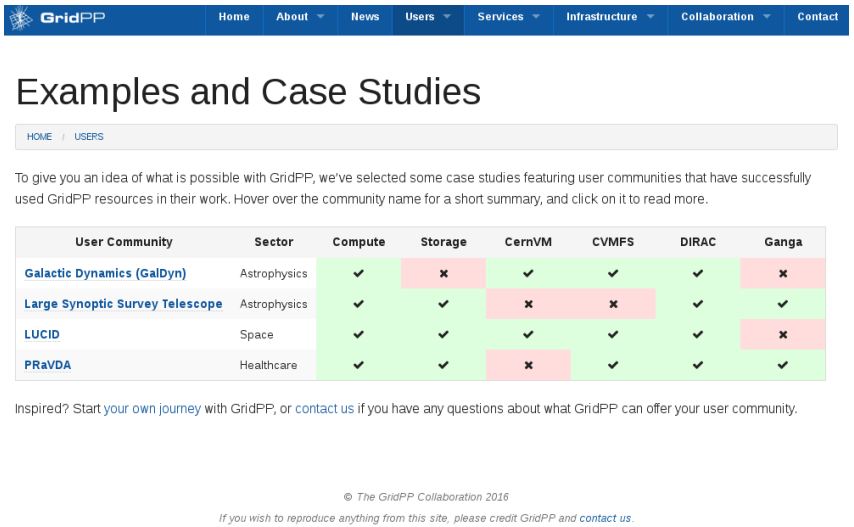
Ganga 6.2.0 has recently been released.

We consider this the new “gold standard” release and would encourage all users to update to this asap.



**This roadmap is not set in stone and may fluctuate/evolve.*

Working with smaller VOs



GridPP | Home | About | News | Users | Services | Infrastructure | Collaboration | Contact

Examples and Case Studies

HOME / USERS

To give you an idea of what is possible with GridPP, we've selected some case studies featuring user communities that have successfully used GridPP resources in their work. Hover over the community name for a short summary, and click on it to read more.

User Community	Sector	Compute	Storage	CernVM	CVMFS	DIRAC	Ganga
Galactic Dynamics (GalDyn)	Astrophysics	✔	✖	✔	✔	✔	✖
Large Synoptic Survey Telescope	Astrophysics	✔	✔	✖	✖	✔	✔
LUCID	Space	✔	✔	✔	✔	✔	✖
PRaVDA	Healthcare	✔	✔	✖	✔	✔	✔

Inspired? Start your own journey with GridPP, or contact us if you have any questions about what GridPP can offer your user community.

Case Studies https://www.gridpp.ac.uk/users/case-studies/ and working with other smaller groups.

Our work with smaller VOs has been to make resources available to them through the GridPP DIRAC instance hosted at Imperial College.

Ganga with small VOs

Supporting a smaller VO differs to supporting a larger one.

- **Larger VOs**
  Larger VOs can require the use of a some complex backend with many settings for managing data/jobs.
  Complex jobs often require many different things to be correctly configured during job configuration stage.

- **Smaller VOs**
  Jobs tend to require less configuration.
  Job often requires packing of complex environment and making it available to the WN.

First Impressions of smaller VOs

Groups working in smaller VOs tend to be fairly apprehensive about what is involved in moving analyses to the grid.

Experience with distributed computing comes from configuring jobs to run on just one site which has a well defined setup.

Often start with the simplest example, sending a short bash script and a tarball to a WN and running an executable.

Many potential avenues for improving use of available resources.

