PROCEEDINGS OF SCIENCE

SuperB evaluation of DIRAC Distributed Infrastructure

Giacinto Donvito[*1], **Armando Fella**[2], **Bruno Santeramo**[1,3], **Vincenzo Ciaschini**[4], **Francesco Giacomini**[4], **Alberto Gianoli**[5], **Eleonora Luppi**[5], **Matteo Manzali**[5], **Luca Tomassetti**[5], **Matteo Rama**[6], **Domenico Del Prete**[7], **Guido Russo**[7], **Silvio Pardi**[7], **Roberto Stroili**[8], **Marco Corvo**[8], **Stefano Longo**[8], **Alejandro Perez**[2], **Fabrizio Bianchi**[9]

[1]*INFN Sezione di Bari*
[2]*INFN Sezione di Pisa, University of Ferrara and CNRS*
[3]*University of Bari*
[4]*INFN CNAF*
[5]*INFN Sezione di Ferrara*
[6]*INFN Sezione di Frascati*
[7]*INFN Sezione di Napoli*
[8]*INFN Sezione di Padova*
[9]*INFN Sezione di Torino*
[10]*University of Ferrara*

E-mail: giacinto.donvito@ba.infn.it, armando.fella@pi.infn.it, bruno.santeramo@ba.infn.it, vincenzo.ciaschini@cnaf.infn.it, francesco.giacomini@cnaf.infn.it, alberto.gianoli@fe.infn.it, eleonora.luppi@fe.infn.it, matteo.manzali@fe.infn.it, luca.tomassetti@fe.infn.it, rama@slac.stanford.edu, domenico.delprete@na.infn.it, guido.russo@na.infn.it, silvio.pardi@na.infn.it, roberto.stroili@pd.infn.it, marco.corvo@pd.infn.it, stefano.longo@pd.infn.it, alejandro.perez@pi.infn.it, fabrizio.bianchi@to.infn.it

The recently founded Nicola Cabibbo Lab will host the SuperB experiment: an asymmetric energy e^+e^- collider and detector that will provide a uniquely sensitive probe of New Physics in the flavor sector of the Standard Model. SuperB distributed computing group performed a detailed evaluation of DIRAC Distributed Infrastructure in terms of service capabilities, efficiency and reliability for two main use cases: end user analysis and Montecarlo simulation production. The new DIRAC release 6 has been configured to respond to SuperB requirements all over the majority of DIRAC functionalities. Evaluated DIRAC data management capabilities and the native DIRAC File Catalogue system against LFC (LHC File Catalogue) in terms of features and reliability. Testbed and configuration descriptions has been reported including test and evaluation results, using sites that belongs both to the EGI and OSG grid distributed infrastructure.

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

[*]Speaker.

PoS(EGICF12-EMITC2)030

1. Introduction

SuperB [1] is an international experiment aiming at the construction of a very high luminosity asymmetric e^+e^- flavour factory. SuperB experiment is one of the most important Flagship Project for the Minister of University and Research in Italy M.I.U.R. [2] 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. Electrons and positrons, circulating in a vacuum chamber 1.8 km in circumference, located underground, will be brought into collision using the innovative new "crab waist" design, recently tested at the INFN Frascati Laboratory. The design of SuperB, on the other hand, allows production of a hundred times the existing data sample with power consumption below that of the current generation of facilities. Moreover SuperB will provide a very powerful synchrotron light that will allow to visualize both biological or inorganic structures at a nanometer resolution contributing to solid state physics, biology, nano technologies and biomedicine science fields.

2. Super*B* Distributed Computing infrastructure and requirements

The SuperB experiment needs large samples of Montecarlo 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 [3] and CMS [4] 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 Technical Design Report.

The distributed computing infrastructure, as of January 2012, includes several sites in Europe and North America as reported in Fig. 1. Each site implements a Grid flavor depending on its own affiliation and geographical position, EGI [5] and OSG [6] Grid flavour have been enabled at present time.

