The distributed production system of the SuperB project: description and results

D Brown¹, M Corvo², A Di Simone³, A Fella⁴, E Luppi⁵, E Paoloni⁴, R Stroili², and L Tomassetti⁵

¹Lawrence Berkeley National Laboratory, Berkeley, USA
²University of Padua and with INFN, Padua, Italy
³University of Rome Tor Vergata, Rome, Italy
⁴INFN Pisa, Pisa, Italy
⁵University of Ferrara, Physics Department and with INFN, Ferrara, Italy

E-mail: armando.fella@pi.infn.it

Abstract. The SuperB experiment needs large samples of Monte Carlo simulated events in order to finalize the detector design and to estimate the data analysis performances. The requirements are beyond the capabilities of a single computing farm, so a distributed production model capable of exploiting the existing HEP worldwide distributed computing infrastructure is needed. In this paper we describe the set of tools that have been developed to manage the production of the required simulated events. The production of events follows three main phases: distribution of input data files to the remote site Storage Elements (SE); job submission, via SuperB GANGA interface, to all available remote sites; output files transfer to CNAF repository. The job workflow includes procedures for consistency checking, monitoring, data handling and bookkeeping. A replication mechanism allows storing the job output on the local site SE. Results from 2010 official productions are reported.

1. Introduction

SuperB is an international enterprise aiming at the construction of a very high luminosity asymmetric $e^+e^-$ flavour factory. The physics studies achievable by such a machine will provide a uniquely important source of information about the details of the new physics uncovered at hadron colliders in the coming decade. A heavy flavour factory such as SuperB will be a partner, together with the Large Hadron Collider (LHC) and eventually the International Linear Collider (ILC), in ascertaining exactly what kind of new physics has been found.

The SuperB experiment needs large samples of Monte Carlo simulated events in order to finalize the detector design, to estimate the analysis performances and to support the Technical Design Report studies. The total computing resources needed for one year of data taking at nominal luminosity are of the same order as the corresponding figures estimated, in the spring of 2010, by the Atlas [4] and CMS [5] experiments for the 2011 running period. Therefore a distributed production model capable of exploiting the existing HEP worldwide distributed computing infrastructure is needed. So far, the main effort of the computing group has been devoted to the development and the support of the simulation software tools and the computing production infrastructure needed for carrying out the detector design and performance evaluation studies for the Detector TDR.
Table 1. List of sites and Grid technologies involved in SuperB distributed computing model as of January 2011

| Site name                     | Grid flavor   |
|-------------------------------|---------------|
| CNAF Tier1, Bologna, Italy    | EGI/gLite     |
| RAL Tier1, Oxford, UK         | EGI/gLite     |
| IN2P3 Tier1, Lyon, France     | EGI/gLite     |
| Caltech, California, USA      | OSG/Condor    |
| SLAC, California, USA         | OSG/Condor    |
| Queen Mary, London, UK        | EGI/gLite     |
| RALPP, Manchester, UK         | EGI/gLite     |
| GRIF, Paris/Orsay, France     | EGI/gLite     |
| VICTORIA, Victoria, Canada    | Westgrid      |
| INFN-LNL, Legnaro, Italy      | EGI/gLite     |
| INFN-Pisa, Pisa, Italy        | EGI/gLite     |
| INFN-Barì, Bari, Italy        | EGI/gLite     |
| INFN-Cagliari, Cagliari, Italy| EGI/gLite     |
| INFN-Napoli, Napoli, Italy    | EGI/gLite     |
| INFN-Torino, Torino, Italy    | EGI/gLite     |

This paper describes in details the distributed production system that has been developed to accomplish such a goal.

2. The distributed production environment

The distributed model have been designed is capable of fully exploit the existing HEP worldwide Grid computing system, in particular it has been able to access resources on EGI, OSG and WestGrid infrastructures [10, 11, 12, 13].

The LHC Computing Grid (LCG) architecture [18] was adopted to provide the minimum set of services and applications upon which the SuperB distributed model could be built, and the INFN Tier1 site located at CNAF (Bologna) was chosen as the central site where job submission management, the Bookkeeping Data Base, and the data repository would reside.

Jobs submitted to remote sites transfer their output back to the central repository and update the Bookkeeping Data Base containing all metadata related to the production input and output files. The system uses standard Grid services such as WMS, VOMS, LFC, StoRM, GANGA [15, 16, 17, 19, 20, 14].

The distributed computing infrastructure, as of January 2011, includes several sites in Europe and North America as reported in Table 1. Each site implements a Grid flavor depending on its own affiliation and geographical position. The EGI Workload Manager System (WMS) allows a job’s progress through the different Grid middleware flavors to be managed transparently.

