Exploiting grid resources for data simulation by using a general-purpose framework

A. Fella[*]
INFN Sezione di Pisa and University of Ferrara
E-mail: armando.fella@pi.infn.it

G. Donvito[†]
University of Bari
E-mail: donvito@ba.infn.it

E. Luppi
University of Ferrara and INFN
E-mail: luppi@fe.infn.it

M. Manzali
University of Ferrara and INFN
E-mail: manzali@fe.infn.it

L. Tomassetti
University of Ferrara and INFN
E-mail: tomassetti@fe.infn.it

E. Vianello
University of Ferrara
E-mail: vianello@fe.infn.it

We present a general-purpose software framework, which allows different multi-disciplinary communities to take advantage of a distributed computational infrastructure. It has been designed specifically for organizations that cannot afford the costs of specialized and complex frameworks, but that still require a user-friendly, standard and highly customizable access to the grid. Our framework heavily relies on a bookkeeping database, where both application-specific and infrastructure metadata are stored, and which is tightly coupled with a web portal. Such a web interface includes various functionalities allowing the different user roles to perform their typical tasks: managers can create and configure resources, application and simulation session parameters, experts are able to perform low level testing tasks and shift takers access the portal sections of job submission, resource and job monitor.

EGI Community Forum 2012 / EMI Second Technical Conference
26-30 March, 2012
Munich, Germany

[*]Corresponding author.
[†]Speaker.

1. Introduction

Many research activities, from several science fields, rely on data simulation, which is a high CPU-consumer. Sequential computation may require months or years of CPU time, so a loose-parallel distributed execution is expected to give benefits to these applications. In fact, the large number of storage and computation resources offered by a grid environment allows to consistently reduce the computing time by splitting the whole task in several parts and executing each part on a separate node.

The potentially intensive grid usage by a large variety of research communities is limited by the difficulty of using the complex software infrastructure of grid computing.

High energy physics experiments made a pioneer work [1, 2, 3] on this but their software, in terms of resource and workload management tools, is still hardly usable by small and mid-size organizations that may have similar computational requirements, mostly owing to the large technical expertise needed.

In the past years some disciplines approached grid technologies. We can mention geology, oceanography, computational chemistry, astronomy, satellite earth observation, climate study, medicine and biology [4, 5, 6, 7, 8]. Grids are being used in health-care in a various ways. There are three main types of grids that can be used: computational grids being used to solve large-scale computation problems in health-care research; data grids that don't share computing power but provide a standardized way for data mining and decision support and collaborative grids that let users share information and work together on extremely large data sets. In particular, bioinformatics evaluates applications in the fields of genomics, proteomics, transcriptomics and drug discovery, reducing data calculation times by distributing the calculation on thousands of computers using the grid infrastructure network. The potential of large distributed computing infrastructures is crucial when dealing with both the complexity of models and the enormous quantity of data, for example, in searching the human genome or when carrying out docking simulations for the study of new drugs.

We developed a prototype software suite that can be seen as a lightweight framework for generic experiments, which focuses on basic functionality, designed specifically for organizations that cannot afford to use the more specialized HEP frameworks but require an easy-to-use interface to the grid.

LHC experiments, in collaboration with grid organizations, during last ten years developed workload and data management systems like Panda [9], Dirac [10] and Alien [11], which give excellent results in terms of efficiency and reliability; these positive results are however not achieved with simplicity of configuration and maintenance. General-purpose goal perspective and, in general, matching small and mid size VOs requirement are not included.

The development of the general-purpose framework described here started from the need of the SuperB project [12, 13, 14], a new High Energy Physics (HEP) experiment, of a grid based detector simulation facility. Section 7 of this paper gives a detailed description of the SuperB experiment and its system utilization case. The effort developing a distributed simulation production system capable of exploiting multi-flavor grid resources resulted in a general purpose design based on a minimal set of standard grid services capable to fit the requirements of many different Virtual Organizations. A web-based user-interface has been developed which takes care of the database interactions and of the job preparation; it also provides basic monitor functionality. The web interface provides a submission interface allowing an automatic submission to all the available sites or a fine grain selection of job submission parameters.

The customization of the web-interface, the bookkeeping database, job executable and site requirements are the key points to achieve the goal as well as small installation and configuration footprint.

