Journal of Physics: Conference Series **396** (2012) 032023 doi:10.1088/1742-6596/396/3/032023

The LHCb Data Management System

J.P.Baud[1], Ph.Charpentier[1], K.Ciba[1], R.Graciani[2], E.Lanciotti[1], Z.Màthè[1,*], D.Remenska[3], R.Santana[4]

[1] CERN, European Organization for Nuclear Research, Switzerland

[2] University of Barcelona, Spain

[3] NIKHEF, Amsterdam, The Netherlands

[4] CBPF, Rio de Janeiro, Brazil

Philippe.Charpentier@cern.ch

Abstract. The LHCb Data Management System is based on the DIRAC Grid Community Solution. LHCbDirac provides extensions to the basic DMS such as a Bookkeeping System. Datasets are defined as sets of files corresponding to a given query in the Bookkeeping system. Datasets can be manipulated by CLI tools as well as by automatic transformations (removal, replication, processing). A dynamic handling of dataset replication is performed, based on disk space usage at the sites and dataset popularity. For custodial storage, an on-demand recall of files from tape is performed, driven by the requests of the jobs, including disk cache handling. We shall describe the tools that are available for Data Management, from handling of large datasets to basic tools for users as well as for monitoring the dynamic behavior of LHCb Storage capacity.

1. Introduction

The LHCb experiment [1] is one of the four large experiments at the LHC collider at CERN, Geneva. The experimental setup consists of a forward magnetic spectrometer, with a very precise tracking system, a set of calorimeters, two Ring Imaging Cherenkov detectors and a muon detection system.

The aim of the experiment is to study the matter-antimatter asymmetry (CP violation) using two types of elusive particles: those contain b-quarks and those containing c-quarks. It also studies very rare decay modes of these particles in order to test with high precision the Standard Model of Particle Physics and possibly obtain hints for new physics.

Such a high precision experiments requires to collect very large amounts of data, even though the number of events used for physics analysis is only a small fraction of the collected events.

The present paper describes the LHCb Data Management System (DMS) and its operational usage during the first year of LHCb data taking.

2. The LHCb Computing Model in a nutshell

LHCb uses a very sophisticated two-levels trigger system for selecting interesting collisions: a first level (L0) is a customized piece of hardware that selects at maximum 10^6 collisions per second (out of

[*] Also University of Dublin, Ireland

the up to 35 10^6 beam crossings per second in the collider), whose data is read out into a huge farm of 20,000 CPU cores running the High Level Trigger (HLT) application. Out of the several 10's billions proton collisions every second, from 2 to 5 thousands are finally retained and recorded. The LHCb Computing model [2] defines the processing chain or real data as well as simulated data and their distribution on the Grid.

2.1. RAW data distribution

The LHCb online system records the information from the selected collisions (RAW data) as streamed files. Each event has a size of about 60 kBytes, which for a trigger rate of 5,000 event/s represents a throughput of 300 MB/s.

The data recorded by the Online system (RAW files) are then transferred to the Tier0 mass storage system and registered in the LHCb DMS's catalogs. After successful migration to custodial storage and verification of the integrity of the files, they are replicated to one of the 6 LHCb Tier1s on custodial storage as well.

2.2. Data Reconstruction

After replication of the RAW data, reconstruction jobs are automatically submitted for being executed at CERN and on the Tier1s, producing as output a ROOT file containing all reconstructed objects for each event (SDST, about 50 kB per event). The SDST is uploaded to custodial storage at the site where it is produced.

2.3. Data stripping

For their physics analysis, LHCb physicists need to apply a very severe selection on the reconstructed events. The selected events are split into broad classes (streams) corresponding to different types of physics analysis. This is achieved by centrally managed production jobs running a *stripping* application, whose output dataset is a set of streams of selected events that can be in two formats: DST containing all of the event's information (120 to 150 kB) or MDST containing a reduced piece of information (10 kB) relevant for a particular physics analysis. The overall streamed dataset size is about 10% of the original RAW or SDST size.