2.1. Simulation production workflow

The production of simulated events is performed in three main phases:

(i) Distribution of input data files to remote site Storage Elements (SE). Jobs running on each site are able to access input files from local SE avoiding a file transfer on the Wide Area Network.

(ii) Job submission, via the SuperB GANGA interface at CNAF, to all available enabled remote sites.

(iii) Job stage out of files to the CNAF repository.

The job workflow, shown in Fig. 1, includes also procedures for correctness check, monitoring, data handling and bookkeeping metadata communication. Reliability and failover conditions...
have been implemented in order to maximize the efficiency for copying the output files to the CNAF repository. A replication mechanism permits to store the job output to the local site SE. The job input files composed by background simulated events are accessed by job simulation executable performing physics and background events merging in order to create the final output file.

The job submission procedure includes a per site customization to adapt the job actions to site peculiarities: e.g., file transfer to and from three different data handling systems: StoRM [20], dCache [21] and DPM [22].

2.2. Production Tools
Both the job submission system and the individual physicist require a way to identify interesting data files and to locate the storage holding them. To make this possible, the experiment needs a data bookkeeping system to maintain the semantic information associated to data files and keep track of the relation between executed jobs and their parameters and outputs.

Moreover, a semi-automatic submission procedure is needed in order to keep data consistent, to speed up production completion and to provide an easy-to-use interface for non-expert users. To accomplish this task, a web-based user interface has been developed which takes care of the database interactions and the job preparation; it also provides basic monitoring functionalities.

A Web-based User Interface (WebUI) has been developed that is bound to the bookkeeping database which provides inputs for the job preparation and monitoring.

It presents two different sections for Full Simulation and Fast Simulation, each of which is divided into submission and monitor subsections. A basic authentication and authorization layer, based on central experiment subversion and ldap system permits the differentiation of users and grants access to the corresponding sections of the site.

A typical production workflow consists of an initialization phase, during which the meta-data of a bunch (or several bunches) of jobs are inserted into the database, and of the subsequent phase of submission either to a batch system or to a distributed environment. The simulation jobs interact extensively with the database during their lifetime to update data and insert outputs and logs.

A production software layer and a database manager layer have thus been developed to
interface the database with the jobs. The prototype service uses a RESTful [24] interface in order to allow the communications between centralized or distributed jobs and a centralized database.

The Fig. 2 shows the job communications with the bookkeeping database during its life cycle: the job bunch is initialized by the production software layer, contextually the DB job table is updated. The status modification is registered via RESTful communication to the DB. At job completion the DB job output and job log tables are updated. Specific test sessions can be enabled via production interface; such a submission scenario enables the job log file to be directly recorded into a specific job log table field for debugging purpose. The production monitor tools access the DB during all offline operations.

The WebUI provides basic monitoring features by querying the bookkeeping database. The user can retrieve the list of jobs as a function of run number range, generator, geometry, physics list, site, status etc. The monitor provides, for each job, the list of output files — if any — and direct access to the log files. Reports on output file size, execution time, site loading, job spread over channels, and the list of the most recently completed jobs (successfully or with failures) are also provided.

The following is the detailed list of services included into the production web suite:

- production creation: this web interface permits the Bookkeeping DB initialization and the production environment setup;
- production request definition: this interface permits the insertion of sub-production lines in terms of physics simulation parameters (physics channels). Sub production priority and minimum job identification number can be defined.
- monitor system: through this section it is possible to access various monitor features: a parametric search engine on jobs metadata, a job log le analysis subsystem, a production report generator, a general set of parametric production result graphs;
- submission interface for shift takers interaction: such a web page includes an automatic submission panel permitting the user to select the available sites and generate the job submission scripts with a single interaction; it includes a per physics channel production selector comprising a job status inline monitor. The bottom of the web page hosts a table showing a per site submission and testing panel; the system calculates the site availability status and dynamically defines the submission policy.
- submission interface for expert interaction: permits a ne grain selection of job submission parameters as physics generator, geometry, background lter, analysis channel. Bookkeeping DB can be enabled to collect job log le for debugging purpose.
eLog system: it is a semi-automatic integrated eLog system collecting information about shift takers actions during production sessions.

The output of the production submission phase is a set of job scripts, ready to be launched on the distributed production infrastructure. The improvements in direct interaction between production tools actions and Grid access tools are in progress.

2.3. 2010 official production runs

The production system has been used during the first large scale productions during 2010. The official productions characteristics and results have been summarized in Figure 3.