2. Project goal, conceptual design and requirements

The project we describe in this paper refers, as background computational environment, to the science grid and more specifically to the LHC Computing Grid (LCG) [22]. A community in grid environment takes the name of Virtual Organization (VO). The project intends to provide an all-in-one software tool giving to small-/mid-size VOs such as experiments, organizations, or communities in general access to distributed resources. Such a tool suite should allow an easy, quick and highly customizable access to the grid resources while keeping the technical details hidden to the client organization. The underlying idea is to minimize the platform requirements in terms of hardware and human resources, development efforts and grid service customization and configuration. The project was designed to manage client applications built up to perform Parallel, Embarrassingly parallel (EP), Coarse-grained (often EP) calculations. The design has been kept light and is based on a minimum set of standard grid services.

Web portal provides a session section dedicated to VO specific task definition. The session interface compilation should be the first act of the VO manager: it customizes its specific work environment. This step will adapt the framework to specific requirements of the VOs and their experiments. More in detail, it includes the definition of job run time environment and job executable parameters, the creation of input and output datasets creation, the recording of application versions and of distributed resources activation.

The following list summarizes the requirements on the technical framework:

- The enabled grid initiatives are European Grid Infrastructure (EGI) [15], Open Science Grid (OSG) [18] and all the grid infrastructures adopting gLite [17] middleware
- The client VO should be defined and enabled at involved grid sites
- Jobs should run where the data to be accessed reside (data driven paradigm)
- Job input files should be of the order of 10GB, job output files should be smaller than 3GB each, RAM consumption should be smaller than 1GB
- A gLite User Interface (UI) should host the framework suite including the backend subsystems: Ganga, PostgreSQL9, Apache, Lcg-Util (see next section for details)
- At least 1 Storage Element (SE) should be available to store jobs result
- At least 1 SE should be available for job input retrieval per site

The system can be replicated and work properly in any EGI site where the related VO has been enabled. The grid services the system relies on can be reached and utilized by remote access.

3. Distributed infrastructure

The framework design is based on a number of VO-enabled sites and one site for the central submission, where the framework is installed. Submitted jobs are instructed to transfer the output files to a predefined target site, discriminating on execution meta-data. The web interface can define a target site for each defined output dataset. A database system is mandatory in order to store all meta-data related to the session input and output files and to allow the retrieval of information. Moreover, it stores the execution status of the jobs and site usage information. The submission scheduling is also based on this database.

The system design assumes a main EGI site hosting the job submission manager, the bookkeeping database and a storage repository. Jobs submitted to remote sites transfer their outputs to the Storage Element of a predefined target site and update the bookkeeping database.

Each site may implement different grid flavors. One of the main problems interfering with the grid concept regards the cross grids interoperability [16]: many steps forward have been done towards a solution and nowadays the choice of using the EGI Workload Manager System (WMS) permits to transparently manage the jobs lives through the different grid middlewares.

The grid services involved in system workflow are widely adopted and long-term supported by grid initiatives; specific attention has been given in selecting the compatible with multi-flavor grid projects. A brief list of the main employed components follows:

- Job brokering service: the Workload Manager System in addition to the job brokering specific tasks, manages jobs across different grid infrastructures (OSG, EGI, NorduGrid [19], WestGrid [20], etc), performs job routing, bulk submission, retry policy, job dependency structure, etc

- Authentication and accounting system: Virtual Organization Membership System (VOMS) [21] is a service developed to solve the problems of granting users authorization to access the resources at the Virtual Organization (VO) level, providing support for group membership and roles.

- File meta-data catalog: the LCG File Catalog (LFC) [22] is a catalog containing logical to physical file mappings. In the LFC, a given file is represented by a Grid Unique Identifier (GUID). Data handling: LCG-Utils [23] allows to perform data handling tasks in a fully LFC/SRMV2 compliant solution.

- Job management system: GANGA [24, 25] is an easy-to-use front-end for job definition and submission management implemented in Python. It provides interfaces for different back-ends (LSF, gLite, Condor, etc.) and includes a light job monitor system with a user-friendly interface. The framework has been configured to use LCG back-end, cross-compatibility among different grid-middleware is guaranteed by the WMS service.

- Storage resource manager: SRM [26] provides data management capabilities in a grid environment to share, access and transfer data among heterogeneous and geographically distributed data centres. StoRM [27, 28], dCache [29], DPM [30], Lustre [31] and Hadoop [32] are some implementations in use by the remote production sites.

4. Bookkeeping database

Both the job submission system and the individual user need a way to identify data files of interest and to locate the storage holding them. Moreover the prompt availability of information on the execution status of jobs and their specific parameters is crucial to the users in order to plan their activities and summarize the results. To make this possible, the developed framework needs a data bookkeeping system to store the semantic information associated to data files and keep track of the relation between executed jobs and their parameters and outputs. The bookkeeping database is extensively used by the job management system in order to schedule subsequent submissions and bring detailed job status and updated site availability information.