The DSTs and MDSTs are then merged into files of 5 GB and replicated for physics analysis. The replication policy (data placement) can be determined depending on the space availability. The current baseline policy is the following:

- 3 replicas on T0D1 storage at selected Tier1s and one at CERN.
- 2 archive replicas on T1D0 custodial storage, one at CERN and one at a Tier1.

2.4. Physics analysis

Physics analysis is performed at CERN and Tier1s using the DST and MDST streams that have been replicated on T0D1 as described above. Job brokering is targeting sites where the datasets are accessible form the analysis application using protocol access to the storage (xroot, [gsi]dcap, file…).

2.5. Monte-Carlo simulation

Precision physics also requires large samples of simulated data. Monte-Carlo simulation is highly CPU consuming (typically 100s per event). Simulation jobs are brokered on all available Grid sites and upload their data to designated Tier1s scratch space on T0D1. The resulting files are then merged into 5GB files, in a DST format similar to that of the real data (with additional information on the simulated event). Simulated events have a typical size of 400 kB.

Simulation DSTs (possibly MDSTs) are then replicated according to a data placement policy, which is currently:

- 2 replicas on T0D1 storage at selected Tier1s and one at CERN.
- One archive replica on T1D0 custodial storage at CERN or a Tier1.

Analysis of simulated data is performed at Tier1s as for real data.

2.6. The LHCbDirac Grid framework

For all its Workload and Data management activities, LHCb relies on the LHCb extension of the now popular DIRAC Grid Community solution[4]. LHCbDirac contains LHCb-specific additions and extensions to DIRAC. For example the Bookkeeping system described later is purely LHCb-specific, as well as all the tools that rely on it.

DIRAC and LHCbDirac consist of a set of services and agents that collaborate for ensuring a consistent and resilient access to Grid and non-Grid computing resources (CPU and storage). DIRAC is used in several other computing systems [5].

The Data Management System described in this paper is integral part of LHCbDirac.

3. Datasets

The granularity for data references in LHCb is at the level of individual files, either for Data Management operations or for job input data definition. Depending on their usage, files can be viewed from two different perspectives:

- From the data access and job brokering point of view, files are uniquely identified by a Logical File Name (LFN) and a Global Unique Identifier (GUID). Each file can in turn have one or several replicas located at different storage elements (SE).
- From the user's perspective, files are seen as part of a dataset that corresponds to specific conditions and periods of data taking (or simulation conditions), and that have been processed (reconstruction and stripping) in similar conditions.

These two views of datasets are reflected in the existence of two catalogs in the LHCb DMS:

- A replica catalog contains the whole information on individual files and their replicas. This catalog is used for data manipulation and job brokering.
- A so-called bookkeeping catalog contains information on the provenance of the files. This catalog is used by physicists for selecting the set of files they want to analyse.

3.1. The Replica Catalog

3.1.1. Logical namespace

File metadata as well as replica information are kept in the LHCb replica catalog. It is a central catalog (at CERN) with distributed replications at each of the 6 Tier1s (read-only) for scalability and redundancy. Currently the replica catalog is based on the LCG File Catalog (LFC)[7].

Only files that have a physical existence in an SE are registered in the LFC, therefore with one or more replicas. Each file is identified by an LFN and a GUID. Following some construction rules, the LFN is built by the jobs that produce the files. The LFN namespace is hierarchical, essentially for ease of use and scalability. For convenience, the namespace contains information about the origin of the data, but this information is never used. The GUID is attributed by the software framework when the file is created. It is used within the files for further reference to event data between files. When opening a file, the framework also checks the consistency between its GUID and the one registered in the catalog.

At the top of the LFN path, the LFN reflects the content of the files (e.g. `/LHCb/data/2012/RAW` for RAW files collected in 2012) while the lower part of the path refers to the "production number" that created the file, and the number of the job that created it (e.g. `/00012345/0000/00012345_00006789_1.DST`). Productions are further split in subdirectories in order to limit the number of files to less than 10,000 per directory.

