LSST:UK Consortium

LSST:UK Science Centre

Technical Report on DC2 UK Grid Runs

Project Acronym	LUSC-A
Project Title	UK Involvement in the Large Synoptic Survey Telescope
Document Number	LUSC-A-13

Submission date	24/OCT/19
Version	1.0
Status	Published
Author(s) inc. institutional affiliation	James Perry, University of Edinburgh
Reviewer(s)	Bob Mann (Edinburgh) George Beckett (Edinburgh)

Dissemination level	
Public	

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	16/SEP/19	Initial draft version	James Perry
0.2	24/OCT/19	Minor changes to address comments from Bob Mann	James Perry, Bob Mann
1.0	14/NOV/19	Document finalised for publication	George Beckett

Table of Contents

Index of Figures

Index of Tables

1 Executive Summary

This report describes the ImSim[1] runs performed on the UK computational grid during 2018 and 2019 for DESC Data Challenge 2 as part of the agreed UK contribution to DESC operations. Run 1.2i was performed on grid resources in 2018, and part of Run 2.1i in 2019.

As no-one on the project had recent experience of accessing grid resources, there was quite a steep learning curve before meaningful work could start. For authentication purposes, it was necessary to obtain an X509 user certificate from the UK National Grid Service certificate authority [2] and then register this with the LSST Virtual Organisation. Jobs could then be submitted to GridPP resources via a virtual machine image provided by CERN. ImSim and its dependencies were deployed on CVMFS [3], a distributed virtual file system that allows software repositories to be mounted remotely on compute resources.

For Run 1.2i, jobs were submitted to the grid via the Ganga job management system [4] recommended by GridPP. A lot of experimentation was required to find job parameters that worked well on the grid. The Python scripting interface provided by Ganga was used to automate the submission process as much as possible, however it remained quite a labour-intensive process with a significant amount of manual intervention required. Input data was fetched from grid storage elements (to which it was copied from NERSC), and output data was similarly written directly to grid storage elements.

A total of 2,001 visits were to be simulated, each visit being split into 52 grid jobs (with each job processing 4 image sensors), although this was later reduced to 35 jobs (with each job processing 6 image sensors). The total number of jobs to be run was therefore in the range 70,000-100,000. As the runs progressed, the Ganga user interface started to become a bottleneck, limiting the rate of completion to around 3,000 jobs per day. Ultimately, around 50% of the jobs were completed on GridPP, with the remainder processed by DESC on Theta, a traditional HPC resource in the USA.

Some changes were made to the grid setup for Run 2.1i. This time, a new job submission system was developed around the underlying DIRAC software [5], bypassing Ganga and allowing for higher levels of throughput and automation. The version of ImSim installed on CVMFS was also updated to a later version. IN2P3 provided access to their grid nodes at LAPP and Lyon, supplementing the UK grid infrastructure with further compute resources.

It was agreed that year 6 of the Run 2.1i schedule would be run on European grid resources, with the possibility of continuing onwards to year 7 and beyond. Year 6 consisted of around 4,000 visits. This time each visit required 52 grid jobs, each processing 4 image sensors. Once again, input data was transferred from NERSC to UK grid storage elements (with assistance from IN2P3) and output data was written to grid storage.

With the exception of periods when outages caused delays, job throughput was much better than it had been for Run 1.2i: 15,000 jobs completing per day was typical. Year 6 had been nearly completed on the grid and year 7 had already been started when concerns were raised that the output from the grid runs did not match the output from the runs happening at NERSC. It was later discovered that there were problems with the validity of all the output data (from both the grid runs and the NERSC runs), and the programme was paused while this was investigated.

Although the grid contribution to Run 2.1i was more successful than that for Run 1.2i, concerns were raised that the level of effort required to properly make use of the grid resources was beyond what LSST:UK could provide, and that this was having an adverse effect on the job throughput. It was agreed that, in the first instance, LSST:UK would attempt to address this by engaging more fully with GridPP's operations team, and also by introducing some sort of automated monitoring to try to detect problems as early as possible.

2 Introduction

2.1 Purpose

This document describes LSST:UK's usage of UK and French grid computing resources to contribute to DESC's Data Challenge 2. This work was part of the agreed UK contribution to DESC operations.