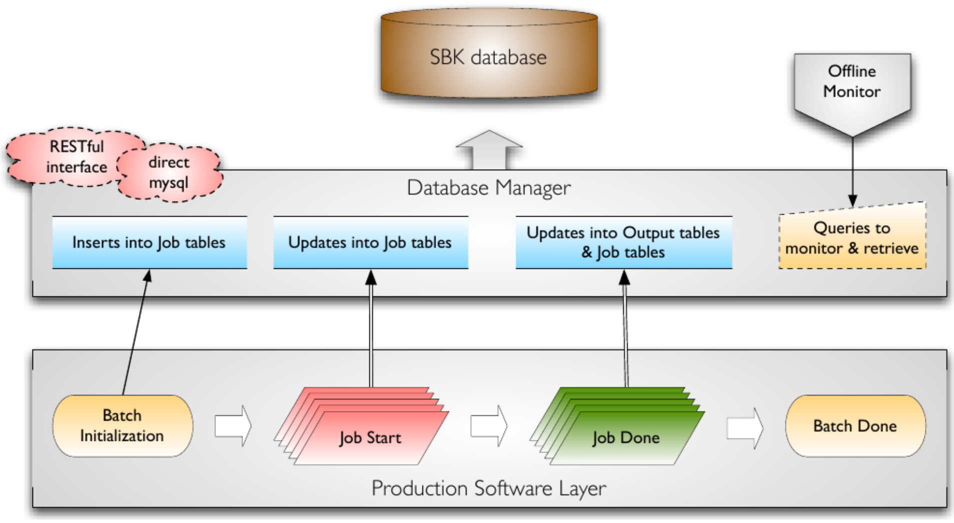


Figure 1: Job life cycle interactions with SuperB bookkeeping database (SBK).

The bookkeeping database was modeled according to the general requirements of a typical simulation production application; its design is sufficiently general to accommodate several use cases from many fields of science while still being self-consistent and structured. Moreover, the schema can be easily extended in order to take into consideration specific requests by new applications still keeping core functionality unaffected.

The present design of the centralized database adheres to the relational model and its current implementation makes use of PostgreSQL rDBMS in a centralized way. The practicability of including and integrating a different data model, schema-free and/or document oriented [33], for instance, is under study. This may help in extending the capabilities of the present, centralized system, by making easier the inclusion of new attributes and structures and moving towards a distributed database solution which can exploit a incremental replication with bi-directional conflict detection and management.

As discussed in the next sections, the bookkeeping database needs to interact both with the submission portal and the job in execution on the WNs. Depending on the sender/receiver these communications are therefore managed by a direct interface to PostgreSQL or a RESTful interface [34]. The latter case is required from remote sites because typically only outbound communication

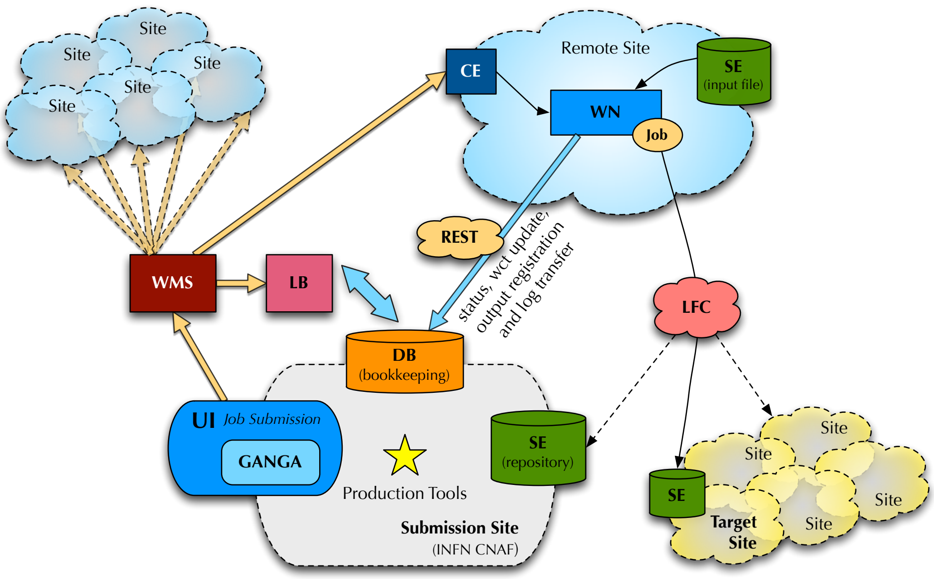


Figure 2: Job life cycle schema.

over http/https is allowed. Strong authentication, by means of X509 proxy certificates over https, is used to grant jobs access to the database.