3.1.2. Storage Elements

Mostly for accounting purposes, LHCb uses internally logical Storage Elements for each type of data. In turn each LHCb SE is physically residing on a Grid Storage Element at a site.

The LHCb SEs are described in the DIRAC Configuration Service (CS) for LHCb. The description contains an access protocol (currently mostly SRM), and information on the physical SE:

- SRM endpoint, port and version of SRM
- SRM space token
- The Web Service URL used for accessing the path (e.g. `/srm/managerv2?SFN=` for SRM v2.2)
- Base path in the physical SE (a.k.a. SAPath)

This information prevents the necessity to query the BDII for any DMS operation. Other protocols can be supported, in particular rfio, but any plugin can be created provided the protocol implements the necessary API or CLI.

SRM is used by the DIRAC DMS using the gfal and lcg_util python bindings. RFIO uses the Castor CLI interface. Certain protocols may not allow the full-required functionality but can be sufficient for certain operations like file access or file transfer.

LHCb uses currently three spaces at each Tier1:

- LHCb-Tape: T1D0 custodial storage
- LHCb-Disk: T0D1 storage for centrally managed datasets (production and RAW data)
- LHCb-USER: T0D1 for user datasets stored on the Grid. This is a separate space to avoid users to block production operations.

3.1.3. Physical namespace

The Physical File Name (PFN, a.k.a. SURL for the SRM protocol) is constructed using a rule, from the logical SE information and the LFN:

SURL = srm:<endPoint>:<port><WSUrl><SAPath><LFN>

The SRM Space Token name is used as additional parameter for put and get operations, as well as for transfers using the File Transfer Service (FTS).

3.2. The Bookkeeping Catalog

3.2.1. Selection criteria

Physicists are not interested in the way the files they want to use have been created, neither on the way the logical and physical namespace are organized. For example the production number that is part of the LFN is irrelevant for them. Their view of datasets is based on the following:

- *Origin* of data (e.g. real or simulated data) and *year* of reference (e.g. 2012).
- Data taking or simulation *conditions*: beam energy, detector configuration, magnetic field polarity…
- Processing chain (e.g. reconstruction application and alignment quality, stripping selection…). This information is called the *Processing Pass* name.
- *Event type*: mostly for simulation, indicating which type of b-particle decay was simulated, for compatibility also used for real data.
- *File Type*: mostly useful in case the same processing produces multiple streams of data. Used however for all datasets.

3.2.2. Bookkeeping database model

The bookkeeping catalog [8] contains provenance information and metadata for all files that have ever been produced by any processing step in LHCb production jobs.

The bookkeeping database (BKDB) is based on a relation model that basically links input files to jobs that themselves produce output files etc… The BKDB contains provenance detailed information on the jobs. Files are primarily referred by their LFN, which ensures the connection to the replica catalog. In addition to the usual metadata (GUID, size, creation date…), the BKDB contains a flag for each file indicating whether the file has a physical replica (and therefore is registered in the LFC) or not. It also contains other information like the number of events, the associated run number etc…

3.2.3. Bookkeeping namespace

Seen from the user's perspective, the BK namespace appears like a file system. Several views can be used, but the most commons are (the first field is used to indicate the view used):

- [cond:]/<origin>/<year>/<conditions>/<processingPass>/<eventType>/<fileType>
- evt:/<origin>/<year>/<eventType>/<conditions>/<processingPass>/<fileType>

When selecting a dataset, a user specifies the BK path as described above. He gets back a list of real files (i.e. having at least one replica) matching this path. For practical reasons, files that should not be used for analysis (for example non-merged temporary files) have an additional visibility flag that by default hides them from users. They are however obtainable for production tools and jobs.

3.2.4. Quality flags

Each file has a quality flag associated that allows to classify a posteriori files according to the quality of their content. Initially files are created with a quality *Unchecked* or the quality of the input data of the job that created them. The quality flag may be changed at any time.

