



BESIII and SuperB: Distributed job management with Ganga

I. Antoniev¹, Z. Deng², J. Ebke³, U. Egede⁴, A. Fella⁵, A. Galvani⁶, Y. Han², M. Kenyon¹, L. Lin², E. Luppi⁷, M. Manzari⁷, J. Moscicki¹, C. Nicholson², A. Richards⁴, M. Slater⁸, V. Spinoso⁹, L. Tomassetti⁷, X. Zhang²

¹CERN, Geneva, Switzerland, ²Institute of High Energy Physics (IHEP, China), ³Ludwig-Maximilians-University Muenchen, Germany, ⁴Imperial College London, London, UK, ⁵CNRS, INFN Pisa and University of Ferrara, Pisa, Italy, ⁶University of Ferrara, Ferrara, Italy, ⁷University of Ferrara and INFN, Ferrara, Italy, ⁸Birmingham University, UK, ⁹INFN Sezione di Bari, Bari, Italy



Abstract

A job submission and management tool is one of the necessary components in any distributed computing system. Such a tool should provide a user-friendly interface for physics production and analysis users group to access heterogeneous computing resources, without requiring knowledge of the underlying grid middleware. Ganga, with its common framework and customizable plug-in structure is such a tool. This paper will describe how experiment-specific job-management tools for BESIII and SuperB were developed as Ganga plugins to meet their own unique requirements, discuss and contrast their challenges met and lessons learned.

BESIII and SuperB

The BESIII experiment, at the Institute of High Energy Physics (IHEP) in Beijing, studies electron-positron collisions in the tau-charm threshold region, and has been taking data since 2009 with an average luminosity of $10^{32} \text{cm}^{-2} \text{s}^{-1}$. The SuperB experiment is an asymmetric flavor B factory, under construction in Rome, which will take data in 2016 with a peak luminosity of $10^{36} \text{cm}^{-2} \text{s}^{-1}$. In both cases, the luminosity is a factor 100 higher than what has been previously achieved for a similar collider.



Figure 1 BESIII in Beijing



Figure 2 SuperB in Rome

Distributed computing environment

To meet the challenge of rapidly increasing data volumes in the coming years, BESIII and SuperB are now developing their own distributed computing environment over large-scale distributed resources. In the BESIII distributed computing model, IHEP(Beijing) acts as a central site, mainly responsible for the processing and storage of raw data as well as bulk reconstruction and analysis, while remote sites are responsible for MC production and analysis. To meet peak resource needs at a certain time and make advantage of non-grid resources such as remote clusters and cloud, BESIII plans to use Ganga and DIRAC (Distributed Infrastructure with Remote Agent Control) to provide end users with uniform access to heterogeneous resources.