It is important to stress that such an intensive use of the bookkeeping database by our framework is crucial and distinguishes it from other portal-like solutions available to the community.

5. Job workflow

The submission model is based on direct job brokering on computational resource, without a pilot model. The resource availability monitor allows the system to individuate free resources and efficient sites using standard tools and multiple algorithms: Nagios service availability monitor, effective site banning procedure based on last job bunch efficiency per site.

Job stage in and stage out have been managed via a pure LCG-Utility tool suite; SRMV2 protocol is at the base of all the file access tasks, a failover command chain and retry mechanism guarantees a high efficiency in transfers operations.

The job workflow shown in Fig. 2 is synthesized in the following three steps:

1 Pre-submission steps: the VO managers should install the VO software on all the enabled sites into the official software area. The job input files to be accessed during stage in phase should be transferred to the site's SEs, these transfers should be performed via ClientSRM or LCG-Utils commands ensuring that the files will be recorded into the LFC catalog using replication model. These operations should be performed once per submission set, whenever the application will change its input and software dependencies; in section 7 the SuperB experiment real case has been described; in such a scenario the VO manager is asked to perform pre-submission steps on a simulation production frequency base, about every five months.

2 Job preparation and submission: the web portal provides the interface for job preparation in terms of definition of specific session setup, executable parameters and distributed resources selection. The portal works with session manager VOMS proxy identity, it permits job submission and monitor task to be performed in complete integration with proper grid environment. Submission type is limited to bulk approach and is managed by a Ganga engine configured for such task.

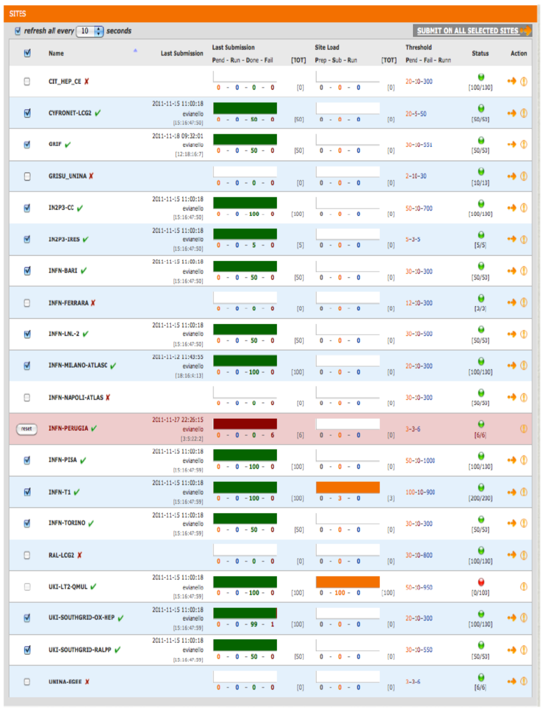
3 Job running time: WMS service routes the job to the requested CE and takes care of grid flavor interoperability. The job starts on a remote site worker node with a general environment test check, subsequently performs the transfer of input files from local SE, launches the VO specific executable and finally sends back to target site SE the generated output and log files. During its entire life cycle the job communicates its status updates to the database via RESTful protocol. Input and output file transfers are performed by LCG-Utilities commands allowing run time file registration to the catalog service, LFC.

6. Web interface

An automated submission procedure is of the utmost importance in order to speed up the user workflow completion, to keep data and meta-data consistent, and to provide an easy-to-use interface for non-expert users. As a matter of fact, the major hurdle in accessing the grid infrastructure for non-expert users is given by its intrinsic complexity and their lack of expertise. To accomplish this task, a Web-based user interface has been developed, which takes care of the bookkeeping database interactions, the job script preparation according to the user's input, and the job management; it also provides basic monitor functionality and procedures to query the stored meta-data. The various grid monitor projects nowadays in place can be fully used side by side with the system embedded monitor features.