Working with smaller VOs

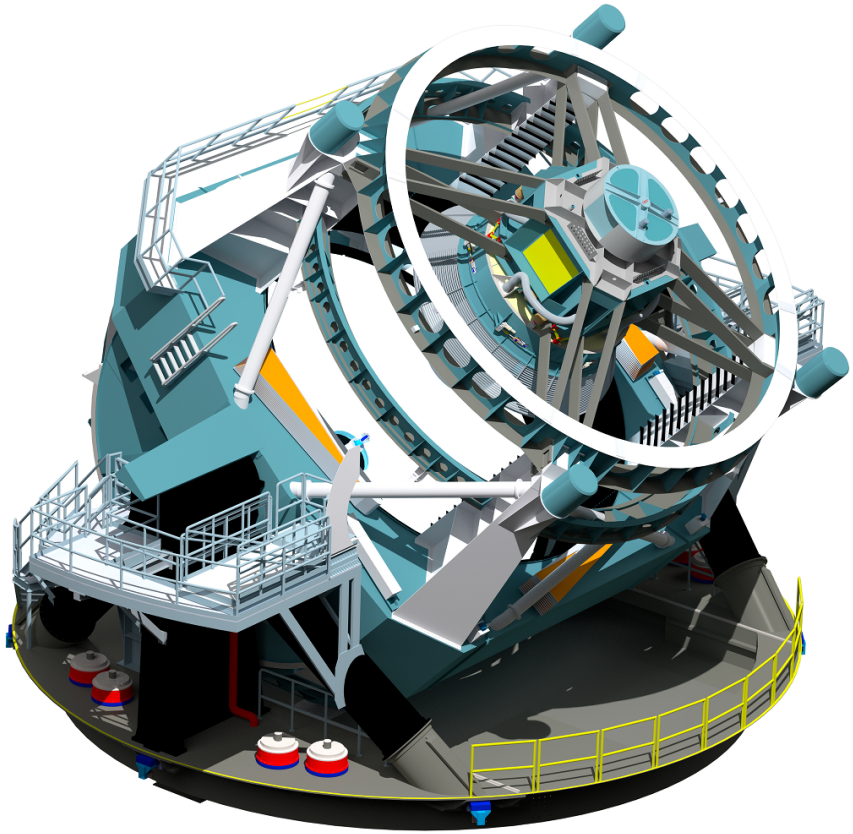
Working with smaller VOs has been improved through moving the main project to Github.

We collect some small amount of metadata on Ganga sessions and jobs submitted (nothing scary).

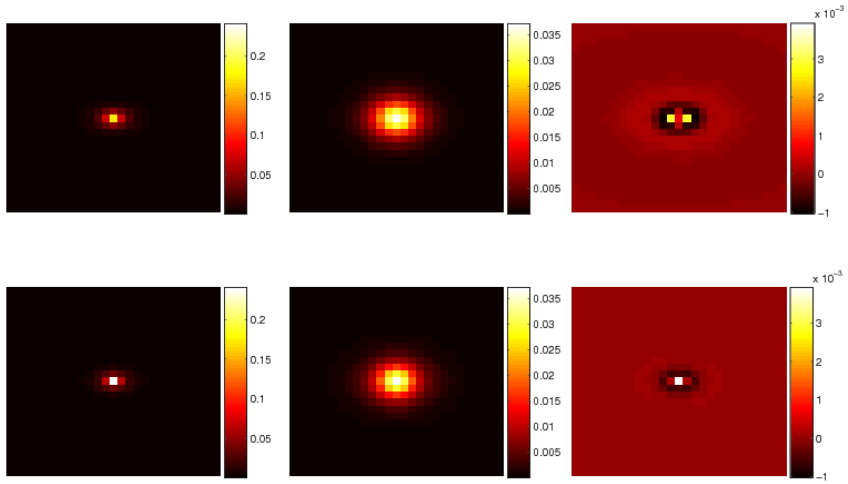
Compared to 2014/2015 monitoring data indicates in 2016 an increase in the number of jobs run by non-CERN groups using Ganga.

The UK appears to be a growing fraction of non-CERN use which suggests we're starting to see some new users.

Working with LSST



Telescope



Im3Shape Application

arxiv.org/abs/1302.0183

LSST workflow

Typical LSST workflow is very similar to other workflows in HEP.

Many files need to be processed requiring many CPUs which results in many more output files.

Problems faced by LSST which may be common to other users:

- How to distribute large software packages that are still potentially in development?
- How to manage data on a users job compared to a production jobs?
- How to regularly track large numbers of jobs using Ganga client-side.

lm3Shape on the grid

lm3Shape application used as an example workflow.
https://arxiv.org/abs/1302.0183

Full environment for the lm3Shape application is approximately 500Mb.
The best solution for distributing this was found to be uploading the file to DIRAC storage and downloading it to every worker node that needs to be run on.

Originally explored distributing lm3Shape via CVMFS but this was decided against. This may likely be re-considered again in the future.

Data usage

Input data for Im3Shape is typically between 1 and 3 data files each about 500Mb in size for each job with each job accessing only a small fraction of the overall data in each file.

First set of jobs run with a 1 to 1 mapping between data files and jobs but eventually will run with a 3 to 1 mapping.

Some experiences to note:

- Managing user vs production output data sometimes have different requirements, naming schemes etc.
- Post-processing of data is often required.
  (this needs development)
- Data transfers currently all managed within GridPP DIRAC.

