BESIII and SuperB: distributed job management with Ganga

I. Antoniev\(^1\), Z. Deng\(^2\), J. Ebke\(^3\), U. Egede\(^4\), A. Fella\(^5\), A. Galvani\(^6\), Y. Han\(^2\), M. Kenyon\(^1\), L. Lin\(^7\), E. Luppi\(^8\), M. Manzali\(^8\), J. Moseicki\(^9\), C. Nicholson\(^9\), A. Richards\(^4\), M. Slater\(^10\), V. Spinoso\(^11\), L. Tomassetti\(^8\), X. Zhang\(^2\)

\(^1\)CERN, Geneve, Switzerland
\(^2\)Institute of High Energy Physics (IHEP, China)
\(^3\)Ludwig-Maximilians-University Munich, Germany
\(^4\)Imperial College London, London, UK
\(^5\)CNRS, INFN Pisa and University of Ferrara, Pisa, Italy
\(^6\)University of Ferrara, Ferrara, Italy
\(^7\)Soochow University, China
\(^8\)University of Ferrara and INFN, Ferrara, Italy
\(^9\)Graduate University of the Chinese Academy of Sciences, China
\(^10\)Birmingham University, UK,
\(^11\)INFN 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 groups and ordinary analysis users 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 plug-ins to meet their own unique requirements, discuss and contrast their challenges met and lessons learned.

1. Introduction
The BESIII experiment [1], at the BEPCII accelerator at the Institute for High Energy Physics (IHEP), Beijing, studies electron-positron collisions in the tau-charm threshold region. BESIII started to take data in 2009 with a luminosity of \(10^{32}\) cms\(^{-2}\)s\(^{-1}\), giving an annual volume of raw data in excess of 200 TB. Including simulated data, the total volume of data to be processed can reach 1PB per year. With a planned increase in data taking in the next year, it is clear that the current centralized computing facility at IHEP is not sufficient to meet the experiment’s data processing needs.

The SuperB experiment [2] is an asymmetric flavor B factory, under construction in Rome, which will take data in 2016 with an estimated luminosity of 15 ab\(^{-1}\) per year. With the extremely high planned
luminosity it will be an invaluable source of data on the details of the new physics which will possibly be uncovered at hadron colliders.

In both cases, the luminosity is a factor 100 higher than what has been previously achieved for a similar collider.

2. Distributed Computing System with Ganga

To meet the challenge of rapidly increasing data volumes in the next few years, BESIII and SuperB are now developing their own distributed computing environments across distributed resources.

2.1. BESIII distributed computing environment with DIRAC and Ganga

BESIII offline data processing mainly includes raw data processing, Monte Carlo production and physics analysis. These tasks were originally all done at IHEP, where the raw data is produced. In the new grid-based computing model as shown in figure 1, IHEP 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 Monte Carlo production and small scale physics analysis. IHEP also acts as a regular grid site and provides resources for Monte Carlo production and physics analysis. Besides the steady increase of resources required due to the growing data volume over the years of data taking, another fact also needs to be considered: peak requirements on a short time scale of months or weeks, which often happen after data taking periods and before important physics conferences. To meet peaks in resource demand during these periods, BESIII plans to use more flexible resources such as cloud and volunteer computing as a complement to the grid.

![Figure 1 BESIII distributed computing model](image)

To take advantage of the various distributed resources available, while hiding complexity and heterogeneity from end users, BESIII plans to use Ganga [3] and DIRAC (Distributed Infrastructure with Remote Agent Control) [4] as a middle layer between end users and resources, to provide uniform and convenient access to large-scale heterogeneous resources. As a pilot-based distributed resources management framework, DIRAC is able to provide a single scheduling mechanism to distribute jobs to remote clusters, clouds and grids [5]. Ganga is used as a unified frontend user tool to allow users to process BESIII data on local clusters and DIRAC-based distributed resources, as shown in figure 2.
2.2. SuperB distributed computing
The SuperB distributed computing infrastructure [6], as of May 2012, includes several sites in Europe and North America. EGI and OSG Grid flavor resources have been enabled at the present time. The LHC Computing Grid (LCG) architecture [7] was adopted to provide the minimum set of services and applications upon which the SuperB distributed systems has been built. Particular attention has been dedicated to the choice of Grid services to include in the project. Those selected are widely used by Grid user communities; moreover, the Grid initiatives are committed to supporting them in the long term. A brief description follows:

- Job brokering service: the Workload Management System (WMS) [8], in addition to the specific job brokering and resource matching functionality, implements an inter-Grid interoperability layer permitting job management tasks across different Grid infrastructures.
- Authentication and accounting system: Virtual Organization Membership System (VOMS) [9] manages user authorization to access Grid resources at Virtual Organization (VO) level, providing support for group membership and roles.
- File metadata catalog: the LCG File Catalog (LFC) [10] is a catalog containing logical-to-physical file mappings.
- Data handling: LCG-Utils [11] allows data handling tasks to be performed in a fully Grid compliant manner in terms of both file metadata cataloging and data access.
- Storage resource manager layer: SRM [12] provides data management capabilities in a Grid environment to share, access and transfer data among heterogeneous and geographically distributed data centres.