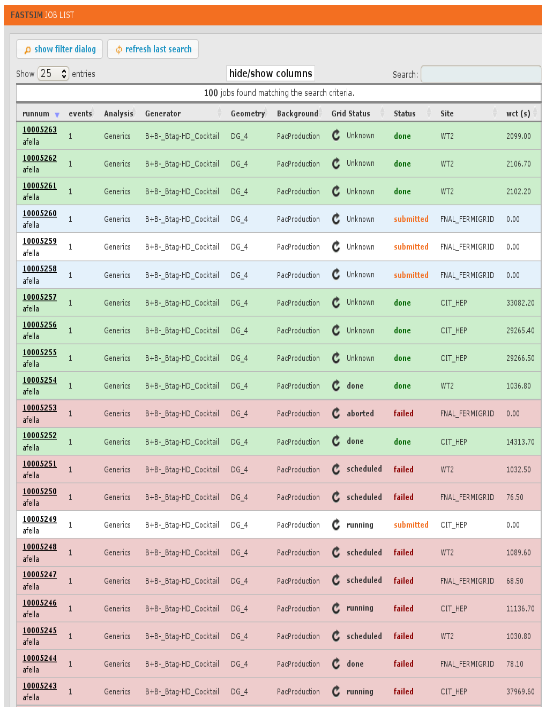
The Web-interface has been developed in PHP server-side scripting language and makes use of Javascript for AJAX functionality. Obviously, it is strictly bound to the bookkeeping database in order to allow the job definition (job initialization phase) and monitoring. It may present several sections, depending on how many job types (executables or set of executables) should be submitted; each of them is divided in a submission and a monitor subsections. Their contents, which mainly consist of web forms, are dynamically generated from the bookkeeping database schema and state in order to include the job-type specific fields. At present time, VO specific constraints should be configured in the Web-interface; although this is a minor effort to cope with, in order to provide an agnostic tool, a completely user-configurable interface is under development. It includes an additional abstraction layer and a corresponding configuration interface.

The Web-interface provides basic monitoring features by means of querying the bookkeeping database and Logging and Bookkeeping (LB) service. The user can retrieve the list of jobs as a function of their unique identifier (or range of them), their specific parameters, the execution site, status, and so forth. The monitor provides, for each job, the list of output files, if any, and a direct access to the corresponding log files. Reports on output file size, execution time, site loading, job spreading over workflow requirements, and the list of the last finished jobs (successfully or with failures) are also provided, see Figure 4. In Figure 3 is shown the portal interface dedicated to shift takers: submission action can be performed per site or on all the enabled sites, submission bulk size per site is predefined in the resource interface. In case of site failure the interface permit the submission of a special test job bunch configured to retrieve log files directly on bookkeeping DB with an increased debug information level. Sites status can be toggled depending on test results.



☑	Name	Last Submission	Last Submission Pend - Run - Done - Fail	[TOT]	Site Load Prep - Sub - Run	[TOT]	Threshold Pend - Fail - Runn	Status	Action
☐	CIT_HEP_CE ✗		0 - 0 - 0 - 0	[0]	0 - 0 - 0	[0]	20-10-300	● [100/100]	
☑	CYFRONET-LCG2 ✓	2011-11-15 11:00:18 evianello [15:16:47:50]	0 - 0 - 50 - 0	[50]	0 - 0 - 0	[0]	20-5-50	● [50/50]	
☑	GRIF ✓	2011-11-18 09:32:01 evianello [12:18:16:7]	0 - 0 - 50 - 0	[50]	0 - 0 - 0	[0]	30-10-551	● [50/50]	
☐	GRISU_UNINA ✗		0 - 0 - 0 - 0	[0]	0 - 0 - 0	[0]	2-10-30	● [10/10]	
☑	IN2P3-CC ✓	2011-11-15 11:00:18 evianello [15:16:47:50]	0 - 0 - 100 - 0	[100]	0 - 0 - 0	[0]	50-10-700	● [100/100]	
☑	IN2P3-IRES ✓	2011-11-15 11:00:18 evianello [15:16:47:50]	0 - 0 - 5 - 0	[5]	0 - 0 - 0	[0]	5-3-5	● [5/5]	
☑	INFN-BARI ✓	2011-11-15 11:00:18 evianello [15:16:47:50]	0 - 0 - 50 - 0	[50]	0 - 0 - 0	[0]	30-10-300	● [50/50]	
☐	INFN-FERRARA ✗		0 - 0 - 0 - 0	[0]	0 - 0 - 0	[0]	12-10-300	● [3/3]	
☑	INFN-LNL-2 ✓	2011-11-15 11:00:18 evianello [15:16:47:50]	0 - 0 - 50 - 0	[50]	0 - 0 - 0	[0]	30-10-500	● [50/50]	
☑	INFN-MILANO-ATLASC ✓	2011-11-12 11:43:55 evianello [18:16:4:13]	0 - 0 - 100 - 0	[100]	0 - 0 - 0	[0]	20-10-300	● [100/100]	
☐	INFN-NAPOLI-ATLAS ✗		0 - 0 - 0 - 0	[0]	0 - 0 - 0	[0]	30-10-300	● [50/50]	
reset	INFN-PERUGIA ✓	2011-11-27 22:26:15 evianello [3:5:22:2]	0 - 0 - 0 - 6	[6]	0 - 0 - 0	[0]	3-3-6	● [6/6]	
☑	INFN-PISA ✓	2011-11-15 11:00:18 evianello [15:16:47:59]	0 - 0 - 100 - 0	[100]	0 - 0 - 0	[0]	50-10-1000	● [100/100]	
☑	INFN-T1 ✓	2011-11-15 11:00:18 evianello [15:16:47:59]	0 - 0 - 100 - 0	[100]	0 - 3 - 0	[3]	100-10-900	● [200/200]	
☑	INFN-TORINO ✓	2011-11-15 11:00:18 evianello [15:16:47:59]	0 - 0 - 50 - 0	[50]	0 - 0 - 0	[0]	30-10-300	● [50/50]	
☐	RAL-LCG2 ✗		0 - 0 - 0 - 0	[0]	0 - 0 - 0	[0]	30-10-800	● [100/100]	
☐	UKI-LT2-QMUL ✓	2011-11-15 11:00:18 evianello [15:16:47:59]	0 - 0 - 100 - 0	[100]	0 - 100 - 0	[100]	50-10-950	● [0/100]	
☑	UKI-SOUTHGRID-OX-HEP ✓	2011-11-15 11:00:18 evianello [15:16:47:59]	0 - 0 - 99 - 1	[100]	0 - 0 - 0	[0]	20-10-300	● [100/100]	
☑	UKI-SOUTHGRID-RALPP ✓	2011-11-15 11:00:18 evianello [15:16:47:59]	0 - 0 - 50 - 0	[50]	0 - 0 - 0	[0]	30-10-550	● [50/50]	
☐	UNINA-EGEE ✗		0 - 0 - 0 - 0	[0]	0 - 0 - 0	[0]	3-3-6	● [6/6]	