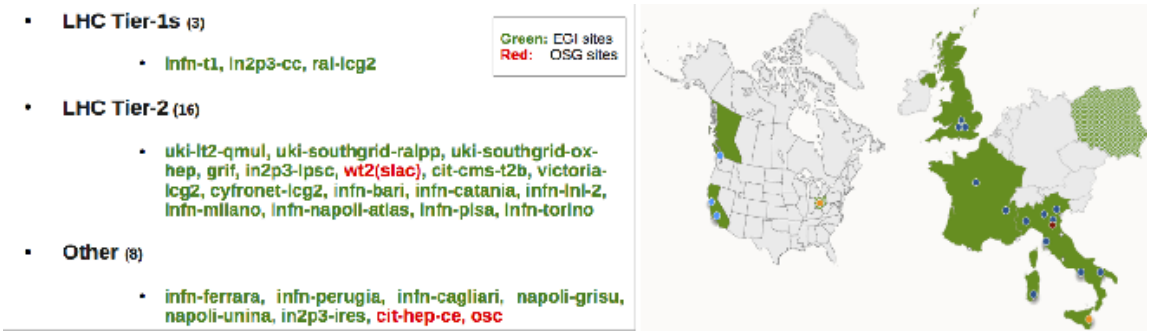

Figure 1: SuperB Grid Sites distribution

The LHC Computing Grid (LCG) architecture [7] was adopted to provide the minimum set of services and applications upon which the Super*B* distributed model could be built.

To give a shape of the computing efforts SuperB experiment will involve, the estimation of network, CPU and disk space resources consumption have been reported:

- 5Gbyte/s of network bandwidth as experiment output
- 1000 PB amount of data collected after 5 year of experiment life
- about 50000 CPU Cores needed for all the activities for the first year of data taking

The three main computing intensive experiment tasks will be modelled to accomplish a fully distributed computational environment: Montecarlo simulation production, data reprocessing and chaotic final user data analysis. In parallel with the Detector Technical Design Report (TDR) definition support, the Computing group works are focusing on a dense R&D program permitting to face technology and resources in 5 years future scenario. It is of worth to cite the R&D plan involving three computing centers located in the South of Italy with the aim of accessing resources as located in a single, large, computing facility.

3. DIRAC overview

DIRAC [8] platform provides a powerful set of instruments to exploit and control a distributed computing infrastructure. DIRAC capabilities include a complete management of workload and data, like simulation production activities, user data analysis use case, data storage access and transfers, file catologuing, massive data transfers, software distribution, monitoring and accounting functionalities. Resources (Computing Elements, Storage Elements, FTS servers, File Catalogues) can be easily configured and managed form a single head portal. DIRAC is structured as several components interacting each other in order to perform a great variety of actions, in order to cope almost all aspects related to a distributed computing environment. The modular structure of DIRAC permits to extend it with custom components, moreover, such a components can be deployed on several servers in order to increase system performance and reliability.

DIRAC natively implements the pilot job paradigm. Pilot jobs can be described as a place holder grid jobs (see fig. 2). Once executed on a worker node, the pilot job performs a detailed check of environment, install a set of useful agents to monitor and execute the DIRAC job wrapper. Pilot job asks to a central task queue for a workload matching resources offered by worker node,then executes the received workload and upload output results on the configured storage area. Thanks to pilot job mechanism the jobs priority can be centrally and easily managed.

Pilot jobs can work in filling mode: the same pilot job can download and execute more then one proper workloads retrieving them, subsequently, from central task queue.

Filling mode helps to minimize the total matching (see fig. 8) and execution (see figures 5, 6, 7) time for a massive jobs submission.

DIRAC can use two kind of file catalogues: LHC File Catalogue (LFC) [9] and DIRAC File Catalogue (DFC) [10], the latter offers some advanced file metadata management, see section 7.

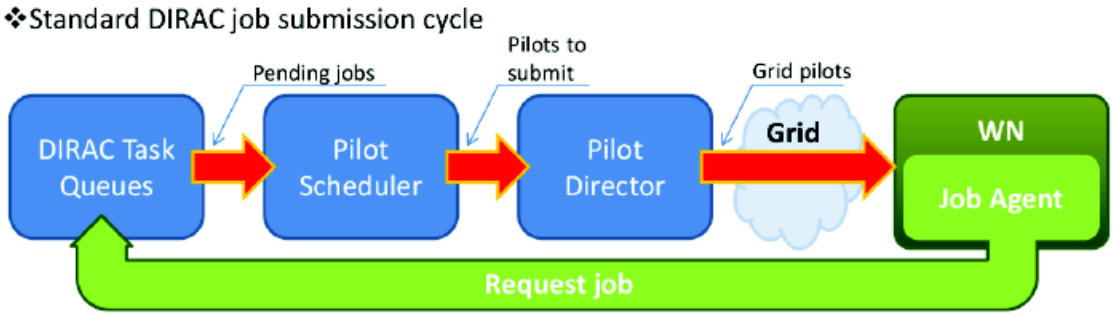

Figure 2: Pilot job workflow

Both EGI and OSG Grid flavor sites can be managed via DIRAC.
DIRAC provides a complete API set useful to write custom application able to interact with DIRAC core layer. Cloud resources can be accessed thanks to the Virtual Machine management Dirac module (VMDIRAC) [11], developed by Belle II Computing Group [12].

