A GENERAL PURPOSE SUITE FOR JOB MANAGEMENT, BOOKKEEPING, AND GRID SUBMISSION

Armando Fella[1], Eleonora Luppi[2] and Luca Tomassetti[2]

[1]INFN Sezione di Pisa ,via Pontecorvo, 3 I-56127 Pisa, Italy
armando.fella@pi.infn.it
[2]University of Ferrara and INFN, via Saragat, 1 I-44122 Ferrara, Italy
eleonora.luppi@fe.infn.it, luca.tomassetti@fe.infn.it

ABSTRACT

This paper briefly presents the prototype of a software framework permitting different multi-disciplinary user communities to take advantage of the power of the Grid computing. The idea behind the project is to offer a software infrastructure allowing an easy, quick and customizable access to the Grid to research groups or organizations that need to simulate big amount of data.

KEYWORDS

Distributed Systems, Distributed Applications, Grid Computing, Job Management, Monte Carlo Simulation

1. INTRODUCTION

Many research activities from several fields, such as high energy, nuclear and atomic physics, biology, medicine, geophysics and environmental science, rely on data simulation, which are high CPU-consuming. 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 allow to consistently reduce the amount of computation time by splitting the whole task in several parts and executing each part on a single node.

The potential intensive Grid usage by a large variety of research communities is reduced by the difficulties that researchers can encounter in using the complex software infrastructure of Grid computing. High energy physics experiments made a pioneer work [1-3] on this but their results are still hardly available to small and mid-size organizations that may have similar computational requirements, mostly due to the large amount of needed technical expertise.

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 [4-8]. Grids are being used in healthcare 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 healthcare 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

DOI: 10.5121/ijgca.2011.2202

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 general-experiment framework which focuses on basic functionalities, designed specifically for organizations that cannot afford the use of the more specialized HEP frameworks but that still require an easy-to-use interface to the Grid.

This prototype started from the necessity of the SuperB [9] project, a new High Energy Physics (HEP) experiment, to be able in simulating detector systems.

The effort in developing a distributed simulation production system capable of exploiting multi-flavor Grid resources resulted in a general-purpose design based on minimal and standard set of Grid services and capable to fit the requirements of many different Virtual Organizations [10]. A web-based user-interface has been developed which takes care of the database interactions and the job preparation; it also provides basic monitor functionalities. 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. CONCEPTUAL DESIGN AND DISTRIBUTED INFRASTRUCTURE

The prototype we developed is based on some simple requirements: manage the applications, manage data and metadata, submit jobs on the Grid, monitor applications and Grid infrastructure status. Its design has been kept light and based on a minimum set of standard Grid services.

The prototype design involves a limited number of sites with one site acting as a central repository of data. This model can be easily transformed to a decentralized design in which selected job subset are instructed to transfer the output files to a predefined site, discriminating on execution metadata. A database system is mandatory in order to store all metadata related to the production 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 based on this same database.

The centralized system design includes a main European Grid Infrastructure (EGI) [11] site hosting the job submission manager, the bookkeeping database and the central storage repository. Jobs submitted to remote sites transfer their output back to central repository and update the bookkeeping database. In addition to the central service site, the framework requires a proper configuration of the remote Grid sites on which the jobs will run.

Each site may implement different Grid flavor, depending on its own affiliation, geographical position and political scenario. One of the main problem interfering with the Grid concept itself regards the cross Grids interoperability [12]: many steps forward a solution have been done and nowadays the choice of using the EGI Workload Manager System (WMS) [13] permits to manage transparently the jobs life through the different Grid middlewares.

The involved Grid services are briefly described in the following:

Job brokering service: the Workload Manager System in addition to the job brokering specific tasks, manages jobs across different Grid infrastructures (OSG [14], EGI, NorduGrid [15], WestGrid [16], etc…), performs job routing, bulk submission, retry policy, job dependency structure, etc…

Authentication and accounting system: Virtual Organization Membership System (VOMS) [17] 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. A Virtual Organization is a group of entities sharing the same project and identity on the Grid.

File metadata catalogue: the LCG File Catalogue (LFC) [18] is a catalogue containing logical to physical file mappings. In the LFC, a given file is represented by a Grid Unique Identifier (GUID).

Data handling: LCG-Utils [19] permits to perform data handling tasks in a fully LFC/SRMV2 compliant solution.