Running on the WN

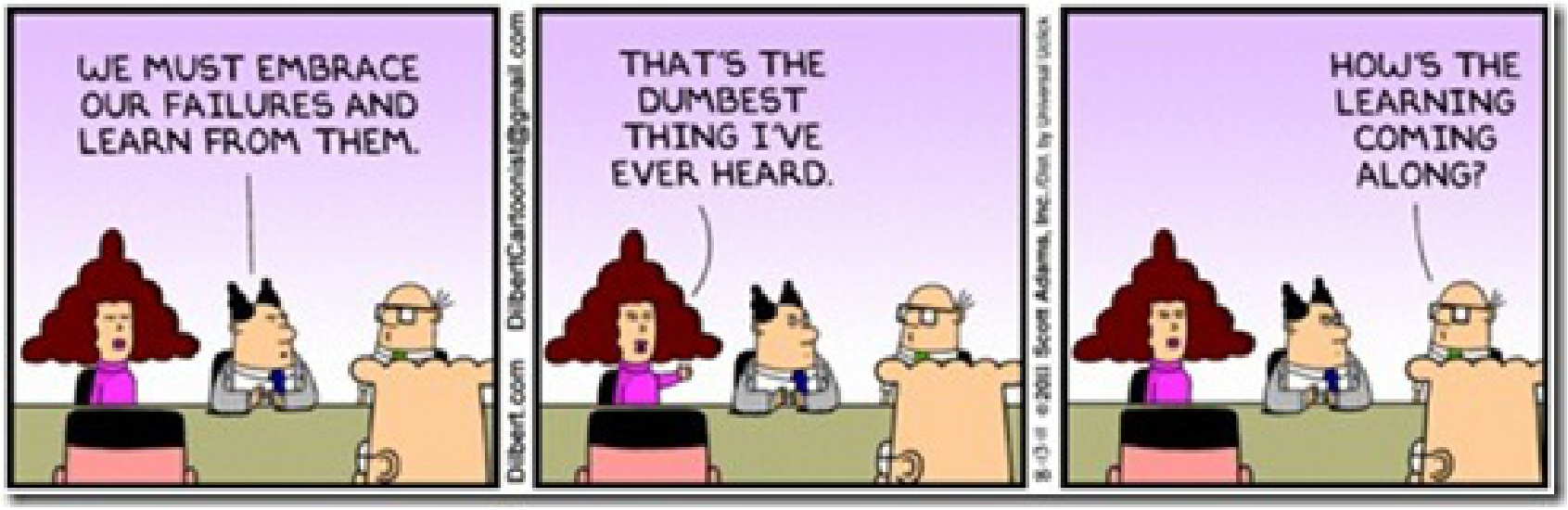
Jobs are constructed to request that DIRAC perform all file transfers to WN ahead of payload execution.

A short Python script auto-generated by Ganga is sent to each WN which handles extracting lm3Shape and dealing with input and output files for each job.

Intention is to run the same job script on the DIRAC WN as when running any jobs locally.

All of this scales dramatically when LSST starts taking data. For now development is using data from previous astro experiments.

Lessons Learned



Experiences from small VOs and new Users

Working with new users and smaller VOs has exposed the Ganga project to new ways of performing similar tasks.

Migrating to Github has improved the feedback between Ganga users and developers.

Some of the results of this are:

- Performance bottlenecks have been fixed.
- Problems with backend communication have been fixed.
- Data integrity has been improved.
- Stability has improved.

Experiences from working with LSST

Supporting larger VOs typically means constructing a large python script detailing what is required to run each job.
This has a higher maintenance overhead associated with it.

We are now more easily able to write a smaller script and run the same script everywhere as a result of recent developments.

This allows us to be sure that jobs that are running in a “Local” environment are almost identical to jobs run via GridPP DIRAC.

Initial support has been successful but there are potential areas of improvement in the future.

Improvements for larger VO support

Many of the changes that have been made to better support smaller VOs feed back directly into improved support for larger VOs.

Changes originally designed for smaller VO support have been quickly adopted as the best way of supporting larger VOs. This is already benefiting LHCb users.

Development is even more focussed on an expansible Core framework which has several smaller plugins to support each experiment better.

Summary

- Working with smaller VOs to support GridPP DIRAC has been rewarding.
- Increased adoption of Ganga has seen an increase of development activity on the project.
- Able to support multiple VOs well with many existing features within Ganga.
- Developing better support for multiple experiments feeds back into already supported experiments.
- Ganga 6.2.0 is out now and multi-VO support will arrive shortly in 6.3.

Summary

Thanks for listening!