Started inside LHCb community, today several VOs and computing center are using DIRAC and some of those are also actively involved in its development. The following table reports the Dirac current involved communities:

Community	Description
Belle II	collaboration/experiment
BEPC	collaboration/experiment
BES	collaboration/experiment
BIOMED @ CREATIS, Lyon	topical community
CC/Lyon	regional community
CESGA	regional community
CTA	collaboration/experiment
GISELA	regional community
GLAST	collaboration/experiment
ILC	collaboration/experiment
IOIT/Hanoi	regional community
LHCb	collaboration/experiment
SuperB	collaboration/experiment

Table 1: DIRAC community

4. Testbed setup

At INFN-T1 and INFN-BARI was configured a testing installation to evaluate DIRAC capabilities from the point of view of Super*B* computing model needs.

Three servers, equipped with a gLite UI and a valid host certificate, were configured for testing purposes:

- bbrbuild01.cr.cnaf.infn.it: 64 bit VM, 2GB ram, 47 GB disk space, Scientific Linux 5.4
- sb-server-04.cr.cnaf.infn.it: 64 bit VM, 2GB ram, 17 GB disk space, Scientific Linux 5.4
- gridtest-05.ba.infn.it: 64 bit VM, 2GB ram, 20 GB disk space, Scientific Linux 5.7

Several resources were rightly configured and used via DIRAC, more in detail were configured 23 EGI and 1 OSG sites, 50 Computing Elements, 19 Storage Elements, 1 DIRAC File catalogue (DFC).

Main goal of testbed was to test job and data management capabilities of DIRAC, so following components were installed:

- Framework core system of DIRAC
- DataManagement storage elements and data management, file catalogue, FTS transfers
- WorkloadManagement jobs management
- Accounting monitoring functionalities
- Configuration computing elements automatic discovery and configuration

First test was conducted using DIRAC v5rX release, but since September 2011 the new v6rX release is under test. Several enhancements were introduced in DIRAC v6rX, several LHCb specific functionalities were removed from the framework in order to offer a well customizable tool for a generic-VO needs.

Thanks to availability of three servers for testing, several installation procedures were performed in order to produce a detailed documentation for the installation and configuration procedures, including configuration files.

Configuring a computing element in DIRAC is really easy because a dedicated agent performs all discovery and configuration actions, so the DIRAC admin user have only to instruct the system with the following two commands:

```
dirac-admin-add-site LCG.INFN-BARI.it INFN-BARI
  cream-ce-2.ba.infn.it cream-ce-1.ba.infn.it
```

```
dirac-admin-allow-site LCG.INFN-BARI.it 'added INFN-BARI'
```

Storage elements and VOMS servers can be easily configured via web interface provided by DIRAC.

DIRAC user management permits a fine grain permission configuration based on groups and every group can be mapped via VOMS attributes.

The two authoritatives Super*B* VOMS servers, INFN-T1 and INFN-PADOVA, were setted up in failover configuration. For each DIRAC server several WMS can be defined: in testbed were configured the wms-multi.grid.cnaf.infn.it WMS.

INFN-T1 has two endpoints in SuperB dedicated storage elements due to different write permission: in DIRAC these two endpoints were configured as two different storage elements at INFN-T1 with groups permission to use them driven by VOMS role attribute.

To enable filling mode a dedicated agent must be installed and configured.

For testing purposes, servers used were adequate, but in order to use DIRAC in production additional servers should be used to setup a distributed environment. During tests up to 10k jobs were submitted and about 2k jobs were running simultaneously, but to avoid database bottleneck a server equipped with 10GB ram is recommended by developers to run MySQL database. Server hosting DIRAC Sandbox Service must have sufficient disk space (the recommended space amount should be defined considering the data model of experiment), server running DIRAC Workload-Management System should have an adequate CPU to decrease time spent in job sanity check.