By default, when querying the Bookkeeping System users are receiving only files with a quality flag set as *Good*, but users can select any other flag or set of flags, on their own risk.

3.2.5. Bookkeeping browsing

Users can browse the bookkeeping information either using a standalone GUI or via the LHCbDirac Web portal. An example of the bookkeeping path search is shown in Figure 1.

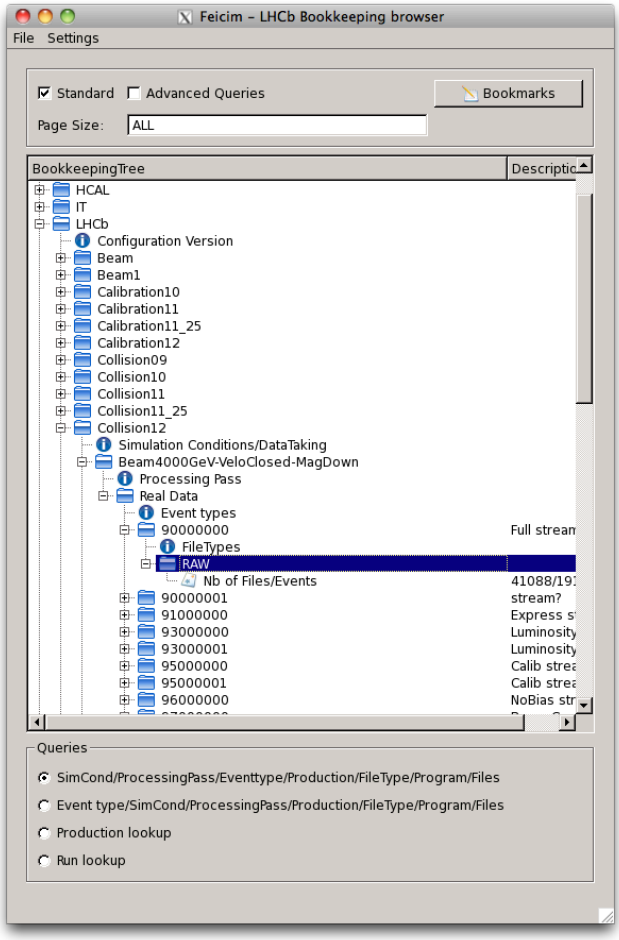

Figure 1: LHCb bookkeeping browsing window

4. Dataset-based transformations

The LHCb Workload Management and the Data Management systems in the LHCbDirac framework rely heavily on so-called *Transformations*[9]. A Transformation performs an action (processing, replication, removal…) on a dataset. In this context datasets are mostly defined as Bookkeeping queries.

The Bookkeeping dataset is expressed as regular bookkeeping query, including whenever necessary Data Quality flags. An LHCbDirac agent is querying the Bookkeeping System and constructing the corresponding list of files. Another agent is then creating tasks, that can be either processing or data management tasks. Each tasks is specified by its set of files and its target storages. **Error! Reference source not found.**Figure 2 shows a diagram for describing how transformations (a.k.a. productions) are created for both the WMS and the DMS.

As datasets are evolving, new files are added to transformations as they become available. The transformation system is therefore entirely data driven and new WMS and DMS tasks are continuously generated.