Figure 3: Shift taker portal interface.



100 jobs found matching the search criteria.

runnum	events	Analysis	Generator	Geometry	Background	Grid Status	Status	Site	wct (s)
10005263 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	Unknown	done	WT2	2099.00
10005262 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	Unknown	done	WT2	2106.70
10005261 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	Unknown	done	WT2	2102.20
10005260 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	Unknown	submitted	FNAL_FERMIGRID	0.00
10005259 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	Unknown	submitted	FNAL_FERMIGRID	0.00
10005258 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	Unknown	submitted	FNAL_FERMIGRID	0.00
10005257 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	Unknown	done	CIT_HEP	33082.20
10005256 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	Unknown	done	CIT_HEP	29265.40
10005255 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	Unknown	done	CIT_HEP	29266.50
10005254 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	done	done	WT2	1036.80
10005253 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	aborted	failed	FNAL_FERMIGRID	0.00
10005252 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	done	done	CIT_HEP	14313.70
10005251 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	scheduled	failed	WT2	1032.50
10005250 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	scheduled	failed	FNAL_FERMIGRID	76.50
10005249 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	running	submitted	CIT_HEP	0.00
10005248 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	scheduled	failed	WT2	1089.60
10005247 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	scheduled	failed	FNAL_FERMIGRID	68.50
10005246 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	running	failed	CIT_HEP	11136.70
10005245 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	scheduled	failed	WT2	1030.80
10005244 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	done	failed	FNAL_FERMIGRID	78.10
10005243 afella	1	Generics	B+B-_Btag-HD_Cocktail	DG_4	PacProduction	running	failed	CIT_HEP	37969.60

Figure 4: Job monitor portal interface.

The authentication and authorization layer is based on Grid Security Infrastructure (GSI). It makes use of myproxy server to realize a VOMS proxy certificate authentication to permit job submission by web portal and to assure secure communication between job and bookkeeping DB. Web portal access is realized via secure socket layer on https protocol. A web portal submission triggers the launch of a GANGA session, which takes care of submissions of a Python parametric-script to the remote sites. The latter is the effective job that will run on the remote site WNs and will communicate remotely with the bookkeeping database and execute the real user's application.

7. Real case and discussion

The framework conceptual design has been implemented in a prototype serving the use case given by the SuperB project [38, 39]. The Super*B* asymmetric energy e^+e^- collider and detector to be built at the newly founded Nicola Cabibbo Lab [40] will provide a uniquely sensitive probe of New Physics in the flavor sector of the Standard Model. Studying minute effects in the heavy quark and heavy lepton sectors requires a data sample of 75ab^{-1} and a luminosity target of $10^{36}\text{cm}^{-2}\text{s}^{-1}$. Super*B* is an international enterprise, which will study rare decay of the *b* quark, in search of new physics beyond the Standard Model, in channels complementary to the searches made at the LHC.