More than 11 billion simulated events have been produced. Over an effective period of 4 weeks, approximately 160,000 jobs were completed with a ~8% failure rate. The peak rate reached 7000 simultaneous jobs with an average of 3500. The total wall clock time spent by the simulation executables is 150 years.

3. Conclusions

The computing group works in developing and supporting the simulation software tools and the computing production infrastructure have brought to optimal results in terms of amount of simulated events and Grid infrastructure exploitation efficiency. A fundamental role has been covered by the web production service layer permitting a displaced job life time management for both expert users and shift takers. The registered job fail rate was mostly due to distributed infrastructure unscheduled downtimes. The future production tools improvements include the code porting to python language, the consolidation of data handling procedures and moving to a higher configurable design. A light-weight general-experiment framework version of the SuperB distributed production tools is planned to be released in the near future.

References

[1] The SuperB Collaboration, SuperB Progress Report, Detector. http://arxiv.org/abs/1007.4241v1
[2] B. Aubert et al., Nucl. Instr. Meth. Phys. Res., A 479, 1 (2002).
[3] The Belle Collaboration, The Belle Detector, Nucl. Instrum. Methods Phys. Res., Sect. A 479, 117 (2002).
[4] The ATLAS Collaboration, ATLAS Detector and Physics Performance Technical Design Report, http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/TDR/access.html.
[5] The CMS Collaboration, CMS Detector Technical Design Report, http://cmsdoc.cern.ch/cms/cpt/tdr/.
[6] F. Bianchi, D. Brown, M. Corvo, A. Di Simone, A. Fella, A. Gianoli, E. Luppi, M. Morandin, E. Paoloni, M. Rama, Tomassetti, Computing for the Next Generation Flavour Factories, Proceeding of conference CHEP 2010, Computing for High Energy Physics, Taipei, Taiwan, 18-22 October 2010
[7] R. Andreassen et al, FastSim: fast simulation of the SuperB detector, Proceeding of conference IEEE NSS-MIC 2010, Knoxville, TN, USA (2010) A. Di Simone, I. Gaponenko, E. Manoni, A. Perez, M. Rama, D. Roberts, M. Rotondo, G. Simi, M. Sokoloff, A. Suzuki, J. Walsh, FastSim: fast simulation of the SuperB detector, Proceeding of conference IEEE NSS-MIC 2010, Knoxville, TN, USA (2010)
[8] D. Brown, M. Corvo, A. Di Simone, A. Fella, E. Luppi, E. Paoloni, R. Strollo, L. Tomassetti First Results from the SuperB Simulation Production System. Proceeding of conference IEEE 2010, NSS-MIC 2010, Knoxville, TN, USA (2010)
[9] C. Bozzi, T. Adye, D. Andreotti, E. Antonioli, R. Barlow, B. Bense, D. Boutigny, C. A. J. Brew, D. Colling, R. D. Cowles, P. Elmer, E. Feltresi, A. Forti, G. Grosdidier, A. Hasan, H. Lacker, E. Luppi, J. Martyniak, A. McNab, A. Petzold, D. A. Smith, J. E. Sundermann, and P. Veronesi, *Using the Grid for the BaBar Experiment*, Nuclear Science Symposium Conference Record, 2003 IEEE, 1626 - 1629 Vol.3

[10] http://www.egi.eu
[11] http://www.opensciencegrid.org
[12] http://www.westgrid.ca
[13] The LCG TDR Editorial Board, *LHC Computing Grid, Technical Design Report LCG-TDR-001 CERN-LHCC-2005-024* (2005)
[14] http://ganga.web.cern.ch/ganga
[15] http://glite.web.cern.ch/glite
[16] http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html
[17] https://twiki.cern.ch/twiki/bin/view/LCG/LfcAdminGuide
[18] http://lcg.web.cern.ch/LCG
[19] http://adm.lbl.gov/srm-ug/doc/SRM.v2.2.html
[20] A Corso et al. *StoRM, an SRM Implementation for LHC Analysis Farms Computing in High Energy Physics (CHEP 2006)*, Mumbai, India, Feb. 13-17
[21] P. Fuhrmann and V. Glzow, *dCache, Storage System for the Future*. New York: Lecture Notes in Computer Science/Springer, 2006, vol. 4128, pp. 1106-1113.
[22] https://twiki.cern.ch/twiki/pub/LCG/DataManagementUsefulPresentations/chep_07\_poster_DPM.ppt
[23] http://hadoop.apache.org/
[24] Fielding R.T., *Architectural Styles and The Design of Network-based Software Architectures*, PhD Thesis, University of California Irvine, 2000