5. Analysis test

The first use case tested in DIRAC was the distributed analysis. The analysis use case was implemented using a real users application reading Monte Carlo data to perform an analysis on $B^+ \to K^+nn$ and $B_0 \to K_0^*nn$ with $K_0^* \to K^+\pi^-$ channel. The output of the application was a small ntuple file plus postscript file with the resulting plot. This application was compiled using standard ROOT libraries, and the SuperB libraries. DIRAC File Catalogue were exploited to deal with data management and data movement over the grid infrastructure. The accessed dataset was about 200GB of root files divided over about 1260 different files. Each of the files contained in the dataset was quite small in size and this could introduce a big inefficiency in dealing with grid data transfers. In order to avoid this, 10 files was packed together and the packed files was registered into DIRAC File Catalogue.

Full dataset (only packed files) was replicated into two different grid sites (INFN-TIER1-CNAF e INFN-Pisa), and the file catalogue was updated accordingly. This have been done in order to emulate a production environment where the data are distributed on several sites and the jobs submitted by the users could run in more than one site. One analysis job was created for each of the packed files, this means that each job processed one input file of about 1.6Gbyte and produced 3 output files (1 root file plus 2 text files).

In order to test the complete set of features provided by DIRAC was exploited the DIRAC automatic procedure to copy the input and the output files, moreover DIRAC was configured in order to copy the input files using the close SE (the SE of the site where the job is executed). The stage output procedure was configured in order to use up to three different storage elements in case of failure. In fact, DIRAC is able to automatically take care of failures in the stage input or output using a different storage provided by the user. This is completely transparent to the end-users as the files are accessed using the Logical File Name on the catalogue that does not depend on the physical location of the file. Users was not askeed to learn how to copy files from/to grid storage elements, but he/she should only configure in the JDL file, the DIRAC File Catalogue Logical File Name for the input file and the name of the expected output files together with a list of SEs for staging the output.

The *parametric job* DIRAC feature were used to automatically create 126 jobs to analyse the whole dataset. Using this feature in fact the user can submit a unique job to DIRAC, and the server

takes care to produce all the needed files in order to process all the requested files. The user could also ask for the status in a very easy way: asking for one single job, the user will know the status of each sub-job.

Jobs were configured to be submitted at hosting dataset sites, INFN-PISA and INFN-T1: test results shown that at INFN-T1 were executed 60% of jobs and at INFN-PISA the other 40% of jobs.

In order to complete the analysis job the user should use a classical bash script that takes care only of:

- Untar the input file
- Set-up the needed environmental variable in order to use the SuperB and ROOT libraries
- Run the executable itself with the correct parameters

Thanks to the fail-over solutions the final user do not see any failures and all the jobs were successfully completed.

6. MonteCarlo production test

In order to have a measurement of DIRAC performance in a production-like task, a comparison test was performed to compare the current production Super*B* stack, named WebUI [13], and DIRAC.

Test involved 10 sites, submitting 100 jobs for each site, for a total of 1.000 jobs submitted. Each jobs simulated 10.000 events for a running time of about 2-3 hours for each job. Every jobs had 5 input files (in total 3 GB) stored in DFC (for DIRAC) and replicated on more than one storage elements. Output data, in according with SuperB schema, were saved in a separated directory for each job at INFN-T1 and registered in DFC.

Parameters used during test:

- analysis: BtoKNuNu
- generator: B+B-_K+nunu
- geometry: DG_4
- background: MixSuperbBkg
- events: 10,000

After 48 hours test was considered finished, below a summary of results:

Some comment to test results:

- up to 753 jobs simultaneously running
- 80 jobs failed at IN2P3-CC because submitted to short queue (MaxCPUTime = 6 seconds), this problem is solved setting properly CPUTime value in JDL definition