The requirements of the project include the production of a large number of Monte-Carlo simulated events in a limited amount of time. In particular, the SuperB project needs two types of Monte-Carlo simulations executables, the Full- and the Fast- simulation [42]. Both simulations can produce several types of events, depending on a set of parameter given to the executable at run-time, and may use a few files as input.

The INFN-Tier 1 site at CNAF in Bologna, Italy, has been chosen as the EGI framework service site; it provides the User Interface to the grid and deploys the core functionality of the framework. The distributed computing infrastructure includes 15 sites in Italy, France, UK, USA and Canada, which deploy three different grid middleware (EGI/gLite, OSG/Condor, Westgrid/gLite). Others will be included in the near future. Each site has been carefully configured by enabling the “superbvo.org” VO, installing the software executables and recording in their Storage Elements the required input files. The prototype has been specialized and customized to fulfill the application-specific requirements in terms of authentication/authorization mode, job parameters uses, data manipulation operations, job scheduling and monitoring.

The general-purpose framework has been successfully used for intensive production cycles of both Full- and Fast Simulation [43]. More than 11 billion simulated events have been produced. Over an effective period of 4 weeks, approximately 180000 jobs were completed with a 8.5% failure rate, mainly due to executable errors (6%), site misconfigurations (23.5%), proxy expiration (47%), and temporary overloading of the machine used to receive the data transfers from the remote sites (23.5%). The peak rate reached 7000 simultaneous jobs with an average of 3500. The total wall clock time spent by the simulation executables is 195 years. The distributed infrastructure and the developed framework prototype have been fundamental in achieving the SuperB production cycles goals. The online and offline monitor included with the web-interface keeps the meta-data information stored in the bookkeeping database available for querying and further processing and analysis.

Other projects with similar objectives can be found in literature. For instance, WS-PGrade/gUSE [44] is a complete web portal solution for the development, execution and monitoring of workflows and workflow based parameter studies on various grid platforms. It screens low-level grid access mechanisms and is used by other projects as their base framework. ComputER [45] and SHIWA [46], for example, offer grid access to scientific communities by customizing and developing new features on top of WS-PGrade. Diane [47] is another tool providing an easy access to the grid infrastructure to application communities. JST [48], a web-based tool, helps in subdividing large applications in independent tasks and executes them on the grid nodes; several bioinformatics applications use this tool to exploit grid resources.

Our framework provides an integrated bookkeeping database with the possibility to customize it to the needs of various research communities. The job management system, moreover, makes intensive use of these bookkeeping data in order to monitor job and infrastructure status. These functionalities address both the user requirement of keeping track of specific job meta-data and the goal of building a light-weight framework. These characteristics are distinctive of our suite.

References

[1] F. D. Paoli, *Cdf way to the grid*, Journal of Physics: Conference Series, vol. 53, no. 1, p. 413, 2006.

[2] C. Brew, F. Wilson, G. Castelli, T. Adye, E. Luppi and D. Andreotti, *Babar experience of large scale production on the grid*, e-Science and Grid Computing, International Conference on, vol. 0, p. 151, 2006.

[3] M. C. Vistoli et al., *Prototyping production and analysis frameworks for lhc experiments based on lcg/egee/infn-grid middleware* in Computing in High Energy and Nuclear Physics, 2006.

[4] T. Scholl and A. Kemper, *Community-driven data grids* Proc. VLDB Endow., vol. 1, pp. 1672-1677, August 2008.

[5] N. Pinardi et al., *Very large ensemble ocean forecasting experiment using the grid computing infrastructure*, Bulletin of American Met. Soc., 2008.

[6] G. Bolzon, K. Bylec, A. Cheptsov, A. Del Linz, E. Mauri, P.-M. Poulain, M. Prica, and S. Salon, *Preliminary deployment of grid-assisted oceanographic applications*, Advances in Geosciences, vol. 28, pp. 39-45, 2010.

[7] L. Carota, L. Bartoli, P. Fariselli, P. L. Martelli, L. Montanucci, G. Maggi, and R. Casadio, *High throughput comparison of prokaryotic genomes* in Proceedings of the 7th international conference on Parallel processing and applied mathematics, PPAM07, (Berlin, Heidelberg), pp. 1200-1209, Springer-Verlag, 2008.