Job management system: GANGA [20, 21] is an easy-to-use frontend for job definition and submission management implemented in Python. It provides interfaces for different backends (LSF, gLite, Condor, etc.) and includes a light job monitor system with a user-friendly interface. The framework has been configured to use LCG backend, cross-compatibility among different Grid-middlewares is guaranteed by the WMS service.

Storage resource manager: SRM [22] provides data management capabilities in a Grid environment to share, access and transfer data among heterogeneous and geographically distributed data centres. StoRM [23, 24], dCache [25], DPM [26], Lustre [27] and Hadoop [28] are some implementations in use by the remote sites involved in the production distributed system deployment at present time.

3. BOOKKEEPING DATABASE

Both the job submission system and the individual user require a way to identify interesting data files and to locate the storage holding them. Moreover the prompt availability of information on the execution status of jobs and their specific meaning and 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.

This same bookkeeping database is extensively used by the job management system itself in order to schedule subsequent submissions and bring completion level of requests and site availability information up to date.

The bookkeeping database was modelled 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 although being self-consistent and structured at the same time. Moreover, the schema can be easily extended in order to take into consideration new applications specificities, nevertheless by keeping core functionalities unaffected.

At the moment, its design adheres to the relational model and the current implementation makes use of MySQL rDBMS in a centralized way. The practicability of including and integrating a different data model – schema-free and/or document oriented, for instance – is under study. This may help in extending the covered use case by making easier the inclusion of new attributes and structures and going towards a distributed 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 either with the submission portal or the job in execution on the WNs. Depending on the sender/receiver these communications are therefore managed by a direct interface to MySQL or a RESTful interface.

The latter case is required from remote sites because typically only outbound communication 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 permits to distinguish it from others portal-like solutions available to the community.

4. Job Workflow

The key concepts at the base of job workflow design are "standard" and "stable": from the point of view of Grid services, job submission path is determined by a direct routing to selected site CE via WMS service, submission method is limited to bulk. The inclusion of WMS service into the job workflow permits to exploit Grid flavor interoperability features. The job JDL file do not include data handling management neither customization in terms of JobType, moreover the design do not include the output sandbox withdraw at job completion as it is described in section 2. Data handling system relies on SRMV2 standard protocol implemented at LCG-Util layer permitting a transparent interaction with heterogeneous site SE storage management systems. Authentication and file catalogue are respectively managed by VOMS and LFC services. All the cited EGI Grid services have been included in LHC experiment computing models and largely used in several VO distributed system design. Furthermore such a set of services have a consolidated set of functionalities and have been identified as long term projects in EGI roadmap definition.

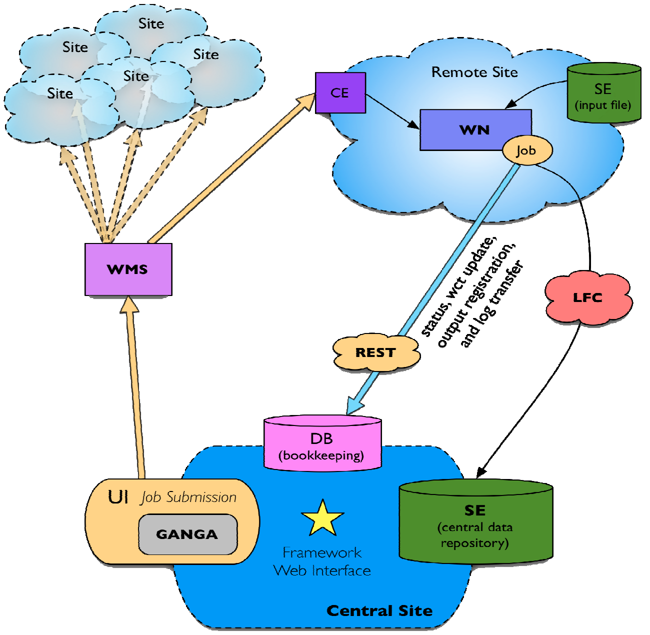

Figure 1. Job Workflow graphical representation.