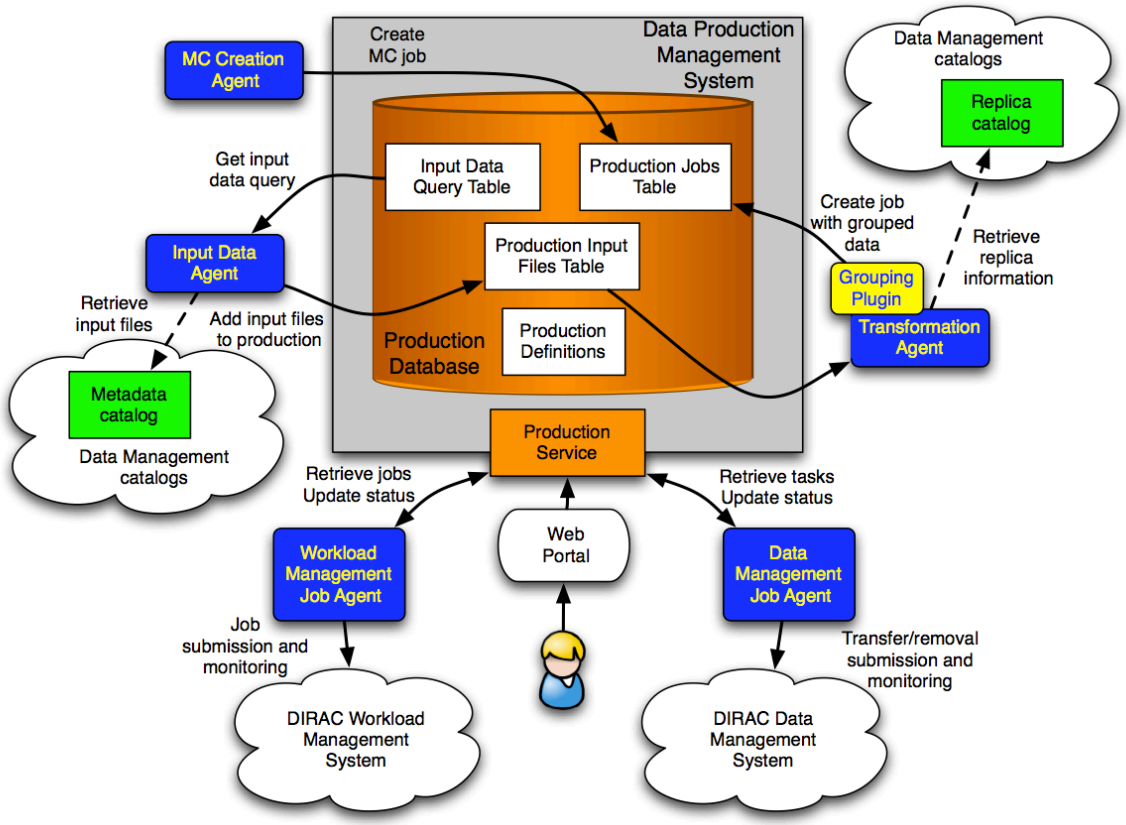

Figure 2: Schematic diagram of the LHCb transformation system, a.k.a. Production Management System

4.1. Processing Transformations

Transformations are set up for processing datasets (e.g. reconstruction of RAW files, stripping of SDST files, merging files…). These transformations are also called *Productions*.

In this case the target storages correspond to the input location (possibly multiple) for the jobs that will be automatically generated to accomplish this task. This allows a simple brokering of the jobs as they are eligible to run at sites that are defined to have access to the target storage SEs.

An agent creates jobs from the tasks that are then submitted to the LHCbDirac Workload Management system.

Upon completion, jobs upload their output to a specified (set of) Storage Elements. They register themselves and their output files in the Bookkeeping catalog, as well as the successfully uploaded output files in the replica catalog.

In case of a failure to upload an output file, the job tries and upload the file to a special LHCb Storage Element at one of the Tier1s, so-called Failover Storage Element. It also registers a task to be executed asynchronously for moving (i.e. replicate and remove) the file from this location to its final destination.

4.2. Replication Transformations

In order to implement the LHCb Computing Model for data placement, LHCb uses Data management transformations.

In this case the target storages of the tasks correspond to the replication destination of the files. These tasks are transformed by an agent into FTS transfer jobs that also ensures an automatic retry mechanism when the FTS transfer fails.

The replication permissions are checked before submitting the transfer requests, using the replica catalog ACLs.

Upon successful completion of the FTS transfer, the replica is registered in the replica catalog and the task is considered completed.