2.2 Glossary of Acronyms

CVMFS – CERN Virtual Machine File System

DESC - Dark Energy Science Collaboration

DIRAC - Distributed Infrastructure with Remote Agent Control

GGUS – Global Grid User Support

GridPP – UK Grid for Particle Physics

LFN – Logical File Name

LSST – Large Synoptic Survey Telescope

NERSC – National Energy Research Scientific Computing Centre

PSF – Point Spread Function

SE – Storage Element

VO – Virtual Organisation

VOMS – Virtual Organisation Membership Service

3 Run 1.2i

LSST:UK began contributing to DESC's Run 1.2i in April 2018. Our contribution involved simulating 2,001 visits using the ImSim software. These visits were split across the WFD and uDDF surveys, each of which consisted of 6 bands (`i`, `g`, `u`, `z`, `y` and `r`). For each visit, 205 separate image sensors had to be simulated.

3.1 Preparation

3.1.1 Authentication

GridPP uses X509 certificates and virtual organisations to implement authorisation and user management. It was therefore necessary to acquire a suitable user certificate from the UK National Grid Service certificate authority, and then join an appropriate virtual organisation before grid runs could commence. The VO used by LSST was hosted on the VOMS server at Fermilab at this time, but was later moved to Stanford.

3.1.2 Accessing GridPP

A properly configured suite of client software is required in order to submit jobs to the grid. The simplest way to obtain access to this was to download a virtual machine image provided by CERN, with the appropriate software already installed and configured. I encountered various problems authenticating to CERN so that I could download the VM, and logging into it once I had it installed, but these were overcome with the help of experienced grid personnel. I was then able to submit simple test jobs to the grid via Ganga, as well as copying data to and from grid storage elements using DIRAC tools.

3.1.3 Deploying ImSim

On GridPP, software is deployed on CVMFS, a file system that allows software to be installed in a repository. The software then becomes visible on all machines that have the repository mounted. This allows software to be made available to all the worker nodes on the grid, ensuring that jobs can run on any of them. It is possible to avoid installing to CVMFS by having each grid job simply fetch and build its software from the Internet on start-up, however this was unsuitable for ImSim due to the large size and long build time of its dependencies.

There were already two CVMFS repositories containing LSST-related software: `/cvmfs/lsst.opensciencegrid.org` and `/cvmfs/lsst.in2p3.fr`. The first of these only contained fairly outdated versions of the base LSST software. It did not contain ImSim nor the LSST simulation stack on which ImSim depends. The second apparently did contain an up-to-date simulation stack but, due to a firewall issue or similar, it could not be mounted on many of the UK grid machines. (It was not visible on my client VM, for example). Due to the unavailability of the software we required on existing CVMFS repositories, and the difficulty of getting access to them, we were instead given access to the `/cvmfs/gridpp.egi.eu` repository, and I deployed the software in there.

At the time, most of the compute nodes on the grid were running Scientific Linux 6.9. For maximum compatibility I therefore built ImSim and its dependencies in a SL6.9 virtual machine and then copied the resulting binaries to the CVMFS repository, allowing us to run ImSim on GridPP.

Updating the software (sometimes required due to bug fixes) was quite a long-winded process, both due to the long build times for the LSST software, and also the fact that it can take several hours for changes made in CVMFS to fully propagate to all of the grid nodes. This would therefore not be a suitable solution for deploying experimental software that changes frequently.

3.2 Running ImSim

3.2.1 ImSim Jobs

Once ImSim and its dependencies had been deployed, it was possible to run small ImSim test jobs on the grid successfully. Running larger jobs, such as the Run 1.2i jobs, required some configuration changes to ensure that the memory and CPU requirements of the jobs were accommodated. These jobs required around 11GB of memory, and most grid resources provided only around 2GB per CPU core, so it was necessary to request 8 cores. This could be done by setting the `8Processors` tag in the DIRAC job scripts generated by Ganga. It was also necessary to request authorisation for the LSST virtual organisation to run multicore jobs, a privilege not granted to VOs by default.

Although it was necessary to request 8 CPU cores to allocate sufficient memory for ImSim, it was not possible to actually use all of these cores, as the memory would once again be exhausted (some of ImSim's memory footprint is shared between processes while some is allocated per process). After some experimentation it was found that we could safely use 4 of the 8 cores, allowing each job to simulate 4 image sensors in parallel. Hence 52 grid jobs were required to process all 205 sensors making up a visit. Later on, after efficiency improvements were made to ImSim, the number of parallel processes was increased to 6, reducing the job count per visit to 35.

