Ganga analysis framework for SuperB experiment, work status and plans

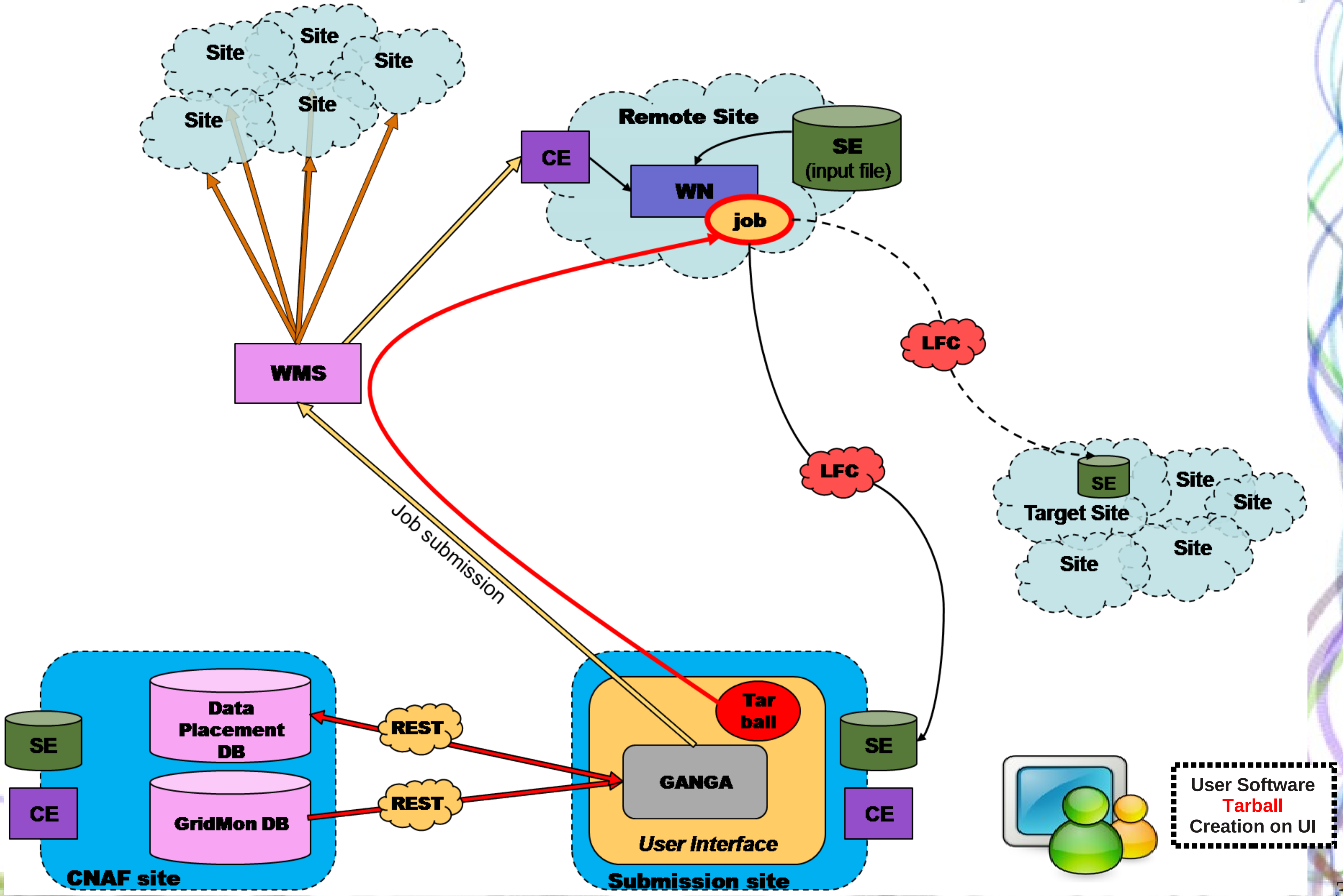
Cristian De Santis is presenting material of
Armando Fella and Andrea Galvani
on behalf of
SuperB Distributed Computing group

