Distributed computing status and future plans

Armando Fella on behalf of Italian distributed computing group

Distributed Computing human network

CNAF	Italian group
Caltech	Frank Porter, Piti Ongmongkolkul
SLAC	Steffen Luiz, Wei Yang
McGill	Steven Robertson
Queen Mary	Adrian Bevan
RAL	Fergus Wilson
LAL and Lyon	Nicolas Arnaud
Bari	Giacinto Donvito, Vincenzo Spinoso
Legnaro - Padova	Gaetano Maron, Alberto Crescente
Napoli	Silvio Pardi
Ferrara	Giovanni Fontana, Marco Ronzano
Pisa	Alberto Ciampa, Enrico Mazzoni, Dario Fabiani

Email list: **superb-grid-mng@lists.infn.it**

Site setup procedure

Wiki reference page:
http://mailman.fe.infn.it/superbwiki/index.php/How_to_Grid/Site_setup

Three main setup procedure steps:

A) the grid site should enable the VO on each Grid element
B) data handling solution validation: LCG-Utils
C) software installation procedure

For each point, the wiki section for simple test exists:
http://mailman.fe.infn.it/superbwiki/index.php/How_to_Grid/Site_setup/Tests

and job based test:
http://mailman.fe.infn.it/superbwiki/index.php/How_to_Grid/Site_setup/Job

Please send an email to list superb-grid-mng@lists.infn.it reporting the site setup status A, B or C to keep track of progress and permit the planning of next step distributed model validation.

Production tool status

Grid submission: GANGA based job submission test

- job data handling tasks successfully tested
  - lfc name space creation, output transfer via lcg-util
- job submission on remote site with data handling successfully tested
- job output merger solution successfully tested

Production tools development (FastSim)

- web production manager tool:
  - used in successful Nov '09 production
- web monitor tool:
  - the jobs update the SBK DB during run time state

See Luca Tomassetti presentation

What is missing to the full Grid DCM

Grid job communication with bookkeeping DB
in a WAN scenario the job should be able to

- Authenticate it self to the DB service
  - GSI compliant authentication via proxy validation
- Creation of a RESTfull based communication service:
  - Frontier and AMGA systems under study

Grid job management

- Retry policy
  - resubmission procedures and error recovery in Grid env
- Grid based monitor system