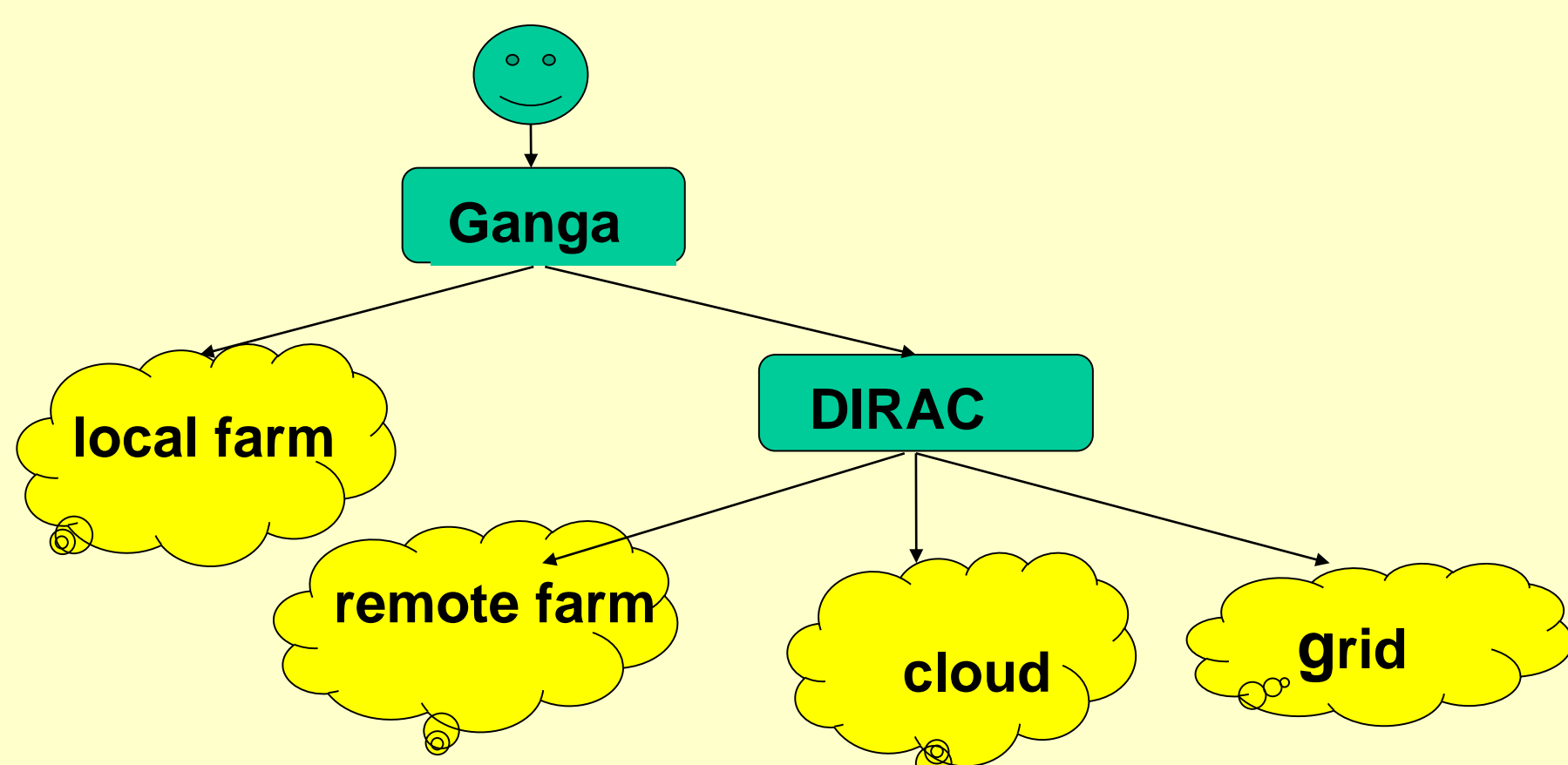


Figure 3 BESIII distributed computing with Ganga and DIRAC

The SuperB distributed computing infrastructure, as of May 2012, includes several sites in Europe and North America. EGI and OSG Grid flavour resources have been enabled at present time. The LHC Computing Grid (LCG) architecture was adopted to provide the minimum set of services and applications upon which the SuperB distributed analysis prototype has been built. Authentication via VOMS service, LFC as file catalog, WMS for brokering purpose and for Grid flavor interoperability features, transfers via Lcg-Utility.