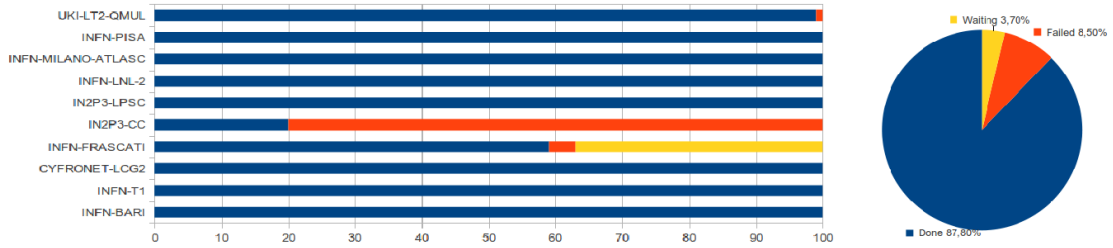

Figure 3: Overall and per site DIRAC test results

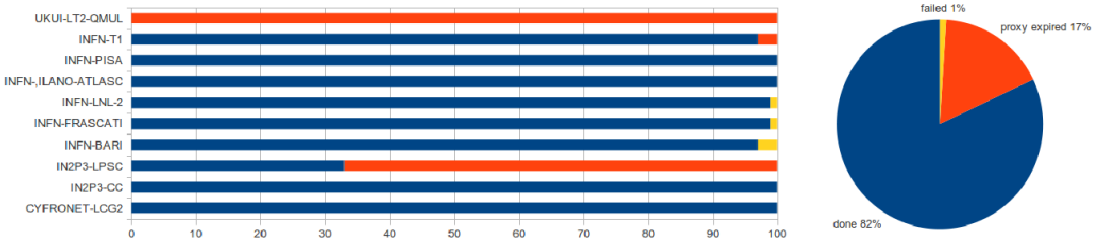

Figure 4: Overall and per site WebUI test results

- All output data were stored at INFN-T1 and registered into DFC
  - LFN:/superbvo.org/test_dir/output_testbed/<job_output_dir>/<file_name>.root
- successfully tested the capabilities of storing output files on a fallback storage element in case of failures of the first one, this features is working fine for input files too

Figures 3 and 4 show a comparison between the current production system used (WebUI) and DIRAC performances.

7. Data management test

One of the most interesting functionality of DIRAC framework is the ability to exploit a grid file catalogue with several advanced features. The usage of a high-level Grid File Catalogue helps to hide the complexity of the grid data management tool. LCG File Catalog (LFC), the most commonly used file catalog in grid, can be considered as a replica catalog offering some metadata functionalities, while DIRAC File Catalogue offers all replica catalog functionalities plus advanced metadata catalog features not available in LFC, like support for user metadata. In order to understand the advanced features that the DIRAC framework provides to the users, the DFC capabilities were tested. For each storage element must be configured the SRM service contact point and the path to use. In DIRAC storage elements are associated to a simple name (INFN-BARI, INFN-PISA, etc.), so the end user could easily refer to each storage element when copying or replicating the data. Standard features like adding, replicating and deleting files were successfully tested. The DFC provides simple interfaces that allow the end user to add a file to the catalogue and to a given Storage Element. Below few examples of adding file, replicating file and replica listing tasks for a given file:

```
dirac-dms-add-file /superbvo.org/user/g/donvito/dir_test/test2
   /lustre/donvito/dirac/bashrc BARI-INFN

dirac-dms-replicate-lfn /superbvo.org/user/g/donvito/dir_test/test2
   FERRARA-INFN

dirac-dms-lfn-replicas /superbvo.org/user/g/donvito/dir_test/test2
'Successful': {'/superbvo.org/user/g/donvito/dir_test/test2':
   {'BARI-INFN': 'srm://storm-se-01.ba.infn.it:8444/srm/manager?
   SFN=/superbvo.org/superbvo.org/user/g/donvito/dir_test/test2'}}
```

The DFC stores also a lot of metadata information about each file uploaded to storage and catalogue as you can see in the following example. This feature is automatic and transparent to the end-user.

```
dirac-dms-lfn-metadata/superbvo.org/user/g/donvito/test3
'Successful': {'/superbvo.org/user/g/donvito/test3': {
'Checksum': 'a0683f04',
'ChecksumType': 'Adler32',
'CreationDate': datetime.datetime(2011, 6, 17, 9, 24, 48),
'FileID': 14L,
'GID': 1,
'GUID': '8531C86D-BD55-A983-3F09-124523587E1F', 'Mode': 775,
'ModificationDate': datetime.datetime(2011, 6, 17, 9, 24, 48), 'Owner':
    'gdonvito',
'OwnerGroup': 'user',
'Size': 967L,
'Status': 1,
'UID': 2}}
```