Input data for ImSim took the form of tar files containing instance catalogues – lists of objects that should appear in the simulated images. These files were uploaded to a grid storage element in advance of the jobs running, using DIRAC data management tools. They could then be referenced by logical filename in the job script, causing the grid middleware to automatically download them to the assigned worker node before the job commenced.

Unfortunately, due to a bug in Ganga, specifying the LFN not only had the desired effect of downloading the data to the worker node, but also constrained the job to only run at sites where a copy of the data was present locally. This limited the sites that our jobs were able to run at, though the problem could be mitigated somewhat by replicating the instance catalogues to multiple storage elements so that more sites would meet the data locality restriction. This bug was later fixed in an update to Ganga.

3.2.2 Automation

Initially, each visit required 52 grid jobs, each (except the last) processing 4 sensors. This meant that in total over 100,000 jobs had to be run on the grid, and even once the process count was increased to 6, this still required 70,000 jobs. Clearly this was too many to handle manually and required some level of automation.

Ganga allows Python scripts to be run using its `execfile` command, and these scripts have access to job objects and other Ganga functionality via a Python API. My first attempt at automation involved writing:

- A script to submit all of the jobs required for a visit in one go.
- A script to monitor the state of the visits currently running.
- A script to submit a single job (rather than an entire visit), used for rerunning jobs that had failed.

I also wrote Bash scripts to automate the process of uploading instance catalogues to the grid from NERSC, and of downloading the output from a visit and copying it to NERSC once the visit was complete.

This was a great improvement on having to manage each individual job manually, but it still required quite a lot of manual intervention. Specifically, it was necessary to manually keep track of which visits were currently running, which jobs within each visit required rerunning, and which visits had had their output data copied to NERSC, as well as manually deleting jobs for visits that had completed. After a few weeks of using this system it became clear that something better was required.

I wrote a second set of scripts, this time keeping track of the various jobs automatically in a set of text files. Although it was still necessary to launch new jobs manually, and to periodically run a script within Ganga to purge completed jobs and rerun failed ones, as well as a shell script to copy the output from completed jobs to NERSC, everything else was at least semi-automated: the system would keep track of what state each job was in, relieving the operator from having to do this.

The large number of possible failure modes made the jobs more difficult to manage. A failed job could mean a job submission system failure, a worker node failure, a failure to download the input data, a failure to upload the output data, a job that exceeded its allotted time or memory, etc. Some of these failures could be detected and overcome automatically, but others still required manual intervention. In the early runs large numbers of failed jobs appeared because certain sensor-visits simply did not create any output - an expected outcome, but one which caused the upload of the output file to fail, and the job would subsequently be marked as failed. This was avoided later on by creating a "dummy" output file in such cases, as well as printing a message to the job's log file signalling to the script that copies output data to NERSC that there was no need to do anything for this job.

The scripts written to automate Run 1.2i can be found here: https://github.com/LSSTDESC/imsim-scripts-dc2/tree/master/gridpp.

3.3 Performance

Most jobs took approximately 2 hours to run on a GridPP worker node, however there was some variation either side of this (see Figure 1). The run time appears to be quite dependent on the contents of the instance catalogue. Jobs that produce no output on any of their sensors, as mentioned above, generally run very fast, only taking a few minutes, while some jobs take much longer than 2 hours. A handful even take longer than the 48 hour time limit on the grid and thus fail.