The job workflow shown in Fig.1 is synthesized in the following three steps:

1. Pre-submission operations: the distributed system design includes the possibility of jobs to have access to input data, of the order of O(10GB) in size. Input job data are expected to be the strictly necessary information permitting job simulation specificity. The transfer of such a set of files to the remote sites SE is an offline operation. The VO specific software should be installed into the involved sites.
2. Job preparation and submission: the job script definition and database session initialization are performed via production tools layer at central site. The VO specific parameters and environment setup are defined through web portal interaction. The resulting script should be launched from UI by a VOMS authorized user. An automatic portal submission is under deployment, it is based on MyProxy service [29]. Job bulk submission is performed by GANGA system properly configured to be able to interact with LCG environment.
3. Job running time: WMS service routes the job to the requested CE and takes care of Grid flavor interoperability. The job starts on 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 central site repository the generated output and log files. During the entire life cycle the job communicates its own status updates to the database via RESTful protocol. Input and output file transfers are performed by LCG-Utilities commands permitting runtime file registration to the catalogue service, LFC.

4.1. Grid submission via GANGA system

Grid computational resources are accessed via gLite suite with the use of WMS. The brokering system permits to distinguish nowadays CREAM CEs [30] from LCG CEs specificities. The short-term planned WMS updates include migration of stuck jobs to new, JDL defined, sites and in general to access Logging and Bookkeeping job historical information. This will increase brokering quality. The stand alone distributed system we are discussing here foresees the enabling of such a new set of features as them will be available. The second core reason in the use of WMS system regards the embedded interoperability active mechanisms permitting to masquerade the differences between Grid flavours interactions.

The job submission management is delegated to GANGA system. Various studies and configuration tests have been performed with the aim of customize GANGA system to be able to work as a simple and efficient submission manager. The lines of intervention can be summarized in the following main groups:

- **sub services clean-up procedure**: the deactivation of all the services around the core submission routine as job monitoring daemons, user interactive interface, job specific feedbacks, integrity checks, automatic GANGA specific resubmission policy;
- **bulk as unique active submission method**: specialization in bulk submission procedure included decreasing of GANGA submission response time;
- **Grid job specific information collection**: Grid job Id is an example of information the bookkeeping DB need to store, it can be retrieved from the submission process itself.

GANGA developer team expressed interests in this use case and an active collaboration on the specific subject started.

A GANGA specific script has been developed to permit run time customization and site specific JDL file generation.

The results of the use of GANGA system in this particular role have been optimal in terms of submission reliability and robustness: a negligible failure rate in submission operations has been registered in a multi submitter environment.

5. WEB-INTERFACE

An automated or at least semi-automatic submission procedure is of the utmost importance in order to speed up the user workflow completion, to keep data and metadata 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 functionalities and procedures to query the stored metadata. 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 functionalities. Obviously, it is strictly bounded 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 subsection. Their content, which mainly consists of web forms, is dynamically generated from the bookkeeping database schema and state in order to include the job-type specific fields.

At the moment, some 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. 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.

A basic authentication and authorization layer, based on a LDAP directory service permits the differentiation of users and grants the access to the corresponding sections of the Web-Interface. Additional authentication mechanisms, such as X509 certificates, proxies, kerberos or shibboleth are under evaluation for further developments.

In the present implementation, the job-initialization interface provides a set of automatically-generated nested scripts. The outermost one (written in PHP) should be executed by a real user through a ssh shell after obtaining a valid Grid credential. This script launches a GANGA session, which takes care of submissions and in turn submit a bash or 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.

The complete automation of the submission process, directly from the Web-interface is under development; it should face the security issues given by the Grid authentication and a solution similar to those found in several Grid-portals [31].

6. RESULTS AND DISCUSSION

The framework conceptual design has been implemented in a prototype serving the use case given by the SuperB project [32, 33].

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 [34]. Both simulations can produce several types of events, depending on a set of parameter given to the executable at runtime, and may use a few files as input.

The INFN-Tier 1 site at CNAF in Bologna, Italy, has been chosen as the central EGI service site; it provides the User Interface to the Grid and deploys the core functionalities of the framework prototype. 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 registering in their Storage Elements the required input files.

The prototype has been specialized and customized to fulfil the application-specific requirements in terms of authentication/authorization mode, job parameters uses, data manipulation operations, job scheduling and monitoring.