3. BESIII and SuperB Ganga Plug-ins
Ganga is a user interface which supports large-scale data analysis in both distributed environments and local clusters, providing a set of “building blocks” to prepare and run a job, as shown in figure 3. With its modular architecture, Ganga makes it easy for BESIII and SuperB to add experiment-specific “blocks” or plug-ins to construct their jobs.
3.1. BESIII workflow requirements and GangaBOSS

BESIII uses Ganga for both production and analysis. A typical workflow for large-scale BESIII analysis jobs, as shown in figure 4, can be divided into three job phases: preparation, run-time, and status monitoring. In job preparation, the BESIII dataset specified by users is resolved into a list of input files by BADGER (the BESIII data management tool) [13]; the BOSS (BESIII Offline Software System) [14] option files (used to define input and output files, algorithms to be loaded, physics parameters to be used, etc) are parsed to get application parameters; and the users’ personal libraries are packaged and uploaded to the worker nodes. Using the given input files and application parameters, the job is then split into a certain number of sub-jobs according to BESIII physics rules, with options files, JDL (Job Definition Language) files and application run-time scripts generated for each sub-job. Sub-jobs are submitted to the back end specified by the user. In the run-time phase, the application run-time scripts are run to prepare the BOSS environment, download input files from a SE, start the BOSS application, validate the results after the application has finished, upload output files to a SE, and finally register output files into BADGER. In the status monitoring phase, users need to know the real-time status of jobs and application logs need to be retrieved for error debugging. If necessary, the data files from sub-jobs can be merged and registered as a new dataset in BADGER.
LHCb/BESIII interface to the Gaudi framework and DIRAC workload management system. In the BOSS plug-in, OptionsParser parses BOSS option files to get the subjob definition parameters; Boss generates and packages the files needed by the BOSS application; BossRuntimeHandler generates application run-time scripts for local clusters, and BossSplitter splits Monte Carlo production jobs according to the BESIII production rules. In the Dirac plug-in, DiracRuntimeHandler generates BOSS run-time scripts to run in the Dirac environment; Dirac generate JDL files and submit jobs to the DIRAC back end, and DiracSplitter is used to split analysis jobs using the DIRAC API, with one input file per job. In the Dataset plug-in, BesDataset defines BES-specific datasets and provides various operations at the dataset level. BDQuery and BDRegister take care of querying dataset information by dataset name or by metadata, and registering new datasets through BADGER.

Figure 5  BESIII Ganga plug-ins

3.2. SuperB plug-ins
The main components of the SuperB data analysis prototype suite are the job wrapper, the bookkeeping and data placement database and the Ganga plug-in software layer. This very simple structure is an interesting peculiarity of our implementation. The job wrapper allows the management of stage-in, user application execution, monitoring and stage-out phases. It interacts with the Ganga plug-in for monitoring purposes only. Bookkeeping and data placement databases are the back end providing the Ganga plug-in with all the information needed regarding distributed resources, dataset structure and placement and output metadata. Through interaction with Ganga, the user can manage personal datasets performing creation, deletion and status updates; in addition, it makes it possible to define job input data, allowing the system to automatically create job input data structures. The main Ganga plug-in components, such as Application, InputDataset, OutputDataset and Splitter, have been extended to implement SuperB specific features. Collaboration with the Ganga developer team has been close and productive; after specific code review and test sessions, the SuperB layer has been included into the official Ganga release.

The SuperB analysis prototype is reported here as a step-by-step workflow description. Figure 6 gives a graphical representation of the workflow:

- Working on the User Interface (UI): user creates his/her own working directory including the application and all it needs for analysis or personal production purposes. Executable scripts must comply with a few environment constraints on input and output file placement on remote worker nodes.
- Job preparation from Ganga interface: the user launches Ganga from the UI. He/she can select the input dataset, declare the events to be processed per job, create the output dataset, define the work directory and executable path and finally submit the jobs. Ganga takes care of transferring the user’s working directory to the job working directory at the execution site.
- Job run-time and stage-in procedure: the SuperB job wrapper prepares the user software packages, checks the environment, transfers the requested input files to a conventional area on the worker node and launches the user application.
• Job stage-out phase: at job completion, the output files will be transferred and registered at the output site on the Grid. The list of logical file names (LFN) of output files is transferred to the UI via a Grid provided functionality called output sandbox, together with the job log file.
• Job Monitoring and output retrieval: the Ganga interface permits the user to check job status, manage jobs (kill, resubmit, copy, etc.) and check job logging information.
• Data movement: the SuperB Ganga plug-in includes a set of specific methods permitting the user to perform transfer of his/her own datasets.

The SuperB plug-in classes are a direct extension of the Ganga core classes; as has already been mentioned, the present prototype imports and extends the Application, InputDataset, OutputDataset and Splitter modules. Short term plans include the development of a merging step performed on root job output files.