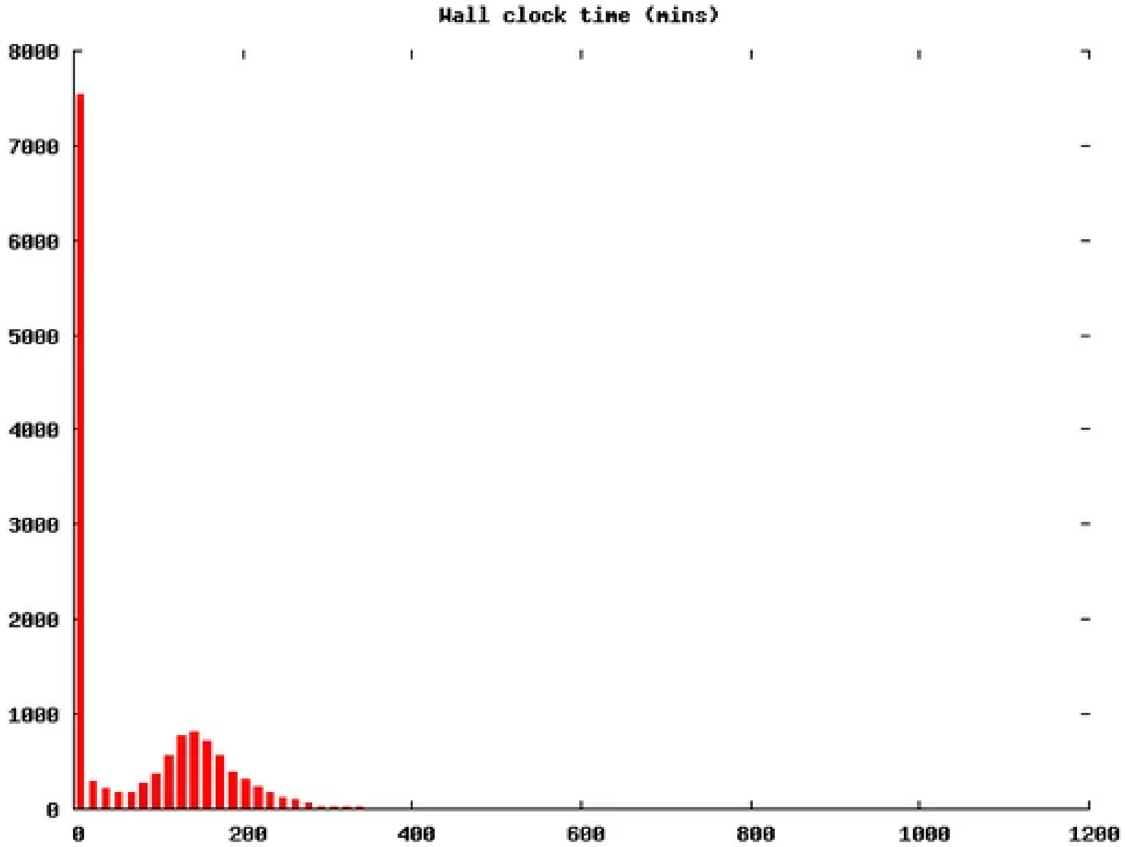

Figure 1 Run time of ImSim jobs on grid

Initially only a small number of jobs were being completed each day. This was partly due to the Ganga bug restricting where jobs were able to run and partly due to the amount of manual intervention required for each visit. However, with the Ganga bug fixed and the automation improved, this throughput increased markedly towards the end of the runs: up to 3,000 jobs were running to completion each day, with around 400 running simultaneously on the grid.

The performance of Ganga itself started to become a limiting factor at this point. Its user interface begins to noticeably slow down with a few thousand jobs in the system, causing many operations to take much longer. Additionally, the Ganga threads which monitor the status of jobs in the background sometimes stop running, either because the user's proxy expired or at random for no obvious reason. If this happens and they are not restarted promptly (generally by restarting Ganga), it can then take a long time (often several hours) for Ganga to "catch up" with the actual state of the jobs on the grid.

As a result of this limited throughput, only about half of Run 1.2i was ultimately completed using the grid. The other half was run on the Theta HPC system in the USA.

4 Run 2.1i

Following our contribution to Run 1.2i, it was decided that in 2019 we would contribute UK grid resources to Run 2.1i. After discussions with DESC it was agreed that we would initially process year 6 of the data challenge schedule, consisting of just over 4,000 visits, with the possibility of moving on to process further years if the first was a success. Other years were simultaneously being simulated at NERSC.

4.1 Configuration

Much of the configuration used for Run 2.1i was the same as that described above for Run 1.2i, so only the differences will be highlighted here. A new version of ImSim (v0.5) was deployed on the grid via CVMFS and tested. At this time both Scientific Linux 6 and 7 were prevalent on the grid worker nodes, however no problems were encountered running the same build of the software on both platforms.