4.3. Removal transformations

The LHCb Computing Model also foresees that after data reprocessing, older datasets are retired in order to free disk space on storage, while keeping the archive replicas.

For removal transformations, the target storages correspond to those SEs from where the replicas should be removed. An agent is performing the actual file removal from physical storage as well as from the replica catalog. Only users with Data Management roles are allowed to remove production replicas or files.

Replica removal is protected against data loss, as it is not possible to remove the last replica of a file. This protection is based on the replica catalog information. As for replication, before any actual removal takes place, the permissions are checked using the replica catalog's ACLs.

The removal transformations can also perform full file removal (all replicas), for example in case some datasets are found to be totally useless because of a mistake in their creation, or because they are only test samples.

5. Custodial data usage

When creating processing jobs, the LHCbDirac system checks the location of the input data files. If the files have T0D1 replicas, these replicas are preferably used for defining the target storages.

If a file doesn't have a T0D1 replica, it must first be staged from the custodial storage onto a disk cache. A stager agent is responsible for handling this pre-staging before allowing the job to be matched by a pilot job.

The stager agent is currently pinning the file for a fixed amount of time (e.g. 6 or 24 hours). It keeps track of the amount of space that is occupied by pin files. Together with the knowledge of the disk cache size, this allows a proper regulation of the staging requests.

Staging requests are done using the SRM "bring online" functionality and polling on the status of the file.

The stager agent's parameters are not easy to tune, owing to the many different disk pool configurations and performance of tape systems at the sites, and the absence of publication of the disk cache sizes. Therefore the tuning is still empirical.

An alternative method is being considered that consists of pinning files for a longer time, assuming that the pinning would be released whenever the processing job completes successfully. A third method is advertised by the recent WLCG TEG recommendations, which consists of temporarily replicating the datasets on a T0D1 storage, the file being removed when not any longer needed. The

advantage of this method is to provide more scalability for data access as usually T0D1 SEs have many more servers than T1D0 SEs. The disadvantages are clearly the need for a disk-to-disk copy and the need for enough spare space on T0D1 storage.

6. Data Management operational experience

Currently the LHCb replica catalog (LFC) contains 2.2 million entries for a total of 5.5 million replicas for production data. The storage space used by production files is 7.5 PB.

About 40% of these files are control histograms that are all replicated only at CERN. Therefore if we exclude the histogram files, the number of data files is 1.2 million for a total of 3.3 million replicas, 1.3 million being located at CERN and the others distributed over the 6 Tier1 sites. This includes the active replicas on disk as well as the archive replicas on custodial storage.

User data represent another 2.8 million files for 4.4 million replicas. These files are however usually small and the total storage used by user data is 460 TB.

6.1. Storage availability

Storage elements may be unavailable temporarily at a site, due to foreseen or unforeseen interventions. During that time, no data management operations or jobs can access the data. It is therefore important to notify the system of this situation. The LHCbDirac Resource Status System allows all components of the system to access this information and take appropriate decisions. Storage element's access permissions are distinct for read, write, list and removal operations. For example archive storage is not allowed to be removed at any time nor to be accessed for read unless a dataset needs to be restored in T0D1 storage.

6.2. Transfer accounting

All DMS transactions are recorded in the DIRAC accounting system. This allows data managers to follow closely all activities and spot rapidly any malfunctioning. Figure 3 shows as an example the follow up of RAW data export to Tier1s that takes place just after an LHC fill is terminated.