---

4. Use cases implemented
This section will introduce use cases implemented using the experiment-specific plug-ins and show how BESIII and SuperB physics users use Ganga to do large-scale production and analysis jobs.

4.1. BESIII production and analysis use cases
Examples of how BESIII uses GangaBoss to support Monte Carlo production and analysis workflows are shown in figure 7. In the Monte Carlo production use case, the class “Boss” can accept two options files (optsfile for simulation and recop Southfile for reconstruction) to enable two separate processes - simulation and reconstruction - to run sequentially in one job. That means reconstruction starts directly after simulation, using the raw data files output by the simulation as input, so when both processes are finished only the DST files output by the reconstruction are reserved, uploaded to a SE and registered in BADGER.
Figure 7 BESIII production and analysis use cases

BossSplitter is used to split production jobs, while DiracSplitter is used to split analysis jobs. In the analysis use case, the “user_release_area” is used to specify the location of user libraries to be packaged and uploaded to work nodes. The “BDQueryByMeta” is used to query BADGER to get dataset properties by metadata. If the output data parameters are not provided explicitly, they are automatically provided by parsing options files.

4.2. SuperB

Currently the data analysis suite implements two use cases. The first use case concerns data analysis and data reduction. It is used to analyze both the official and personal Monte Carlo simulation production datasets and to perform the generic analysis of sub-product datasets. The second use case regards the personal Monte Carlo simulation production.

The left section of figure 8 presents the Ganga code for performing an analysis task. The user needs to initialize the job application object instantiating the SBApp() plug-in class; he/she can provide the system with the software directory containing his/her own personal analysis software and the name of the executable script (analysisExe.sh). Input data object definition can be performed using an interactive procedure launching the jobInputDefinitionWizard() method: this permits the selection of input dataset and execution site. At present, a purely data driven model has been implemented. The user can select a private or official experiment dataset derived from centralized simulation production activity. This step includes the definition of the number of the events number per job and subsequently the number of jobs that should be submitted at the end of the procedure. The SBDataManager class allows the user to create an output dataset that will contain the analysis output. The last two
instructions define the Grid middleware to be enabled for the specific submission (the back end), then submits the job.

The right-hand section of figure 8 shows the code for submission of a set of jobs performing a simulation task. The only difference with the analysis use case procedure is in the job input data definition: the instantiation of the SBInputPersonalProduction class allows the user to set the number of jobs to be submitted, the source of simulated events that should be eventually mixed during the simulation process (background_frame) and the simulation software version to be used.

5. Job monitoring

5.1. BESIII job monitoring and usage statistics

BESIII physics users can use the monitoring tools provided by Ganga and DIRAC to get the status of jobs in real time, as shown in figure 9. More than 9000 jobs have been successfully submitted and completed by the BESIII production group through the Ganga user interface to DIRAC, as shown in figure 10.

![Figure 9 Ganga and DIRAC monitoring](image1)

![Figure 10 production jobs submitted by Ganga](image2)
5.2. SuperB
The Ganga configuration file has been modified to add the job wrapper exit status to the output of Ganga monitor commands and a specific code has been developed to customize the exit status results. Moreover, users can access job monitoring information by enabling a native Ganga feature called webgui, which gives access to a web page reporting job details. Currently the job submissions performed using the Ganga plug-in are related to test tasks.

6. 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 plug-in prototype will be released to the experiment community within the next three months. The SuperB data analysis prototype still needs to be tried and tested seriously by the physicist community; besides general design adjustment, the development plans include implementations of new basic functionality in various areas, including file output merging and monitoring.

References
[1] Design and Construction of the BESIII Detector, 2010 Nucl.Instrum. Meth. A614 :345-399
[2] SuperB Collaboration SuperB Progress Reports – Detector, arXiv:1007.4241v1 [physics.ins-det], Report number: INFN/AE 10/4, LAL 10-115, SLAC-R-954
[3] http://ganga.web.cern.ch
[4] Tsaregorodtsev A et al. 2008 J. Phys. Conf. Ser. 119 062048
[5] Sevior M et al. Belle Monte-Carlo production on the Amazon EC2 cloud. 2010 J.Phys. Conf. Ser. 219 012003
[6] Bianchi F et al. Computing for the next generation flavor factories. 2011 J. Phys. Conf. Ser. 331 072012 doi:10.1088/1742-6596/331/7/072012
[7] The LCG TDR Editorial Board, LHC Computing Grid, Technical Design Report LCG-TDR-001 CERN-LHCC-2005-024 (2005)
[8] http://glite.web.cern.ch/glite
[9] http://hep-project-grid-seg.web.cern.ch/hep-project-grid-seg/voms.html
[10] https://twiki.cern.ch/twiki/bin/view/LCG/LfcAdminGuide
[11] http://lcg.web.cern.ch/LCG
[12] http://sdm.lbl.gov/srm-wg/doc/SRM.v2.2.html
[13] Nicholson C et al. 2012 Proc. Computing in High Energy and Nuclear Physics (New York)
[14] Zhang X et al. Raw data management system on GAUDI Framework. 2005 HEP&NP. 29(12):1170-1174