For Run 1.2i, it had been necessary to allocate 8 CPU cores to each ImSim job in order to gain access to sufficient memory, but only 6 of those cores could actually be used without once again going over the memory limit. We were able to improve on this for Run 2.1i; recent enhancements to ImSim and upgrades to the amount of memory available on typical grid nodes meant that we were instead able to allocate 4 CPU cores per job and use all 4. This significantly increased the pool of grid resources that we were able to access, since there are typically far more 4 core job slots available on the UK grid than 8 core slots.

As before, the ImSim jobs would fetch their input data (in the form of instance catalogues) from grid storage elements, and would also write their output (up to 4 FITS files per job) to grid storage elements.

4.2 Contribution from France

In addition to utilising UK grid resources as before, we also made use of French resources for Run 2.1i. IN2P3 provided us with access to their grid nodes at LAPP and Lyon. As these nodes were already well integrated with GridPP and provided the same interfaces, there was little specific work required on our side to allow jobs to run there - some jobs would automatically be scheduled in France if resources were available. However, due to our CVMFS repository not being available at Lyon, it was necessary to use IN2P3's repository instead. This contained the required version of the LSST simulation stack, but not ImSim itself. Our job launch script was modified to detect which CVMFS repository was in use and, if necessary, to download and extract a tar file containing ImSim from a local storage element. Otherwise, the setup was unchanged.

4.3 Job Submission Process

The Ganga interface used to submit the Run 1.2i jobs presented scalability problems as discussed above, and required a high level of manual intervention to keep the jobs running. It was suggested to us that bypassing Ganga and submitting jobs directly to the underlying DIRAC system might be more suitable for our workflow and, after successful testing, this was the approach chosen for Run 2.1i.

DIRAC provides a Python API for submitting jobs. I wrote a Python script to manage the ImSim jobs, keeping track of which ones were running and which had completed as well as automatically resubmitting jobs that failed, and submitting the next batch of jobs when the total number in the system fell below a certain threshold. This proved to be a much more scalable and lower maintenance system than our previous Ganga-based workflow.

The scripts used to manage the Run 2.1i jobs can be found here: https://github.com/jamesp-epcc/run2.1i-scripts.

4.4 Progress

Initially the rate of job completion on the grid was relatively modest, of the order of a few thousand jobs per day, similar to what we had seen for Run 1.2i. However, we later discovered (with some help from GridPP system administrators) that our job throughput was improved if we had more jobs waiting to run. After increasing the total number of jobs in the system to 50,000 we started to see around 15,000 jobs completing per day, equating to around 300 visits per day, with a few thousand jobs typically running at a time.

This rate of progress was sustained reasonably successfully from late May into June, except when temporary problems caused the jobs to stall. Fortunately the DIRAC-based submission system proved quite capable of handling this number of jobs. By late June over 280,000 jobs had completed on the grid, representing over 5,000 visits. The overall distribution of jobs to sites, as of late June, is shown in Table 1.

Grid Site	Number of Jobs
LCG.RAL-LCG2.uk	138255
LCG.UKI-LT2-QMUL.uk	45747
LCG.UKI-NORTHGRID-MAN-HEP.uk	45069
LCG.UKI-SOUTHGRID-RALPP.uk	17969
LCG.UKI-SOUTHGRID-BRIS-HEP.uk	16838
LCG.UKI-NORTHGRID-LIV-HEP.uk	15251
LCG.IN2P3-LAPP.fr	8947
LCG.UKI-LT2-Brunel.uk	8299
LCG.IN2P3-CC.fr	5779
LCG.UKI-SCOTGRID-ECDF.uk	4691
LCG.UKI-LT2-IC-HEP.uk	4558
LCG.UKI-NORTHGRID-LANCS-HEP.uk	3181

Table 1 Distribution of Run 2.1i jobs to grid sites

Although our original intention was to complete the visits comprising year 6 of the survey first, we were initially only given the starting point of year 6 in the sequence of instance catalogues and not the end. Therefore, we started from the beginning of year 6 and worked through the visits as quickly as the grid would allow us to.

When we checked our progress against the visit numbers in late June, we discovered that although we had only completed roughly 63% of year 6, we had also already run almost half of year 7. Due to various factors such as differences in execution time depending on the input data, jobs on the grid not necessarily running in the order they are submitted, failed jobs having to be rerun, and the input data becoming available out of order, it was not easy to ensure the completion of year 6 before moving onto subsequent years.

4.5 Problems Encountered

Although the Run 2.1i work went more smoothly in general than Run 1.2i, there were still various problems, many of them related to data.