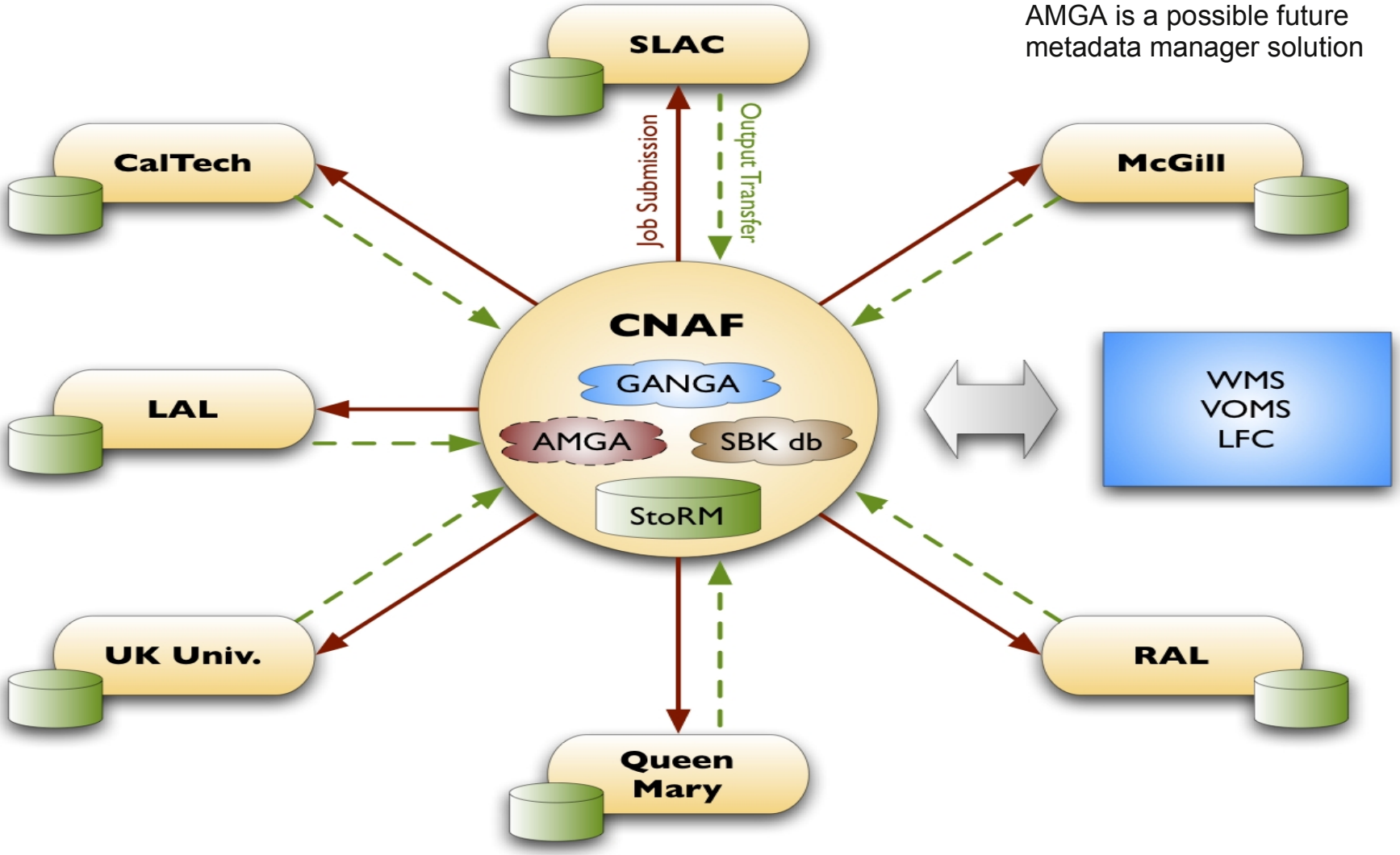
Production design proposals

Next MC productions involve the exploitation of distributed computing resources, two different distributed design systems are proposed:

- the February production distributed system proposal include the installation at remote sites of tools used at CNAF during the Nov '09 test production.
  - the web production interface, local Bookkeeping DB communication, Data handling via direct access to file system
- the setup of a full Grid compliant system will be proposed as solution for April '10 production
  - Full GANGA based job management, Grid Security Interface authentication, central bookkeeping DB communication, data handling via lcg-utils.