Third SuperB Collaboration meeting, 19-23 March, LNF, Frascati(RM)

Presentation Layout

- SuperB experiment overview
- SuperB distributed resources
- Analysis framework: design
- Ganga SuperB plugin
- Work planning and conclusion
- SuperB plugin integration in Ganga general project

Analysis job workflow



User Software Tarball Creation on UI

User analysis work step by step

- **Before accessing ganga**
  - User prepares his sw directory containing executable and configuration files needed for analysis or simulation production purpose
    - Executable should follow few conventions about input file path and location for output files
- **Job preparation from ganga interface**
  - User can select the input dataset, declare the evt to be processed per job, optionally select run and output site, create the output dataset, define executable path, finally submit the job
- **Job run time and stage in procedure**
  - SuperB job wrapper untar user sw package, check environment, transfer requested input files to the conventional area on wn via lcg-cp and launch the user executable
- **Stage out**
  - At job completion, all the output files residing into the conventional area will be transferred and registered in LFC to output site (default: submission site, via lcg-cr). File containing the list of LFN output files is transferred via outputsandbox.
- **Merging step** (to be designed)
- **Data movement**
  - User can perform data transfer of its own dataset via a set of specific methods implemented into SuperB ganga plugin. Not intended to be mass data transfer

Analysis software setup

- User prepares his software directory containing executable script and all files needed for analysis or personal production purpose.
- At job run time on remote node the following environment variables should be used by user executable:
  - WN_INPUTFILES : points to the directory where input files are downloaded.
    - User executable should refer to such a directory for input file access
  - WN_OUTPUTFILES : points to directory where user executable should write its output files to be registered on grid.
  - WN_INPUTLIST : points to a txt file in the $WN_INPUTFILES directory which contains the input files absolute path list.
- The user executable script should comply these information to run correctly.

Analysis use case, general design info

- We use Ganga for job preparation, monitor, submission and limited personal dataset movement
  - Ganga plugin, SuperB job wrapper and DB do all the work
    - No Panda or DQ2 or Dirac like system used/implemented
- Job data management
  - **data driven paradigm**
    - Jobs go where data resides, need a data placement plan for next production, ready before the production start
  - Input files max size: 10GB, Output file max size < 3GB, max job RAM consumption < 1GB
  - Job stage-in
    - The portion of selected dataset assigned to each job is transferred to a conventional area $WN_INPUTFILES via Lcg-Util on running node

Analysis use case, general design info

- We use Ganga for job preparation, monitor, submission and limited personal dataset movement
  - Ganga plugin, SuperB job wrapper and DB do all the work
    - No Panda or DQ2 or Dirac like system used/implemented
- Job data management
  - **data driven paradigm**
    - Jobs go where data resides, need a data placement plan for next production, ready before the production start
  - Input files max size: 10GB, Output file max size < 3GB, max job RAM consumption < 1GB
  - Job stage-in
    - The portion of selected dataset assigned to each job is transferred to a conventional area $WN_INPUTFILES via Lcg-Util on running node

General design info II

- **Job stage-out**
  - Jobs transfer via Lcg-Util the output files on target site SE before completion
    - Default target site is the submission site, in case of problem the system identify a new target site and the user should transfer back the dataset after analysis jobs end.
    - Set of submission site should be still confirmed
      - We can decide to make it match the submission site set with the production dataset site placement: INFN-T1, RAL-LCG2, QMUL, IN2P3-CC, or prefer a human resource availability filter (to be discussed)
  - Access to storage area is VOMS Role driven
    - “Analysis” VOMS Role have write access to related area in SE on submission site
    - Already in place at CNAF (relaxed, Null Role has write access now)
- File catalog LFC
  - Only if user decides to create a dataset for jobs output and the files will be moved/created in WN_OUTPUTFILES at job runtime the system will perform LFC registration
  - Shared unique resource at INFN-T1

Focus on stage out

- If you wish to use the output files of one bulk job (eg: personal production output) as input of another analysis bulk job, you should create a 'dataset' which contains that job's output files.

