

Table of Contents

Planning for Ganga is 2012.....	1
Outstanding items from the Munich meeting (Kuba).....	1
Review of activities.....	1
Core (Ivan).....	1
ATLAS (Johannes).....	1
LHCb (Alex).....	2
Super B.....	2
Documentation (Kuba).....	2
Testing.....	2
Release manager schedule (Ulrik).....	2
Top 5-10 savannah items.....	2
Migration of stuff from LHCb/ATLAS plugins to Core.....	2
New developments (Tuesday).....	3
Creation of directories in the workspace on demand (Mike).....	3
Postprocessing step (Patrick).....	3
Create an easier documentation for developers, maybe the automatically created documentation but only with a few plugins like and.....	3
Make bootstrap process lighter and consider if there are places where it can be avoided (Mark).....	3
Move plugins that are no longer maintained to a legacy area (Mike).....	3
New output structure (Ivan).....	3
Data management.....	3
Outreach.....	3
Packaging: UBUNTU, DEBIAN, REDHAT?.....	3
Cleanup: Which modules are obsolete and can be removed from the release? also external pkgs?.....	3
ATLAS-specific.....	4
Defined actions.....	4
Short term.....	4
Longer term.....	4
Ideas discussed but not formalised.....	4

Planning for Ganga is 2012

Indico agenda [↗](#)

Outstanding items from the Munich meeting (Kuba)

- ◆ OK:
 - ◇ prepared state
 - ◇ Thread handling
 - ◇ XML based repository
 - ◇ Release scripts
 - ◇ Error reporting tool
 - ◇ Task migration to Core
 - ◇ Data driven processing in LHCb
- ◆ NO:
 - ◇ output handling
 - ◇ composite application handler
 - ◇ Metadata for job objects
- ◆ Consolidation
 - ◇ review and reorganisation of bugs: regular bug review is a problem
 - ◇ certificate package and its handling

Review of activities

Core (Ivan)

- Implementation of prepare state. This seems overall to be working. Problems with all jobs being loaded on exit is now solved? Mike to confirm. Need to disable behaviour in prepare method where files can be copied into inputsandbox (as application not always associated with a job). Default methods in IPrepareApp could do with some default implementation to indicate that it should always be implemented.
- Testing and release framework. Much smoother operation now with the automatic and comprehensive scripts. Can a Savannah entry automatically be created through an API? Still some problems in the ATLAS area where some tests are "forgotten" for some releases. Maybe related to corrupt XML files written. This is also a problem for the

ATLAS (Johannes)

- Roughly a quarter of all DA jobs fail in ATLAS. The HammerCloud test (GangaRobot) is used to separate problems related to users and to the system. Among ATLAS users, Ganga has a user share of 10-20%.
- Several systems in ATLAS where Ganga is the backend for system services (HammerCloud, TAG jobs).
- Many new functionalities providing a closer integration with Panda.
- There is still a perception that Ganga and Panda are separate entities. It is impossible to break this perception.
- The use of the GPI is very limited, with a resistance to consider the advantages of this.
- The main data flow is that jobs produce ROOT files that are stored on a scratch SE. These are subsequently downloaded by users for subsequent analysis.

LHCb (Alex)

- Integration of tasks into LHCb has been quite smooth with the exception that some of the names do not seem that well chosen for the way they are used in LHCb.
- Upcoming features is the splitting of the GangaLHCb plugins into the three plugins GangaLHCb, DIRAC and Gaudi. This will make it possible for other communities to use Gaudi or DIRAC without having odd LHCb dependencies.
- A proposal to include a new post processing or checker state in the workflow that can complete/fail jobs.
- Bulk submission of jobs is an important thing to add to speed up submission of jobs to DIRAC
- More development required to better use the meta data coming from the jobs.

Super B

•

Documentation (Kuba)

- ♦ consolidation of the wikis required
 - ◊ review needed
 - delete old stuff
 - refactor good stuff
 - integrate wiki and main web (editing main web is hard / AFS text files)
 - ♦ update of tutorial and dev survival guide -> how to keep them updated? part of the release procedure?
 - ♦ links to (most-up-to-date-version-of) experiment tutorials from Ganga webpage?

Testing

- ♦ system testing: testing framework
 - ◊ works but hard to maintain and extend, in longer term requires a reimplementaion (Kuba)
 - ◊ should exploit parallelism better (still takes too long)
 - ♦ Core: 269/6, Atlas: 11/6, LHCb: 110/4, NG: 1/4, Panda: 4/1, Robot: 29/5
- shall we go for functional testing (as opposed to unit testing)?