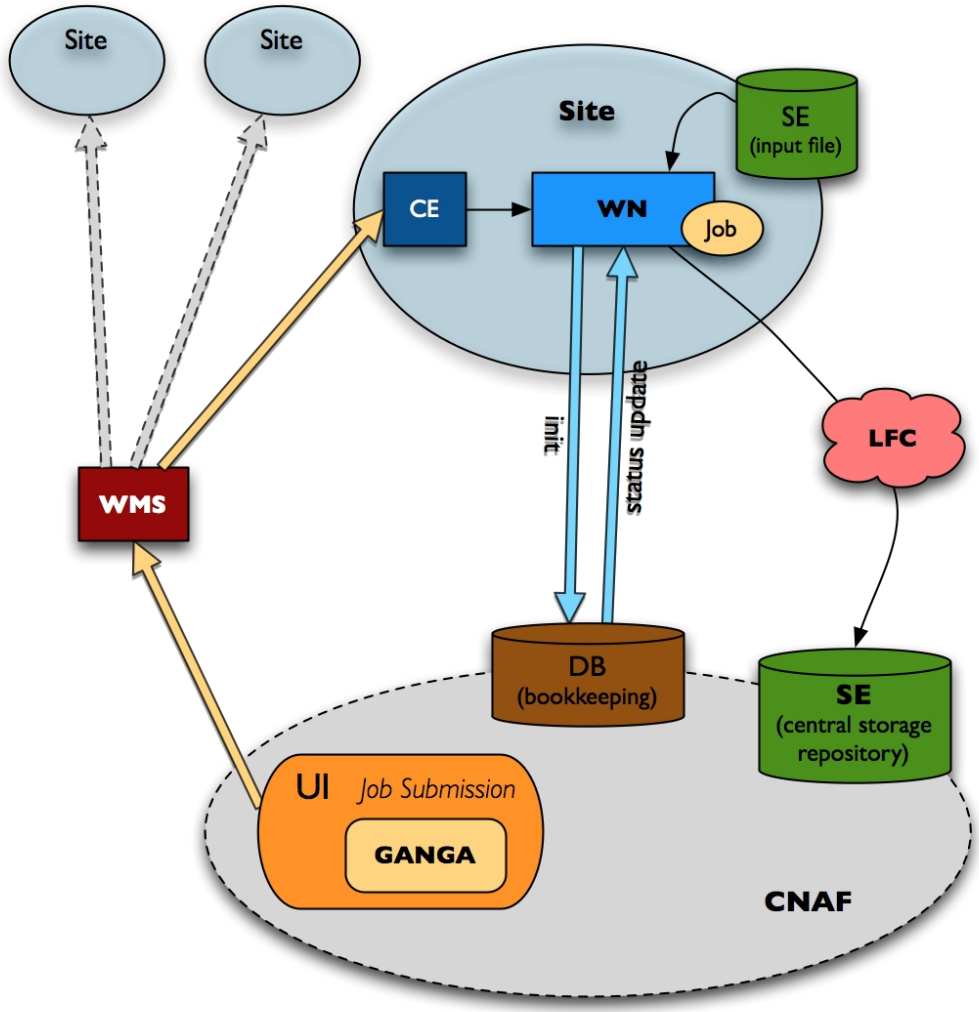
Full Grid integrated production design

AMGA is a possible future metadata manager solution

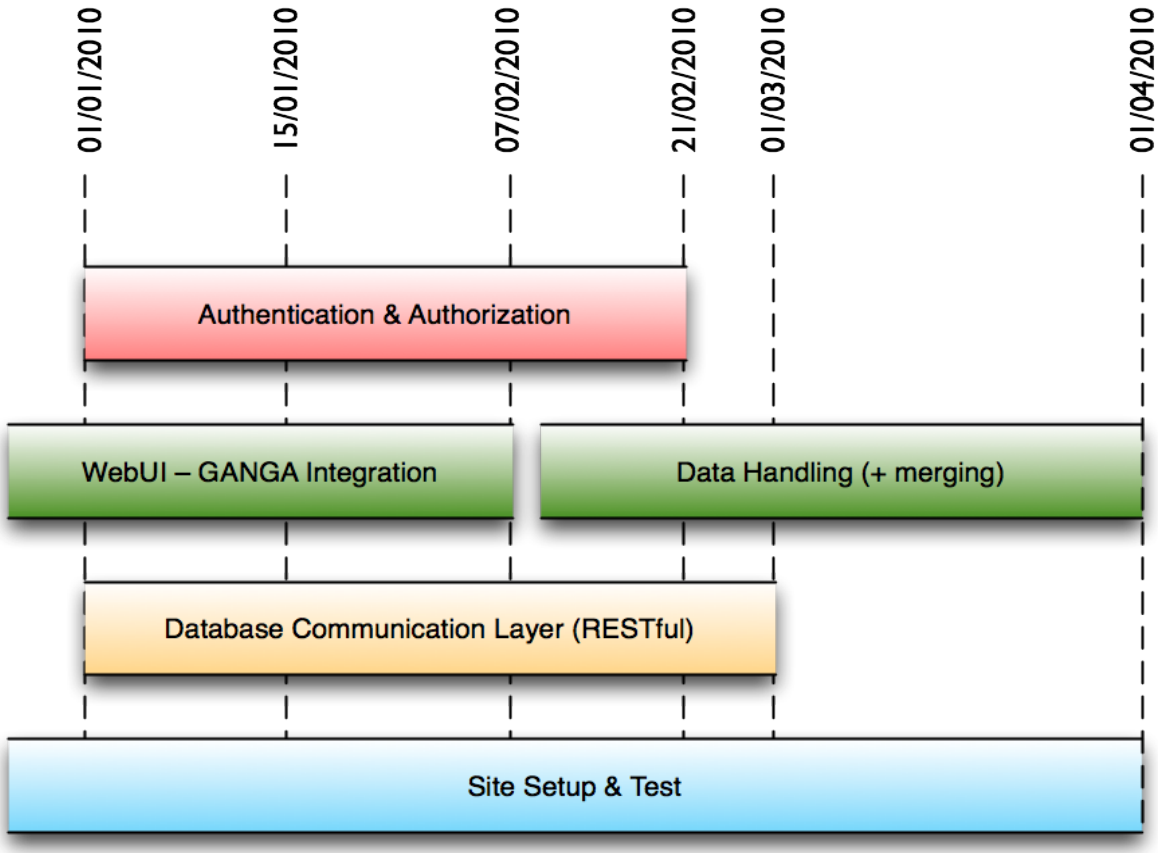


Full Grid integrated production workflow

- The job input files are transferred via
  LCG-Utils to the involved sites
  Storage Elements
- The job submission is performed by GANGA on User Interface at CNAF
- The WMS routes the jobs to the matched sites
- The job is scheduled by the site Computing Element to a Worker Node
- The job during running time
  - accesses the DB for initialization and status update
  - retrieves input files by local Storage Element
  - transfers the output to the CNAF Storage Element