- Every SuperB subjob will return via outputsandbox
  - output_files.txt that contains the list of LFNs registered in the grid.
  - Job log output files
  - Whatever user need to be transferred back within 50MB of occupancy
    - Next command example will printout the output_files.txt belonging to each subjobs.
      - **!zcat $j.outputdir/output_files.txt.gz**

Focus on stage out II

  - **j.outputsandbox = ['graphs.root']**
  - 'graphs' directory residing in relative path with respect the node job home dir, will be transferred to gangadir on User Interface
- **GRID registration**: as we said every file setup to be owned by a dataset, your executable script put in WN_OUTPUTFILES will be registered in GRID:

- Logical file name: **`/grid/superbvo.org/analysis/<user_identity_from_certificate>/<date_idjob_tarballname_jname>/output_<dataset_id>/subjobid_filename`**

- `Example path for CNAF SE filesystem:`
  - **`/storage/gpfs_superb/sb_analysis/<user_identity_from_certificate>/<date_idjob_tarballname_jname>/output_<dataset_id> /subjobid_filename`**
- **Per site areas:**
  - Scratch for non dataset output
  - sb_analysis area for files owned by dataset and registered in Grid

Ganga plugin DB access

- Various test have been developed to identify the best technology to be used for DB communications
  - REST layer
    - Used for Production System
      - Limited query operations, custom extensions are weak
  - Direct http interface layer
    - Slim, based on json coding, can be an alternative
  - Final test should be performed to take a decision
- Plugin and production code have been ported to manage PostgreSQL interaction

Ganga site setup

- One .ini file per site
- One simple Ganga wrapper launcher per site
  - ${VO_SUPERBVO_ORG_SW_DIR}/ganga/install/${GANGA_REL}/bin/ganga --config-path=${CONFIG_INI_PATH}/analysis_superB.ini -o[Configuration]gangadir=${HOME}/gangadir
- Ganga distributed with experiment software via CVMFS (within few months)
- Ganga can be installed on remote sites via job submission
  - SuperB analysis job wrapper distributed with Ganga plugin
  - Still some problems in site batch system dynamic identification by Ganga. I.e. Would need a standard env var containing the batch system name.

Use cases

- Analysis/reduction:
  - Official FastSim/FullSim production dataset analysis
  - Personal FastSim/FullSim production dataset analysis
  - Generic analysis dataset analysis
- Personal Simulation Production (FastSim and FullSim)
- Dataset management
  - Monitor, research, creation, deletion, status management
  - Dataset transfer

Dataset Manager

- Provides the following methods:
  - **createDataset()** job output dataset creation. Interactive guide to dataset creation is available
  - **deleteDataset()** to delete empty ('prepared') dataset
  - **downloadDataset()** to retrieve all files belonging to a dataset from GRID to submission machine
  - **badDataset()** to set dataset status to 'bad'
  - **closeDataset()** to set dataset status to 'close'
  - **openDataset()** to set dataset status to 'open'
  - **showDatasetDetail()** show all metadata of your datasets

Dataset status

- Prepared
  - New dataset is in 'prepared' status
  - Can be deleted
- Open
  - Dataset with at least one output file associated
  - A dataset automatically become 'open' at first submission
- Closed
  - No more usable as output dataset
  - It is not possible append new output files
- Bad
  - Cannot be used as input dataset in analysis job
  - An automatic cleanup procedure will delete such a dataset

Official production analysis (FastSim)

- **Create a new job object and give it a name:**

```
j = Job()
j.name = 'myJob'
```

- **Assign SBApp as the application used by the job:**

```
j.application = SBApp()
```