The DFC gives also the possibility to use an interactive shell in order to browse the file catalogue content and to exploit all the available features:

```
FC:/> cd /superbvo.org/user/g/donvito/dir_test/
FC:/superbvo.org/user/g/donvito/dir_test>ls -l
-r-----rwx 0 gdonvito user
-r-----rwx 0 gdonvito user
967 2011-06-22 17:10:45 test2 967 2011-06-21 09:39:34 test3
```

The last advanced features that tested on DIRAC file catalogue concerns the capability to deal with end user metadata and the file ancestor. Indeed this is a quite powerful approach because the end user could easily *tag* a directory with his/her own arbitrary metadata. This provides to the user the capability to search for files in the catalogue using SQL-like queries on metadata. Below some example showing how this feature works:

```
FC:/superbvo.org/user/g/donvito>metaset dir_test NewMetaInt 3
FC:/superbvo.org/user/g/donvito>metaget dir_test
```

```
!Owner : string
Meta1 : test_dir
!NewMetaInt : 3
FC:/superbvo.org/user/g/donvito>metaset dir_test Owner Giacinto

FC:/superbvo.org/user/g/donvito>find NewMetaInt > 2 Owner=Giacinto
Query: {'Owner': 'Giacinto', 'NewMetaInt': {'>': 2}}
/superbvo.org/user/g/donvito/dir_test/test2
/superbvo.org/user/g/donvito/dir_test/test3
```

8. Filling mode test

Pilot jobs can work in filling mode: in a single time slot, a pilot, once terminated its workload, check remaining time on worker node and ask again to TaskQueue a new workload to process, taking care of time job time consumption.

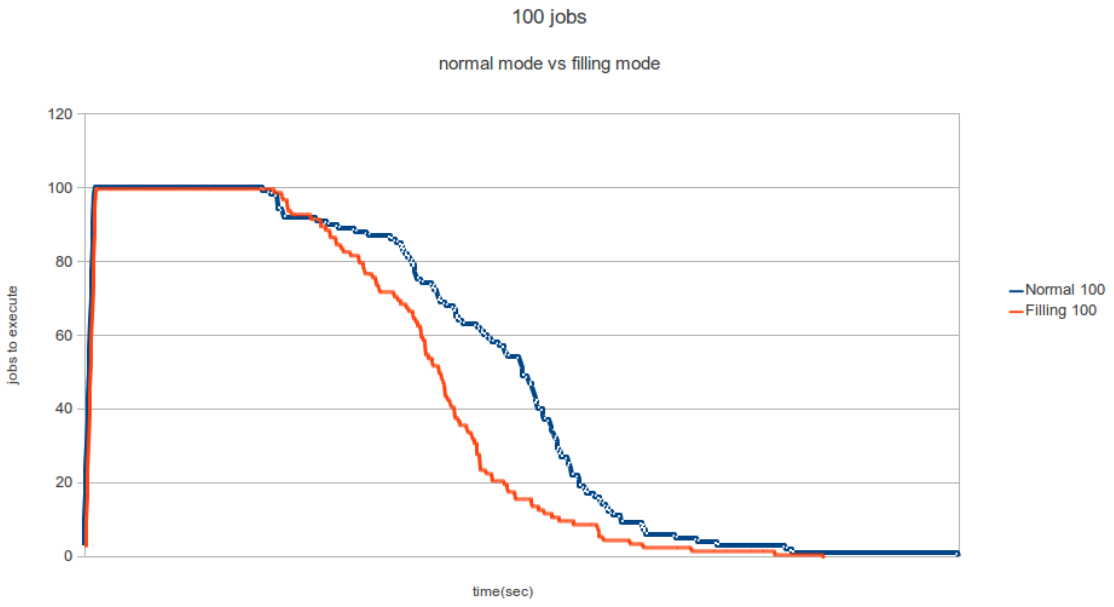

Figure 5: Total time needed to execute 100 jobs: comparison between pilot normal mode and pilot filling mode

To measure gains offered by this feature, a specific test were performed, comparing results obtained enabling and disabling filling mode. Test was performed submitting at same sites 100, 1.000 and 10.000 jobs. Figures 5, 6 and 7 show the total time needed to execute a bulk submission of 100, 1.000 and 10.000 jobs: x-axis time has not the same scale in below graphs.

As can be expected, thanks to filling, can be obtained a significant decrease in total jobs execution time and this effect is more evident increasing the total submitted jobs number.