Full Grid integrated production timeline



February production design description

Production system components to be installed at sites:

- Bookkeeping MySQL DB, phpmyadmin as management web interface (optional)
- Apache web server
- PHP 5 scripting language + apache php module
- Web production and monitor interface software (submission to Grid and to batch)

Requirements:

- One server host accessible from worker nodes where the above components should be installed (prod.server@your.site)
- Disk space to contain input production files (local Storage Element)
- Disk space accessible from WN, the output files should be stored in a system five like file system (Eg.: ext3, xfs, jfs, gpfs....). The output files should be copied back to CNAF central storage at end production time.

February production procedure proposal

1) Background files production at CNAF :

- The background files to be used as input for Fast Simulation are produced at CNAF

2) Background files transfer to sites :

- The background files should be transferred to file system accessible from WN

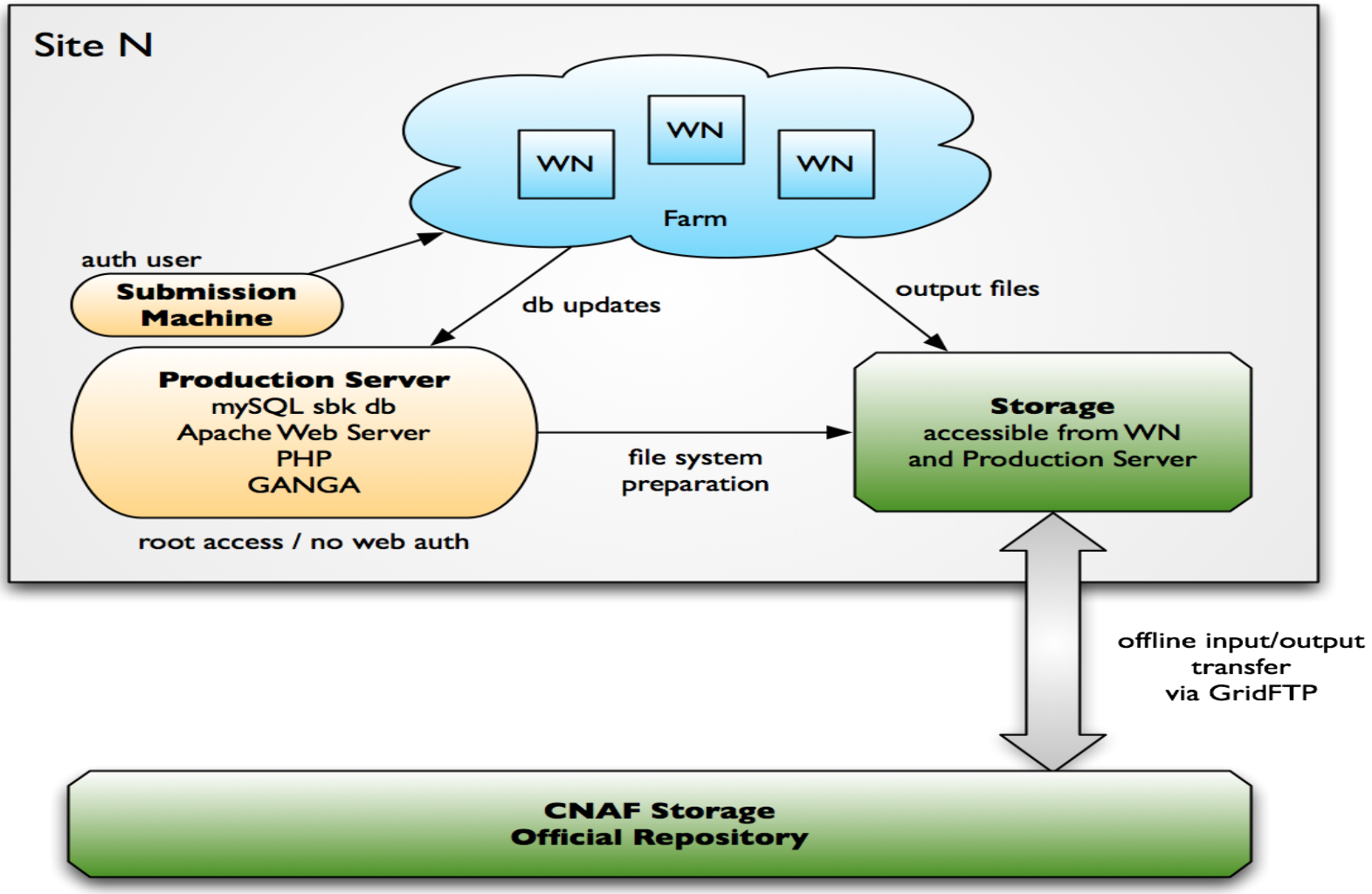
3) Perform the production at sites :