The framework prototype has been successfully used in 2010 for intense production cycles of both Full- and Fast Simulation [35]. More than 11 billion simulated events have been produced. Over an effective period of 4 weeks, approximately 180000 jobs were completed with a ~8% failure rate., mainly due to executable errors (0.5%), site misconfigurations (2%), proxy expiration (4%), and temporary overloading of the machine used to receive the data transfers from the remote sites (2%). 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 prototype have been fundamental in achieving the SuperB production cycles goals. The online and offline monitor included with the web-interface keeps the metadata 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 [36] is a complete web portal solution for the development, execution and monitoring of workflows and workflow based parameter studies on various Grid platforms. It hides low-level Grid access mechanisms and is used by other projects as their base framework. ComputER [37] and SHIWA [38], for example, offer Grid access to scientific communities by customizing and developing new features on top of WS-PGrade. Diane [39] is another tool providing an easy access to the Grid infrastructure to application communities. JST [40] is a web-based tool helping in subdividing large applications in independent tasks and execute them on the Grid nodes; several bioinformatics applications use this tool to exploit Grid resources.

Our framework provides an integrated bookkeeping database and 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 metadata and the goal of building a lightweight framework. These characteristics are distinctive of our suite.

7. CONCLUSIONS

The distributed computing paradigm has evolved in the last 10 years under the push of High Energy Physics communities, becoming the computational model in several research and private projects. Nowadays, small and medium size communities are approaching Grid infrastructures asking for tools capable to provide a simplified Grid resources exploitation and monitor.

This work describes the design and implementation of a lightweight framework for automatic grid submission of simulation application to the Grid science infrastructure. It provides a set of services allowing the management of job submissions and monitor. Moreover, it provides a bookkeeping database layer that can be easily customized for every user needs.

With respect to the other projects available to the scientific community, the suite presented in this work is distinctive because it is easy to personalize and to use not only from the job submission service point of view. In fact, despite the lack of some features, related for example to job workflow management typical of data analysis applications or to certificate management, the possibility to customize an integrated bookkeeping database is a new feature of our suite. This simplify the processes of keeping track of jobs results and managing the job parameters metadata.

The framework has been successfully used in a 15 Grid site environment, producing 35TB of simulated data during one month, and accomplishing the requirement of the SuperB experiment community. Furthermore, the suite has engendered the interests of different groups from biology to astrophysics fields, so further developments are in progress to enlarge the use case set and provide more features.

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,” International Conference on e-Science and Grid Computing, p. 151, 2006.

[3] C. Aiftimiei 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 (CHEP 2006), 13-17 Feb. 2006, Mumbai, India, 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, PPAM’07, (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] M. Bona et al., “SuperB: A High-Luminosity Asymmetric e+ e− Super Flavor Factory. Conceptual Design Report,” 2007.

[10] I. Foster, C. Kesselman, and S. Tuecke, “The anatomy of the grid: Enabling scalable virtual organizations,” Int. J. High Perform. Comput. Appl., vol. 15, pp. 200–222, August 2001.

[11] “http://www.egi.eu/.”

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

[13] “http://glite.web.cern.ch/glite/.”

[14] “http://www.opensciencegrid.org.”

[15] “http://www.nordugrid.org.”

[16] “http://www.westgrid.ca.”

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

[18] “https://twiki.cern.ch/twiki/bin/view/lcg/.”

[19] “http://lcg.web.cern.ch/lcg.”

[20] 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.

[21] “http://ganga.web.cern.ch/ganga.”

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

[23] 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.

[24] “http://storm.forge.cnaf.infn.it/.”

[25] “http://www.dcache.org/.”

[26] “https://svnweb.cern.ch/trac/lcgdm/wiki/dpm.”

[27] “http://www.lustre.org.”

[28] “http://hadoop.apache.org/.”

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

[30] “http://grid.pd.infn.it/cream/.”

[31] Kacsuk et al., “Ws-pgrade: Supporting parameter sweep applications in workflows,” in Proceeding of 3rd Workshop on Workflows in Support of Large-Scale Science, in conjunction with SC 2008, 2008.

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

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

[34] R. Andreassen et al., "FastSim: Fast simulation of the SuperB detector," *Nuclear Science Symposium Conference Record (NSS/MIC), 2010 IEEE* , vol., no., pp.322-326, Oct. 30 2010-Nov. 6 2010, doi: 10.1109/NSSMIC.2010.5873773.

[35] D. Brown et al., "First results from the SuperB simulation production system," *Nuclear Science Symposium Conference Record (NSS/MIC), 2010 IEEE*, pp.1185-1189, Oct. 30 2010-Nov. 6 2010, doi: 10.1109/NSSMIC.2010.5873955.

[36] “https://guse.sztaki.hu/liferay-portal-6.0.5/welcome.”

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

[38] “http://www.shiwa-workflow.eu/home.”

[39] “http://it-proj-diane.web.cern.ch/it-proj-diane/index.php.”

[40] 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.