From a user point of view, system executes one job for each workload, but from a grid perspective, the number of executed jobs is the number of pilot jobs, usually smaller than the number

Figure 6: Total time needed to execute 1.000 jobs: comparison between pilot normal mode and pilot filling mode

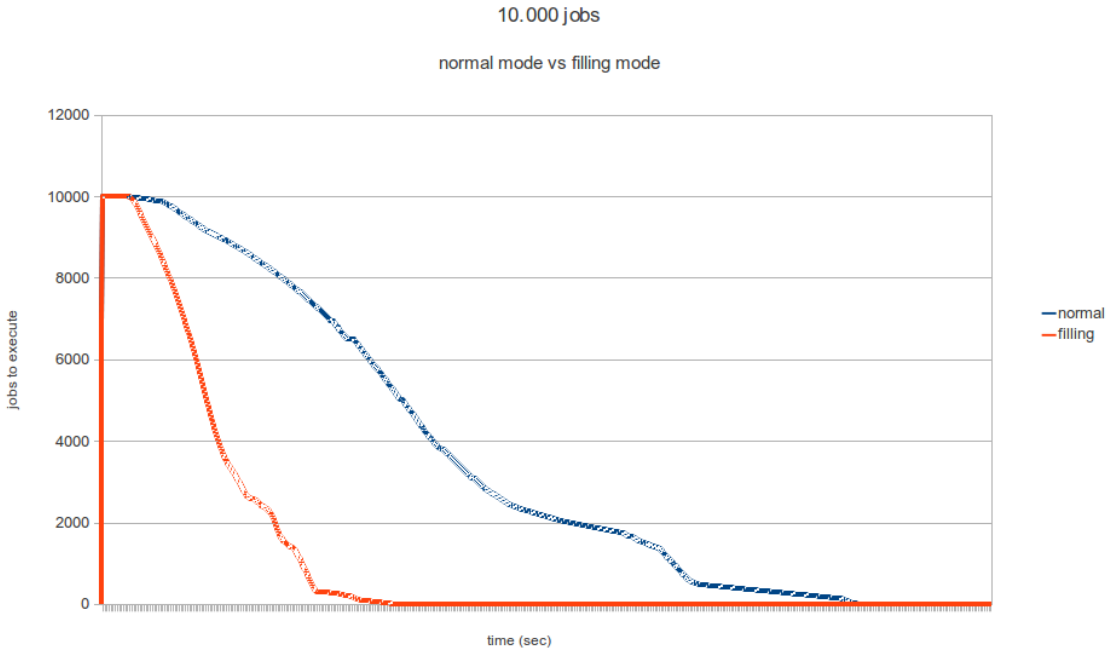

Figure 7: Total time needed to execute 10.000 jobs: comparison between pilot normal mode and pilot filling mode

of workloads. Since a single pilot can execute more than one workload, exploiting only one time the matchmaking operations, filling mode permits a more efficient usage of VO resources.