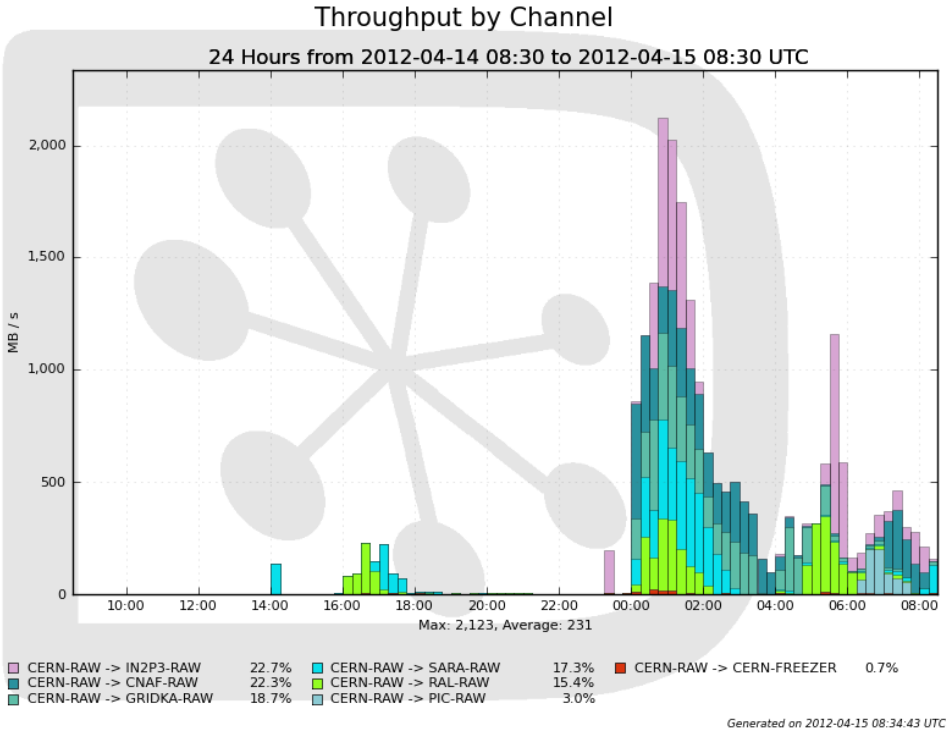

Figure 3: Example of data transfer accounting plots showing the RAW data export to Tier1s after the end of an LHC fill.

6.3. Storage usage accounting

A dedicated LHCbDirac agent is daily scanning the replica catalog and recording information about datasets. The granularity of this agent is at the level of directories in the logical namespace.

Each directory is then related to its bookkeeping namespace "directory" in order to give a view of storage usage based on physics datasets. This accounting contains information on both the logical files and the physical files (replicas), including a breakdown per SE.

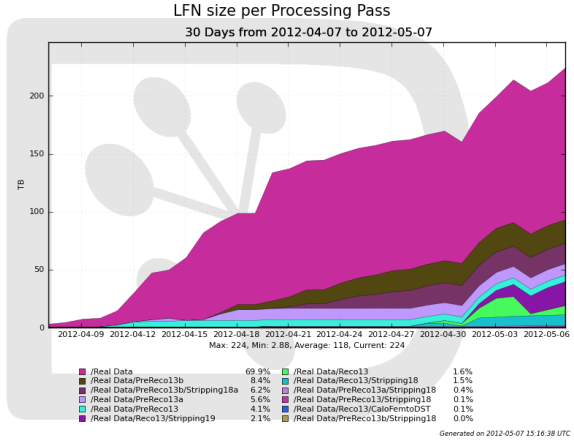

Figure 4: Storage occupancy in the logical space for 2012 data

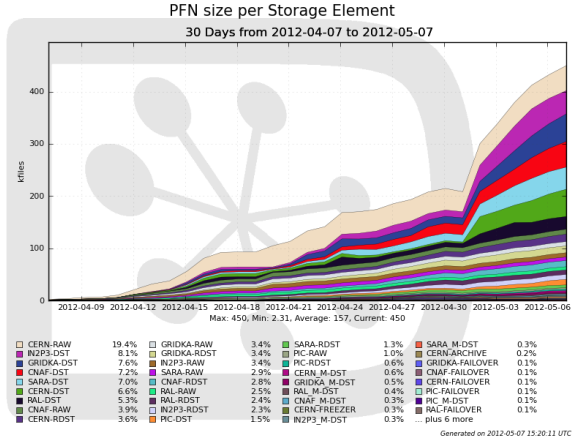

Figure 5: Storage occupancy in the physical space for 2012 per storage element.