- Use the web User Interface to produce the simulated data
- Store the data into an accessible disk space

4) Output files transfer to CNAF central storage :

- The job output files should be transferred to CNAF

February production procedure proposal



Conclusion & discussion

- The improvement of the actual production system solution moving toward the Full Grid compliant solution is the first point into the to do list.

- What we need to understand/investigate:

- Site resources availability update (CPU, Disk, host)
- sites possible config problems:
  - WN access to disk for accessing r/w input and output files
  - off line files transfer tools

- The site setup procedure is in progress, a new set of requirement should be added to verify the sites capabilities and production system installation procedure

BACKUP Slides

D. C. Proposal II, service description

WMS - Work Load Manager:
- able to manage jobs across different Grid flavors: OSG, EGEE, ARCH
- job routing, bulk submission, retry policy, prologue-job-epilogue structure

LFC - LCG File Catalog:
- file name space manager, map Logical File Name in more replica PFN
- E.g.:/superb/cnaf/2009_July/FastSim/DG_1/B0B0bar_generic/200/s2b.root

LCG-Utils
- data handling LFC compliant, file transfer between SEs, replica manager

GANGA
- Job manager, multi platform (LSF,gLite,Condor,,), job monitor, easy to use
- Wiki: http://mailman.fe.infn.it/superbwiki/index.php/How_to_Grid/How_to_GANGA

SRM (StoRM at CNAF) – Storage Resource Manager V2
- LCG-Utils and LFC compliant, on top of heterogeneous storage systems

Tests with GANGA

- The tests submitted are:
  - Hello World
  - Input/Output Sandbox
  - DataRequirements (with config.LCG.DataRequirements and the lcg-utils)
- The backends where the tests were submitted are:
  - Local
  - LSF
  - LCG (Grid)

Merging

GANGA has different merging tools:

- MultipleMerger
- SmartMerger
- **RootMerger**
- TextMerger
- CustomMerger

During the realization of the job we can integrate some custom functions for the merging of root files (like MergeBkgFiles_C.so);

Custom merging function and sandbox, example

ganga.gpi

```
j=Job()
j.backend="LSF"
j.application=Executable(exe=File('script_for_GANGA_SAND.sh'),args=[])
j.inputsandbox=['InputForFastPatch.root','InputForFastPatch2.root','InputForFastPatch3.r
oot','InputForFastPatch4.root','list.txt','MergeBkgFiles_C.so','merge.C']
j.name="RootLSF_SAND"
j.outputsandbox=['merged.root']
```

script_for_GANGA_SAND.sh

```
#!/bin/sh
root -l hello.C
```

merge.C

```
{
gSystem->Load("MergeBkgFiles_C.so");
MergeBkgFiles("list.txt","merged.root");
}
```

LFC structure

The structure of the LFC for SuperB is

- /grid/
  - superbvo.org/
    - caltech
    - cnaf
    - lal
    - lyon
    - mcgill
    - queenmary
    - ral
    - slac
    - test <== directory used for DH test

Every site has its own name space in /grid/superbvo.org

LCG-utils, most used commands 1/2

- Copy and register a file from localhost to the grid:

```
lcg-cr -v -d t3-srm-01.pd.infn.it -l
  lfn:/grid/superbvo.org/test/<file>
  file:$PWD/<file>
```

- Copy a file from the grid to localhost (used also in the prologue of the job for copying the file from LFC to the Worker Node:

```
lcg-cp -v --vo superbvo.org
 lfn:/grid/superbvo.org/test/locale.txt
 file:$PWD/locale.txt
```

LCG-utils, most used commands 2/2

- Replicate a file from an SE to another:

```
lcg-rep -v --vo superbvo.org -d
  storm02.cr.cnaf.infn.it
  lfn:/grid/superbvo.org/test/<file>
```