- **Set the directory where all the job related files (executable, configuration, etc.) are located. Ganga will create a tarball and ship it in the inputsandbox:**

```
j.application.sw_directory = 
'/storage/gpfs_superb/users/ganga_util/GangaSuperB/test/analysisSoftware'
```

- **Executable *relative* path wrt user job package eg: ./analysisExe.sh**

```
j.application.exepath = 'analysisExe.sh'
```

- **Choose your work session type between analysis and personal production:**

```
j.inputdata = SBInputAnalysis()
```

Official production analysis (FastSim)

- **Input dataset selection:**

```
j.inputdata.getDataset()
```

- **You can filter the results interactively or identify univocally the dataset via dataset_id:**

```
j.inputdata.getDataset(
        prod_series='2010_September_311', prodscript='generic')
j.inputdata.getDataset(dataset_id='4f394214a328d55f2900003b')
```

- **Example of Ganga output:**

```
+----+--------------------+------------+--------------+------+---------------------+---------------+--------+
| id | prod_series        | prodscript | generator    | dg   | tcl                 | analysis_type | status |
+----+--------------------+------------+--------------+------+---------------------+---------------+--------+
| 0  | 2010_September_311 | Generics   | B+B-_generic | DG_4 | MixSuperbBkg_NoPair | HadRecoil     | closed |
| 1  | 2010_September_xyz | Generics   | B+B-_generic | DG_4 | MixSuperbBkg_NoPair | HadRecoil     | closed |
+----+--------------------+------------+--------------+------+---------------------+---------------+--------+
choose dataset: 0

Chosen dataset details:
analysis_type: HadRecoil                         |    generator: B+B-_generic
creation_date: 2012-02-13 18:10:49.885510        |    id: 0
dataset_id: 4f394214a328d55f2900003b             |    occupancy: 121915466273
dg: DG_4                                         |    occupancy_human: 113.5GiB
evt_file: 50000                                  |    prod_series: 2010_September_311
evt_tot: 94500000                                |    prodscript: Generics
evt_tot_human: 94.5M                             |    status: closed
files: 1890                                      |    tcl: MixSuperbBkg_NoPair
```

Ganga developer team collaboration

- Great support by Ganga developer group
- Procedures permitting the insertion of SuperB plugin in official CERN Ganga release almost completed
  - Permit the plugin to be distributed directly with official ganga release
  - Automatic bug identification on releasing step
  - Automatic documentation
  - A CERN external account to manage svn operation have been created
  - Start SuperB collaboration in releasing operations on 2013

SuperB Ganga plugin: documentation

- Ganga official documentation:

  http://ganga.web.cern.ch/ganga/user/html/GangaIntroduction/

- Wiki: user guide

  http://mailman.fe.infn.it/superbwiki/index.php/Distributed_Computing/Ganga_setup_for_SuperB

- Epydoc: technical documentation from pydoc

  http://bbr-serv09.cr.cnaf.infn.it:8080/doc/ganga/

- SuperB plugin is integrated in Ganga's help.

  Inside a session simply type help() at the prompt:

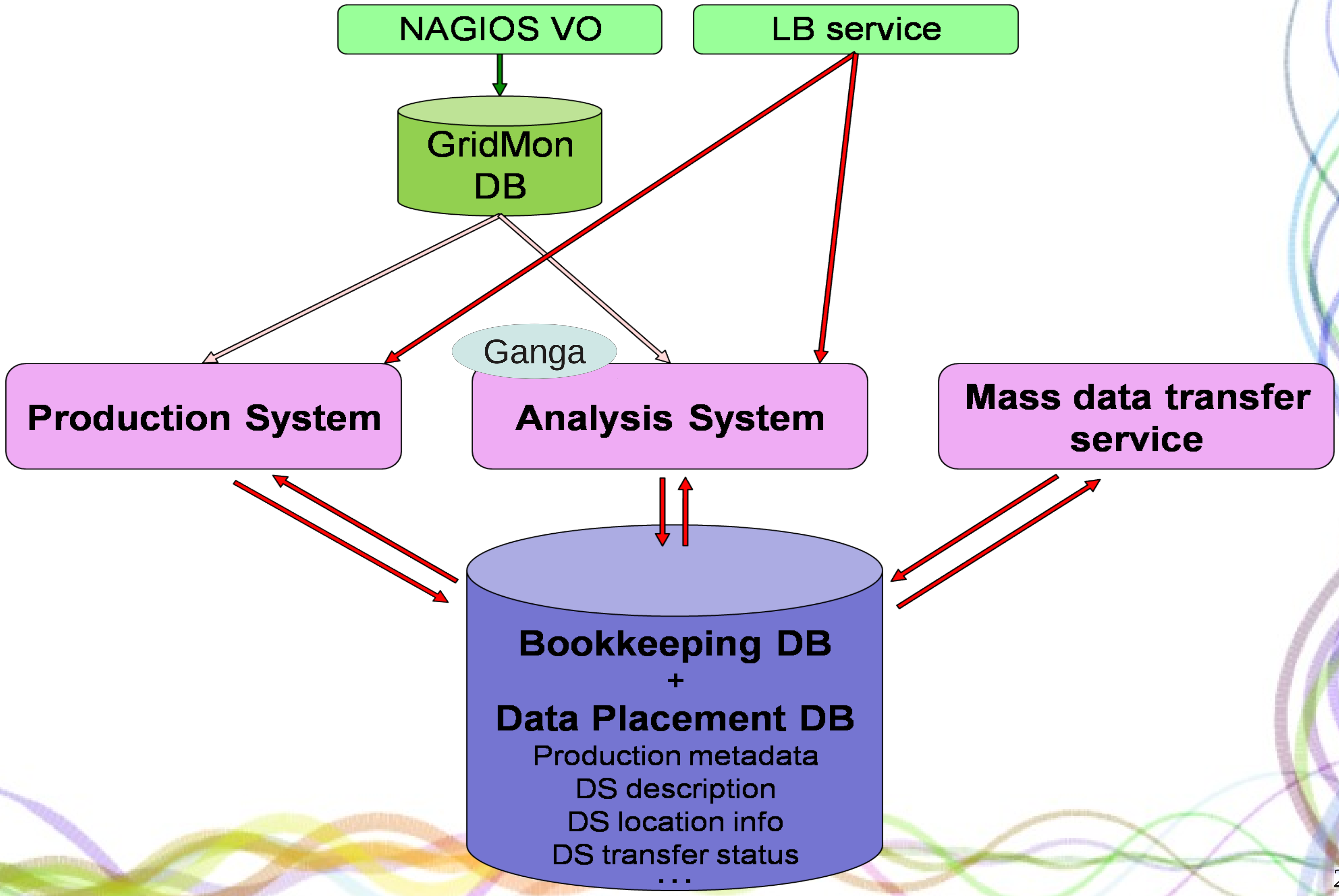
```
In [1]:help()
************************************
This is an interactive help based on standard pydoc help.

Type 'index'  to see GPI help index.
Type 'python' to see standard python help screen.
Type 'interactive' to get online interactive help from an expert.
Type 'quit'   to return to Ganga.
************************************

help>
```

Future work

- Completing code developing for analysis and personal production use case
  - Beta version scheduled for April 15th
- Completing DatasedManager class development (data transfer)
- Create a tight interaction with physics users focused group
  - Training and testing session setup
  - Wiki documentation writing is ongoing
- Ganga developer meeting for code review on last week of April
- Remote site ganga installation and test
- File output merge step implementation
- Resubmission design to be tested
- OSG site exploitation, Condor backend

Thanks all and backup slides

High level scenario


NAGIOS VO
LB service
GridMon DB
Ganga
Production System
Analysis System
Mass data transfer service
Bookkeeping DB
+
Data Placement DB
Production metadata
DS description
DS location info
DS transfer status
...