Release manager schedule (Ulrik)

Top 5-10 savannah items

- ♦ XXX open ones
 - ◊ Atlas: XX
 - ◊ Core: XX
 - ◊ LHCb: XX
- ♦ some outstanding examples:

Migration of stuff from LHCb/ATLAS plugins to Core

- ♦ Tasks package (by Johannes)
- ♦ Atlas features (by Mark)
- ♦ LHCb features (by Mike)

New developments (Tuesday)

Creation of directories in the workspace on demand (Mike)

Postprocessing step (Patrick)

It was agreed that we should modify the merger part of Ganga to include any type of post processing including changing the completed/status of jobs. Should be done for Ganga6.

Create an easier documentation for developers, maybe the automatically created documentation but only with a few plugins like and

Make bootstrap process lighter and consider if there are places where it can be avoided (Mark)

Move plugins that are no longer maintained to a legacy area (Mike)

New output structure (Ivan)

Generally agreed that this is a good idea to implement. Obviously this is aimed at Ganga6. Several ideas came up:

- The code for actually making the copy should be supplied by the different file objects for inclusion in the backend code.
- The different file objects should implement a get, put and remove method
- Through a plugin mechanism it should be possible to let a given backend (like LCG) know that it can store files in a new way on the WN. No changes in LCG.py should be required for this.
- One should consider having a fallback method where the WN copy can then declare that it didn't work and the client will retry.
- Both ALTAS and LHCb will have to supply specific file objects for their File object types.
- CompressedFile should move to a type of flag for the different file types.

An idea: pulling out a "automatic resubmit" functionality to be available for simple split jobs (no tasks) with smart strategies on if/when to resubmit.

Data management

Outreach:

- ♦ EGI UF in April 2011: Mini Dev Days or User Days?
- ♦ A Ganga Blog? For attracting new users, and educating ganga users and giving them new ideas how to use ganga, we could ask active developers (and power users) to contribute to a Ganga blog which gives neat examples or announces new or little known features.

Packaging: UBUNTU, DEBIAN, REDHAT?

Cleanup: Which modules are obsolete and can be removed from the release? also external pkgs?

ATLAS-specific

- General strategy discussion on the future of Ganga in Atlas
- Writing more and robust test case for all important workflows in GangaPanda and GangaAtlas
- GangaPanda: review different workflows in Athena and Executable application
- Better general TRF support either through AthenaMC or Athena.type=TRF on Panda and LCG
- Review monitoring plug-ins in ATLAS applications
- script/athena: add support for all Athena, DQ2 and backend options
- Ganga for ATLAS T3s: US sites with Condor plugins and xrootd-splitter, integration with dashboard monitoring (Kuba)
- DQ2JobSplitter is getting very complicated and difficult to validate. Should we factorize it into SplitByFileSize, SplitByNumFiles, SplitByNumEvents, etc...
 - ◆ Wild idea... What about multi-step/multi-level/recursive splitting? This is a general ganga question. Would work like this. j.splitter = [SplitByNumEvents(),SplitByFileSize]. For this, SplitByNumEvents would act on the master job, but SplitByFileSize would act on each subjob, further splitting if that constraint isn't met.

Defined actions

The actions of the meeting were defined in a set of short term Savannah items as listed below and a set of longer terms wishes where the development effort available to them was not currently visible.

Short term

- XX [↗](#)
- Provide configured state for Gaudi applications [↗](#)
- Start a blog for Ganga developers [↗](#)
- Implement metadata for GangaObject [↗](#)
- Ability to build slice from list of jobs [↗](#)
- Refactor main webpage [↗](#)
- Refactor wiki pages. [↗](#)
- Make Job.submit args configurable [↗](#)
- Make Job.submit args configurable [↗](#)
- Remove subprocess and tarfile from input sandbox [↗](#)
- Implement tasks in core [↗](#)
- LHCb tasks to implement a mechanism to process data as it comes in [↗](#)

Longer term

- Rewrite testing framework
- Make input and output more flexible
- Add Ganga into standard Linux distributions such as Ubuntu and Fedora
- Enhance the GangaTutorial package to become a better starting point for new developers.

Ideas discussed but not formalised

- Ability to create command line options "ganga --backend=Batch ..." in an automatic way, maybe driven by schema
- Generalise the client server model in the DiracServer to be a general tool
- Let the monitoring spyware report running time of jobs

-- UlrikEgede - 08-Feb-2012

This topic: ArdaGrid > GangaPlanning2012
Topic revision: r8 - 2012-02-09 - UlrikEgede



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)