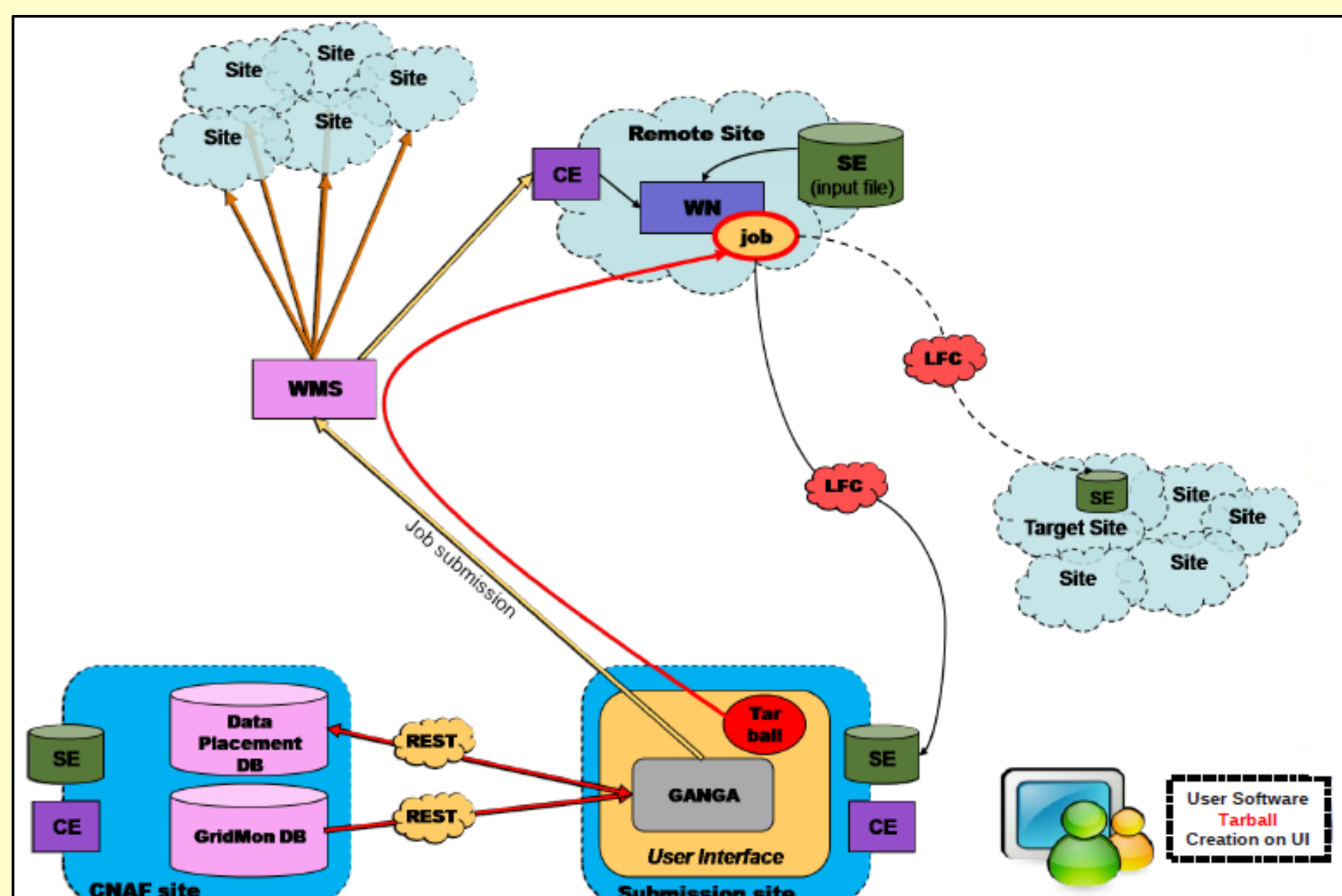


Figure 4 SuperB analysis job workflow

Conclusions and Future work

Despite similarities between the BESIII and SuperB use cases, there are some specific requirements which still need them to have their own experiment-specific plug-ins. BESIII and SuperB have successfully completed Ganga-based production and analysis workflows in their distributed environments. More functionality (such as job and dataset merging, monitoring from a web-based GUI, and so on) is still needed in both BESIII and SuperB cases. The first official version of the SuperB Ganga plugin prototype will be released to the experiment community within the next three months.

BESIII and SuperB Plug-ins

Preparation

Resolve dataset
Parse options files
Package user libraries
Split into subjobs
Generate JDL
Submit subjobs

Run-time

Prepare BOSS Env
Download input files
Start BOSS applications
Validate results
Upload and register output files

Status monitoring

Retrieve job status and log
Merge and register

Preparation

Ganga user application definition
Input dataset selection
Output datasets selection or creation, define Work dir and executable path declaration
Events per job definition and job submission

Run-time, job wrapper operation

User package setup
Input files stage-in via Lcg-Utilities
User executable launch
Output files stage-out via Lcg-Utilities

Status monitoring and data movement

Job status via Ganga proper features
Log files and retrieved via outputsandbox
User dataset can be downloaded to UI
User dataset status management is enabled

BESIII analysis requirements

SuperB analysis workflow

BESIII uses Ganga for both production and analysis. BESIII-specific plug-ins have been developed to support the BOSS (Gaudi-based) analysis framework, DIRAC backend and BESIII dataset management, as shown in Figure 6. This is similar to the LHCb system and is driving the implementation of a common LHCb/BESIII interface to the Gaudi framework and DIRAC WMS. In the BOSS plug-in, OptionsParser parses BOSS option files to get the subjob definition parameters; BossRuntimeHandler generates run-time scripts for worker nodes; and BossSplitter splits MC production jobs according to the BESIII production rules. In the DIRAC plug-in, DiracRuntimeHandler generates BOSS run-time scripts for the DIRAC environment; Dirac communicates with the DIRAC server and DiracSplitter is used to split analysis jobs using the DIRAC API. In the Dataset plug-in, BesDataset is used as a file repository and provides operations at the dataset level. BDQuery and BDRegister take care of dataset querying and registering based on BADGER, the BESIII data management tool.

BOSS
OptionsParser
BossRuntimeHandler
BossSplitter

DIRAC
DiracRuntimeHandler
Dirac
DiracSplitter

Dataset
BesDataset
BDQuery
BDRegister

GangaBoss

Figure 6 BESIII Plug-ins

SuperB use cases include production analysis, permitting analysis of official and personal simulation production datasets, and personal production, permitting simulation of arbitrary user production. Also, SuperB uses Ganga to provide basic dataset management operations, such as monitoring, searching, creation, deletion, transfer, etc. Figure 7 shows the Ganga modules extended by the SuperB plug-in; the class DatasetManager permits basic dataset movement operations.