[8] A. Weisbecker, J. Falkner, and O. Rienhoff, *Medigrid, grid computing for medicine and life sciences* in Grid Computing (S. C. Lin and E. Yen, eds.), pp. 57-65, Springer US, 2009.10.1007/978-0-387-78417-5 5.

[9] http://iopscience.iop.org/1742-6596/119/6/062036/pdf/1742-6596_119_6_062036.pdf

[10] http://www.diracgrid.org

[11] http://alien2.cern.ch

[12] M. Bona et al., *SuperB: A High-Luminosity Asymmetric e+ e- Super Flavor Factory. Conceptual Design Report*, SLAC-R-856, INFN-AE-07-02, LAL-07-15 (2007), 480pp (arXiv:0709.0451 [hep-ex]).

[13] B. OâĂŹLeary et al., B. *SuperB Progress Report - Physics*, arXiv:1008.1541v1 [hep-ex].

[14] E. Grauges et al., *SuperB Progress Report - Detector*, arXiv:1007.4241v1 [hep-ex].

[15] http://www.egi.eu/

[16] M. Flechl and L. Field, *Grid interoperability: Joining grid information systems*, J. Phys. Conf. Ser., vol. 119, p. 062030, 2008.

[17] http://glite.web.cern.ch/glite/

[18] http://www.opensciencegrid.org

[19] http://www.nordugrid.org

[20] http://www.westgrid.ca

[21] http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html

[22] https://twiki.cern.ch/twiki/bin/view/lcg/

[23] http://lcg.web.cern.ch/lcg

[24] F. Brochu, U. Egede, J. Elmsheuser, K. Harrison, R. W. L. Jones, H. C. Lee, D. Liko, A. Maier, J. T. Moscicki, A. Muraru, G. N. Patrick, K. Pajchel, W. Reece, B. H. Samset, M. W. Slater, A. Soroko, C. L. Tan, and D. C. Vanderster, *Ganga: a tool for computational-task management and easy access to grid resources*, CoRR, vol. abs/0902.2685, 2009.

[25] http://ganga.web.cern.ch/ganga

[26] http://sdm.lbl.gov/srm-wg/doc/srm.v2.2.html

[27] E. Corso, S. Cozzini, A. Forti, A. Ghiselli, L. Magnoni, A. Messina, A. Nobile, A. Terpin, V. Vagnoni, and R. Zappi, *StoRM: A SRM Solution on Disk Based Storage System*, in Proceedings of the Cracow Grid Workshop 2006 (CGW2006), Cracow, Poland, 2006.

[28] http://storm.forge.cnaf.infn.it/

[29] http://www.dcache.org/

[30] https://svnweb.cern.ch/trac/lcgdm/wiki/dpm

[31] http://www.lustre.org

[32] http://hadoop.apache.org/

[33] https://en.wikipedia.org/wiki/Document-oriented_database

[34] R. T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, PhD. Thesis, University of California Irvine, 2000.

[35] J. Basney, M. Humphrey and V.Welch, *The MyProxy online credential repository* published online in Wiley InterScience. DOI:10.1002/spe.688.

[36] http://grid.pd.infn.it/cream/

[37] Kacsuk et al., *Ws-pgrade: Supporting parameter sweep applications in workflow*, in Proceeding of 3rd Workshop on WorkïňĆows in Support of Large-Scale Science, in conjunction with SC 2008, 2008.

[38] SuperB Collaboration, E. Grauges, F. Forti, B. N. Ratcliff, and D. Aston, SuperB Progress Reports - Detector ArXiv e-prints, July 2010.

[39] F. Bianchi et al., *Computing for the next generation flavour factories* in Proceeding of the CHEP 2010 conference, 2011.

[40] http://www.cabibbolab.it

[41] http://www.linearcollider.org

[42] R. Andreassen et al., *Fastsim: fast simulation of the SuperB detector*, in Proceeding of the IEEE NSS-MIC 2010 conference, 2010.

[43] D. Brown et al., *First results from the SuperB simulation production system*, in Proceeding of the IEEE NSS-MIC 2010 conference, 2010.

[44] https://guse.sztaki.hu/liferay-portal-6.0.5/welcome

[45] P.Veronesi et al., *Multidisciplinary approach for computing in Emilia Romagna (Italy)*, EGI User Forum, 11-14 april 2011, Vilnius, Lithuania.

[46] http://www.shiwa-workflow.eu/home

[47] http://it-proj-diane.web.cern.ch/it-proj-diane/index.php

[48] G. Cuscela, G. P. Maggi, and G. Donvito, *Job Submission Tool, web interface and WebDAV data management*, EGI User Forum, 11-14 april 2011, Vilnius, Lithuania.