Example:

```
#!/bin/sh

lcg-cp -v --vo superbvo.org  lfn:/grid/superbvo.org/test/locale.txt file:$PWD/locale.txt

cp locale.txt ganga_out.txt

echo 'And now...' >> ganga_out.txt
echo 'I am: ' `whoami` >> ganga_out.txt
echo 'My hostname is: ' `hostname` >> ganga_out.txt
echo 'My pwd is: ' `pwd` >> ganga_out.txt


lcg-cr -v -l lfn:/grid/superbvo.org/test/ganga_out.txt -vo superbvo.org file:$PWD/ganga_out.txt
```

Job centric model, step by step

WMS permits to structure the job execution in a three steps procedure:

Prologue - script to be executed before the job start:
- Check the environment – retry policy instance
  - Software availability, services availability, r/w permission on SE, DB access...
- Read from SBK DB job info (http), configuration file creation (eg: MAC,..)
- Fast Sim case: copy the input files (Bkg, Geom) from ClosestSE to Job workspace

Job script – the container of simulation executable:
- Internal checks, Simulation execution
- FullSim case: the job input files (Bkg, Geometry) is transferred via input sandbox

Epilogue - script to be executed after the job completion:
- Check log/err file – retry policy instance
  - Success: Transfer the log, err, output.root to CNAF StoRM (via lcg-utils)
  - Fail: retry submission policy
- Update the SBK DB

GANGA

- Is an easy-to-use frontend for job definition and management
- Allows trivial switching between testing on a local batch system and large-scale processing on Grid resources.
- Is installed in: `/storage/gpfs_babar6/sb/ganga`
- It can be launched with:
- `/storage/gpfs_babar6/sb/ganga/install/5.4.0/bin/ganga`
- Is configured by the file:

  `~/.gangarc`

D. C. Proposal II, workflow

Data management:

- The input files for Fast Sim (Bkg and Geometry) replicated on all the sites
- The WMS accepts bulk submition within 4000 jobs a time
- LCG-utils transfer based has no limit in file size, sandbox limit = 100MB

1) **Production initialization via web interface:**
   - Fill a form with prod parameters (Prod Series, Geometry, Generator...)
   - Submit the form:
     - SBK DB data insertion (status = prepared)
     - Ad-hoc GANGA macro script creation

2) **Launch the GANGA submission via web or via GANGA interface**
   - Monitor I: use GANGA GUI interface to monitor the job status
   - Monitor II: use the web monitor SBK DB based + Data Handling mon

GANGA: job management frontend

http://mailman.fe.infn.it/superbwiki/index.php/How_to_Grid/How_to_GANGA

GANGA has been installed at CNAF accessible on ui01-lcg.cr.cnaf.infn.it
Features of interest:

- bulk submission
- job output merging
- job monitor via GUI interface, text
- multiple backends: LFC, Condor, gLite..

How to Grid/How to GANGA

Contents [hide]

Example 1/2

This is an example of a submission of a job via GANGA in Grid (LCG backend) specifying the CE. The job uses the DataRequirements attribute:

```
/storage/gpfs_babar6/sb/ganga/bin/ganga -i data.gpi
```

data.gpi

```
j=Job()
j.backend='LCG'
j.application=Executable(exe=File('jobscript.sh'), args=[])
j.outputsandbox=['ganga_out.txt']
j.backend.CE='t3-ce-01.pd.infn.it:2119/jobmanager-lcglsf-superbvo'
config.LCG.MatchBeforeSubmit='False'
config.LCG.DataRequirements = '[ DataCatalogType = "DLI"; DataCatalog =
"http://lfcserver.cnaf.infn.it:8085"; InputData = {"lfn:/grid/superbvo.org/test/locale.txt" };]'
config.LCG.RetryCount='1'
config.LCG.ShallowRetryCount='2'
j.submit()
```

Example 2/2

jobscript.sh

```
#!/bin/sh

lcg-cp -v --vo superbvo.org  lfn:/grid/superbvo.org/test/locale.txt file:$PWD/locale.txt

cp locale.txt ganga_out.txt

echo 'And now...' >> ganga_out.txt
echo 'I am: ' `whoami` >> ganga_out.txt
echo 'My hostname is: ' `hostname` >> ganga_out.txt
echo 'My pwd is: ' `pwd` >> ganga_out.txt


lcg-cr  -l lfn:/grid/superbvo.org/test/ganga_out.txt --vo superbvo.org file:$PWD/ganga_out.txt
```