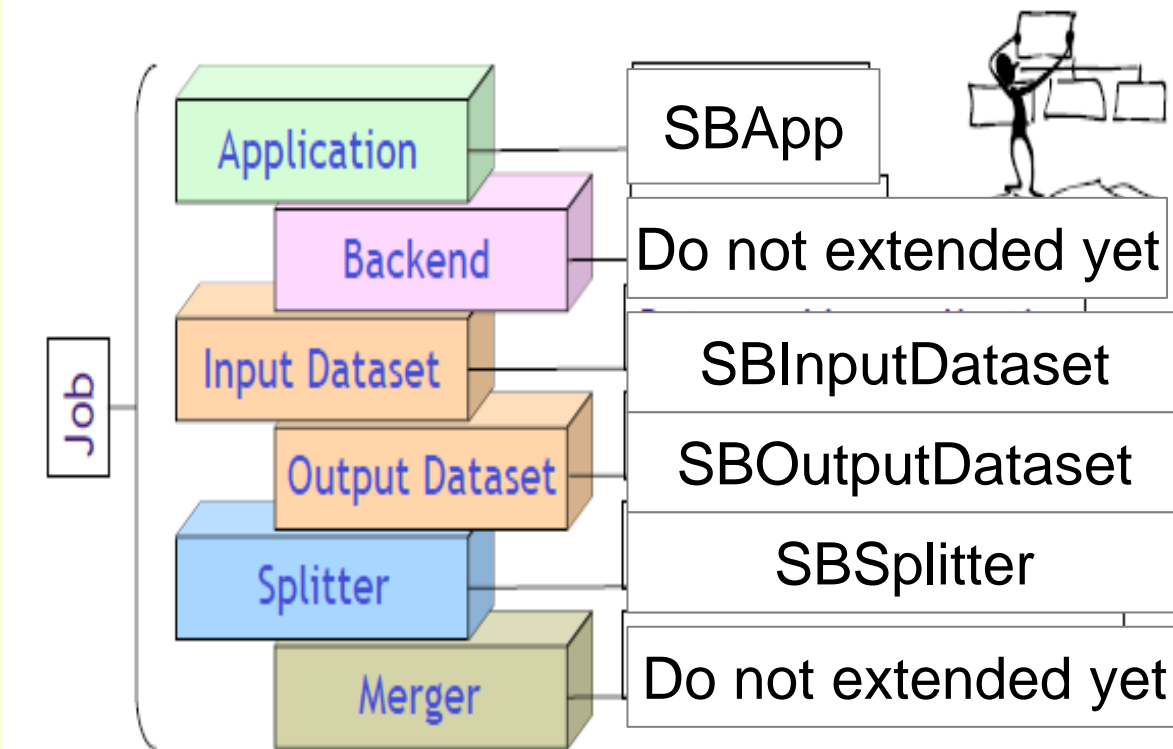


Figure 7 SuperB Plug-in structure

The main components of the prototype SuperB data-analysis suite are the job wrapper, the bookkeeping and data placement databases, and the Ganga plugin layer. Interactions among these three entities realize the user analysis capabilities of a typical mid-sized HEP community.

Implemented use cases

```
j.application = SBApp()
j.application.sw_dir = '/analysisSoftware'
j.application.executable = 'analysisExe.sh'
j.inputdata = SBInputPersonalProduction()
j.inputdata.number_of_subjobs = 5
j.inputdata.background_frame = True
j.inputdata.setSwVersion()
m = SBDatasetManager()
m.createDataset()
j.outputdata = SBOutputDataset()
j.outputdata.setOutputDataset()
j.backend=LCG()
j.submit()
```

SuperB Personal production

```
j.application = SBApp()
j.application.sw_dir = '/analysisSoftware'
j.application.executable = 'analysisExe.sh'
j.inputdata = SBInputProductionAnalysis()
j.inputdata.jobInputDefinitionWizard()
j.outputdata = SBOutputDataset()
m = SBDatasetManager()
m.createDataset()
j.outputdata.setOutputDataset()
j.backend=LCG()
j.submit()
```

SuperB Analysis

```
app = Boss(version='6.6.0',optsfile='sim.txt',recoptsfile='rec.txt')
j=Job(application=app, splitter=BossSplitter(evtMaxPerJob = 50))
j.backend=Dirac()
j.inputsandbox.append("psip.dec")
j.submit()
```

BESIII Production

```
app = Boss(version='6.6.0',user_release_area='/xx/xx',optsfile='pipijpsi.txt')
j=Job(application=app, splitter=DiracSplitter(filesPerJob=1, maxFiles=9))
bkq = BDQueryByMeta(condition='resonance=psip bossVer=655 expNum=exp2')
ds = bkq.getDataset()
j.inputdata = ds
j.backend=Dirac()
j.submit()
```

BESIII Analysis