Figure 4 and Figure 5 show the accounting for 2012 real data respectively in the logical space split by processing pass and in the physical space split by storage element.

6.4. Storage and catalog consistency

As the registration of files in the replica or the bookkeeping catalog is not directly coupled to the actual physical storage, some inconsistencies may occur. Similarly some files may be seen in the bookkeeping as having a physical replica while they don't. It is therefore necessary to periodically check the consistency between the storage and the two catalogs.

6.5. Data popularity

Currently the number of replicas kept on T0D1 storage elements for each dataset is handled by the Data manager based on the knowledge of the storage occupancy and the needs for free space. This is however not optimal as it may happen that even recent datasets usage is such that it does not necessitate a large number of replicas. Similarly the number of replicas for some extensively used datasets (for example before a conference deadline) may not allow an optimal redundancy and temporary additional replication may be advantageous.

Therefore a new reporting has been recently implemented in the LHCb jobs that accounts for the number of replicas actually used by Grid jobs within datasets. The evolution of this usage is a very good indication on the popularity of the dataset. It is foreseen to implement rule-based policies for determining the optimal number of replicas for each dataset, taking into account the storage capacity at each site and the expected storage space needed in the near future. In a first instance this information would help the Data manager in making decisions but could be later automated.

7. Conclusions

LHCb uses the Dirac Data Management System, with LHCb-specific extensions, mostly related to the Bookkeeping database. The DMS is fully integrated with the Workload Management System, as for example files pre-staging is triggered by the submission of jobs.

The Data Management operations rely on the same transformation system as the Workload Management system.

Sharing a large fraction of the base code allows a low level of maintenance costs as well as operational cost: all WMS and DMS operations are followed daily by two persons. A first level operational support follows the progress of operations, success and failure rates and reports problems to the second level support (Grid Expert On Call) who is in charge of interacting with the Grid sites through GGUS tickets and daily participation to the WLCG Operations meeting.

In addition a Production Manager and a Data Manager are in charge of preparing the WMS and DMS transformations and ensuring their successful completion.

References

[1] Augustor Alves Jr A *et al* 2008 The LHCb detector at the LHC *J. Instrum.* **3** S08005

[2] Antunes Nobrega R *et al* 2005 The LHCb computing technical design report *CERN/LHCC* 2005-019

Roiser S *et al* 2012 Major changes to the LHCb computing model in year 2 of LHC data *These proceedings*

[3] Stagni F *et al* 2012 LHCbDirac: distributed computing in LHCb *These proceedings*

[4] Tsaregorodtsev A *et al* 2008 DIRAC: a community grid solution *Journal of Physics: Conference series* **119** 062048

Tsaregorodtsev A *et al* 2010 DIRAC3: the new generation of the LHCb Grid software *J.Phys.Conf.Ser.* **219** 062029 doi:10.1088/1742-6596/219/6/062029

[5] Fifield T *et al* 2010 Integration of cloud, grid and local cluster resources with DIRAC *J. Phys.Conf. Ser.* **331** 062009 doi:10.1088/1742-6596/331/6/062009

Graciani R *et al* 2011 Belle-DIRAC Setup for Using Amazon Elastic Compute Cloud *Journal of Grid Computing* **9** 65-79 doi: 10.1007/s10723-010-9175-7

[6] Tsaregorodtsev A *et al* 2012 Status of the DIRAC project *These proceedings*

[7] Baud J-Ph *et al* 2005 Performance analysis of a file catalog for the LHC computing grid *HPDC* **14** 91-99

[8] Lanciotti E and Máthé Z 2009 The LHCb data bookkeeping system *CERN Document server* http://cdsweb.cern.ch/record/1170456/

[9] Smith A C *et al* 2010 DIRAC data production system *CHEP 2010* Presentation slides