Initially, the same simple script that had been used to copy the Run 1.2i input data to the grid was also used for Run 2.1i. This script simply copied each instance catalogue from NERSC to a local machine using SCP, then uploaded it to the grid via `dirac-dms-add-file`. This had been sufficient for Run 1.2i, but the much larger instance catalogues and greater job throughput of Run 2.1i meant that this step quickly became a major bottleneck. This was solved by instead using the File Transfer Service at CERN to copy files directly from NERSC to grid storage elements, with the help of Bastien Gounon from IN2P3.

Access to some SEs was also made difficult by a VOMS problem. A change to the VOMS server configuration resulted in the `dirac-proxy-init` command no longer generating a valid VOMS extension, preventing authentication with many of the SEs. However, because some SEs erroneously accepted proxies without the extension, this led me to believe that the problem lay with the other SEs rather than with VOMS, and it took some time for the true cause to become apparent. Another (unannounced) VOMS configuration change a few months later also caused problems, preventing LSST from using several grid sites until the certificate settings at those sites had been updated.

Another problem occurred when the disk on the storage element at Imperial College filled up with LSST files. We had to move large amounts of data to an alternative SE at Lancaster at relatively short notice. I had assumed that the SE would stop accepting data once we reached our quota, however this was not the case and it allowed us to continue writing files until the disk was completely full.

On two occasions Run 2.1i was held up by grid outages. Firstly, the storage element at Manchester was offline for a few days for maintenance. A large portion of our input data was stored on this SE so many jobs were unable to proceed until it was once again available. This could have been mitigated by creating a second replica of the data elsewhere, however the other storage and data transfer problems would have made this quite challenging to do at the time.

The second outage was caused by the DIRAC server at Imperial College being moved to a new location. It was not possible to run DIRAC jobs on the grid during this outage. Although the down time itself was relatively short, it took another few days before job throughput returned to its previous high level afterwards.

Later in the process, there were concerns about the validity of the data being produced on the grid. To check this, one of the visits already processed at NERSC was also run on the grid so that the output of the two could be compared. This revealed that the grid runs did not have the correct (atmospheric) PSF enabled in ImSim. Although correctness testing had been performed prior to commencing Run 2.1i, this mismatch was not noticed at that time.

Our grid workflow was then modified to use the same ImSim launcher script that was in use at NERSC, to ensure that identical settings would be used in both environments in future. However, before production runs could resume, a further validity problem was discovered, this time affecting all of the Run 2.1i data, both from NERSC and from the grid. At the time of writing, discussions are still ongoing within DESC about how best to resolve this, so Run 2.1i remains paused.

4.6 LSST:UK Engagement with GridPP

Many of the problems with Run 2.1i could potentially have been avoided with better engagement between LSST:UK and GridPP. For example, if VOMS server changes had been communicated to GridPP in a timely manner, the site configurations could have been updated to reflect the changes much more quickly. Similarly, if LSST:UK had been aware of the upcoming GridPP outages sooner, some of the related issues might have been mitigated.

The effort available within LSST:UK to manage the grid runs falls some way short of what would be ideal – GridPP recommends that virtual organisations have a computing team consisting of a minimum of 2-3 FTEs, while LSST:UK has had less than 50% of James Perry’s time dedicated to this task, supplemented with occasional help from others. However, some positive steps have been identified that would likely improve the situation:

1. LSST:UK will attempt to engage more with GridPP’s processes, attending the weekly operations meeting whenever possible, and using the GGUS ticketing system to report issues with the grid.
2. LSST:UK will deploy a system to monitor the state of GridPP infrastructure by regularly submitting automated test jobs and checking that they complete successfully. This will hopefully result in infrastructure problems being noticed and reported more promptly than they have been in the past.
3. Some of Teng Li’s time will be assigned to LSST grid work. Teng has extensive grid experience which will be very useful for troubleshooting any problems with the LSST jobs.

5 References

[1] https://github.com/LSSTDESC/imSim
[2] http://www.ngs.ac.uk/ukca/
[3] https://cernvm.cern.ch/portal/filesystem
[4] D C Vanderster et al, “Ganga: User-friendly Grid job submission and management tool for LHC and beyond”, Journal of Physics: Conference Series, IOP Publishing, volume 219, no. 7, 2010
[5] http://diracgrid.org/