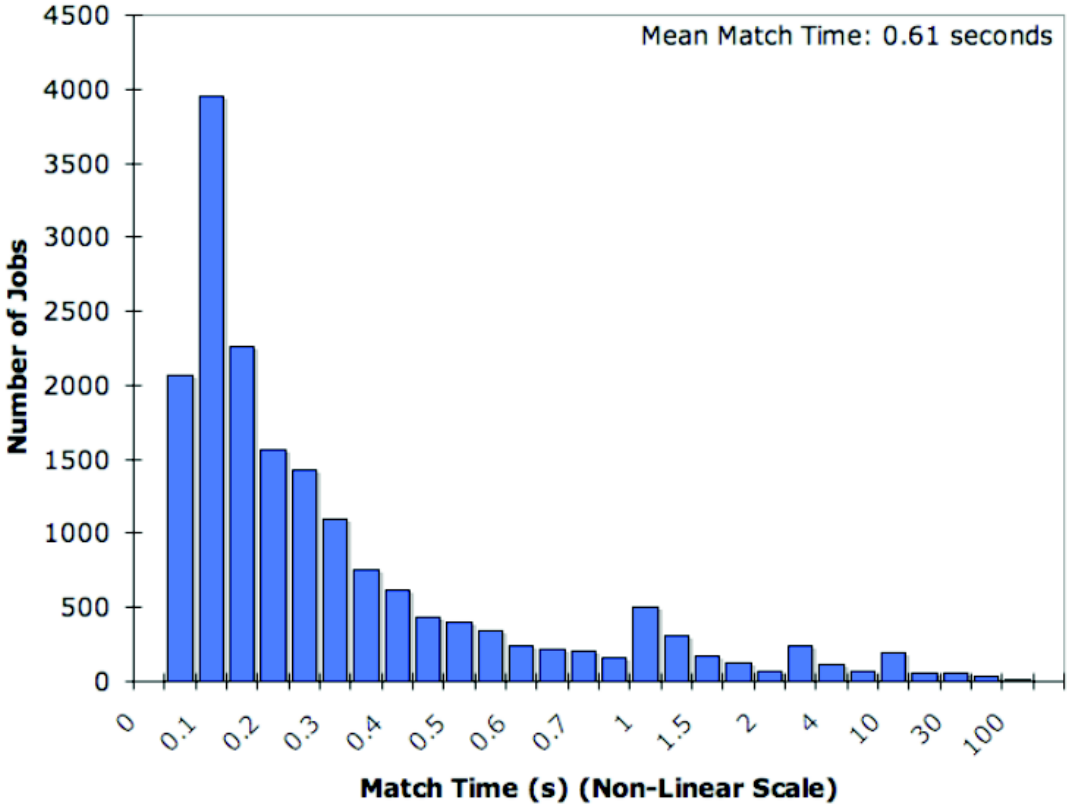

Figure 8: User jobs with many varied requirements present the biggest challenge for the PULL paradigm. DIRAC matcher times for 18K real user jobs submitted between January and August 2007. 92% of jobs are scheduled in under 1 second.

A great advantage of pilot jobs is the smaller time spent in matching computing elements with job requirements, as shown by this measurements performed by DIRAC team (see figure 8)

9. Conclusions and future works

At present time, the SuperB experiment is in an intense R&D phase, that will provide the needed experience and information to write a Computing Technical Design Report, that is expected to be released around middle of 2013. The activities carried on within the SuperB experiment in this phase are carried on the direction of developing a new generation framework and in evaluating already available solutions that could be of help in order to fulfil SuperB use-cases. In this phase of the life of a new collaboration it is important to evaluate all the available solutions as there are at least the LHC experiments that has similar use-cases and that are producing several good solutions for many of the problems that Super*B* is facing now. The job submission and data management are surely two of the areas where already available software like DIRAC could be of help. Moreover the DIRAC framework is used at present time by LHCb experiment and the roadmap of the tool is surely interesting for a Virtual Organization like SuperB that will use a geographically distributed grid infrastructure in order to solve the computational problems of an HEP Experiment that will produce a huge amount of data in the next years.

Giving those considerations, the work carried on, already give us a good feeling on the capabilities of the DIRAC framework, and we are planning to test all the features that it makes available, in order to have the opportunity to understand if this tool could solve the use cases of the Super*B* community.

One of the most important future test will be the DIRAC capability to handle massive data transfer among grid sites. In this case for example DIRAC will be compared to PhEDEx that is the CMS tool for massive data transfer.

References

[1] The Super*B* Collaboration, *SuperB Progress Report, Detector*, http://arxiv.org/abs/1007.4241v1

[2] http://www.istruzione.it/web/hub/home

[3] The ATLAS Collaboration, ATLAS Detector and Physics Performance Technical Design Report, http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/TDR/access.html.

[4] The CMS Collaboration, CMS Detector Technical Design Report, http://cmsdoc.cern.ch/cms/cpt/tdr/.

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

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

[7] http://lcg.web.cern.ch/LCG

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

[9] https://twiki.cern.ch/twiki/bin/view/LCG/LfcAdminGuide

[10] https://github.com/DIRACGrid/DIRAC/wiki/DataManagementAdvanced

[11] Diaz, R. G., Ramo, A. C., Aguero, A. C., Fifield, T., & Sevior, M. (2011). Belle-DIRAC Setup for Using Amazon Elastic Compute Cloud. J Grid Comput, 9(1), 65-79. Springer-Verlag New York, Inc. doi:10.1007/s10723-010-9175-7

[12] Belle II WWW pages, http://superb.kek.jp/

[13] SuperB Simulation Production System http://indico.cern.ch/contributionDisplay.py?contribId=301&confId=149557