DIRAC Data Management: consistency, integrity and coherence of data

Marianne Bargiotti, Andrew C. Smith

E-mail: marianne.bargiotti@cern.ch, acsmith@cern.ch

Abstract. The DIRAC Data Management System (DMS) relies on both WLCG (Worldwide LHC Computing Grid Project) Data Management services (LCG File Catalogues, Storage Resource Managers and File Transfer System (FTS)) and LHCb specific components (Bookkeeping database).
The complexity of both the DMS and its interactions with numerous WLCG components as well as the instability of facilities concerned, has frequently turned into unexpected problems in data moving and/or data registration, preventing a coherent picture of datasets.
Several developments in LHCb have been made in order to avoid data corruptions, missing data incoherence and inconsistences among Catalogues and physical storage both through safety measures at data management level (failover mechanism, check sums, roll back mechanism) and extensive background checks.
In this paper all the tools developed for checking data integrity and consistency will be presented as well as a Storage Usage agent , whose aim is to produce an up-to-date accounting of all LHCb storage usage using the LFC mirror database. The goal of this activity is the development of a generic tool suite able to categorize, analyze and systematically cure the disparate problems affecting the DMS in order to maintain a consistent picture of the main catalogues (Bookkeeping and LFC) and the Storage Elements.

1. Introduction

The DIRAC Data Management system (DMS) handles the management of files from their initial movement from the LHCb Online system to the grid and subsequently maintain metadata for these files in replica and bookkeeping catalogues. It handles the movement of files, both scheduled and automatic, and provides utilities to access metadata to allow these files to be accessed from computational jobs. To do these tasks effectively the system will be required to maintain complete self integrity between all its components and external resources (physical storage).

Data integrity checking is a critical part of the Data Management system ensuring the knowledge and accessibility of data consuming LHCb resources. In section 2 a brief overview of DIRAC and its Data Management System is given. In section 3 the data integrity checking architecture is exposed, while in section 4 and 5 a full description of the actual problem affecting the main catalogues (Bookkeeping and LCG File Catalogue (LFC)) together with the action taken for repairing is presented.

2. DIRAC architecture overview and DIRAC Data Management system

DIRAC (Distributed Infrastructure with Remote Agent Control) [1] project is the LHCb's Grid Workload and Data Management System. It is composed of a set of light-weight services and a network of distributed agents to deliver workload to computing resources [2].

Services : independent functions deployed and administered centrally on machines accessible by all other DIRAC components

Agents : software components requesting tasks from central Services for different purposes.

DIRAC integrates computing resources available at LHCb production sites as well as on the LCG grid.

This architecture implements a pull model to allow Services to carry out their tasks in a distributed approach. In this model the computing resource is actively seeking for tasks to be executed. Payloads are accumulated and validated by a production system and put into a task queue for being later on picked up by the first available computing resource declaring itself and matching the job requirements.
The DIRAC Data Management system deals with these three components:

- **LCG File Catalogue (LFC)**[3] providing information on replica location;
- **Bookkeeping DataBase (BK)** supplies extra information on the content and provenance of data files;
- **Storage Elements** underlying a grid Storage Element (SE), where physical files are stored

The consistency between catalogues and Storage Elements represents a fundamental requirement for a reliable Data Management [4].

3. LHCb File Catalogue

The File Catalogue exposes standard APIs for interacting with several available catalogues and allows the redundancy over all of them. Despite the fact that DIRAC has been designed to work with multiple catalogues, LFC is currently the choice for LHCb. It allows for the registration and retrieval of physical replica locations in the grid infrastructure. It stores both general grid file information (Logical File Name (LFN), GUID (Grid Unique IDentifier), size, date, owner) and replica information (Physical File Name (PFN), host SE).
The information provided by LFC is used by the DIRAC Workload Management System (WMS)[5] for brokering and scheduling jobs accordingly to the location of data they have to access.

DIRAC baseline choice is a central LFC catalog with multiple mirror instances: the presence of one single master (R/W) and many RO mirrors ensures a coherence obtained by a single write endpoint. The redundancy and load balancing is ensured by multiple mirrors updated in real time through Oracle Streams [6]. A sketch of the File Catalogs replication schema can be seen in Fig.1.

3.1. LFC replica registration

Replica registration in the catalogue follows a well defined scheme made up of a preparation phase and two following atomic operations. First of all the GUID check: before the registration in the LCG File Catalogue, at the beginning of transfer phase, the existence of the file GUID to be transferred is checked to avoid any GUID mismatch problem in registration.

After a successful transfer, LFC registration of files is divided into 2 atomic operations:

- booking of meta data fields with the insertion in a dedicated table of lfn, guid and size;
- replica registration.

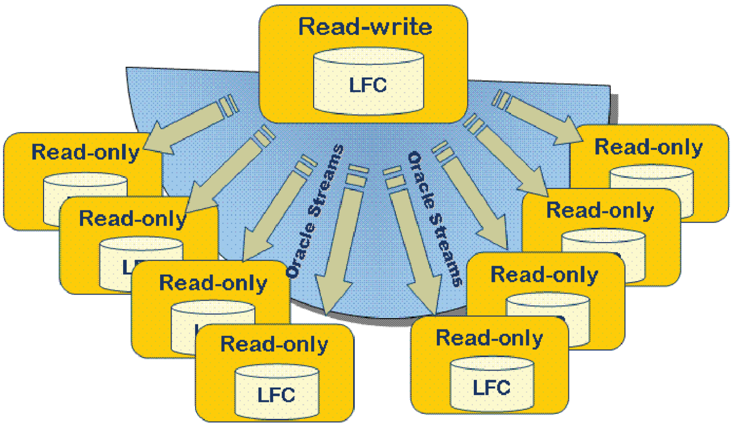


Figure 1. File Catalogues redundancy schema

If either of these step fails, a possible source of errors and inconsistencies is introduced in the system: the file is registered without any replica or with zero size. This represents one of the major sources of inconsistencies in the LFC, as will be detailed later.

4. LHCb Bookkeeping DataBase

The LHCbs Bookkeeping DataBase is the AMGA [7] based system that manages and contains provenance information regarding all LHCbs jobs and files:

- Job: Application name, Application version, Application parameters, which files it has generated etc..,
- File: size, event, filename, guid, from which job it was generated etc.

Three main services are available:

Booking service to write data to the bookkeeping

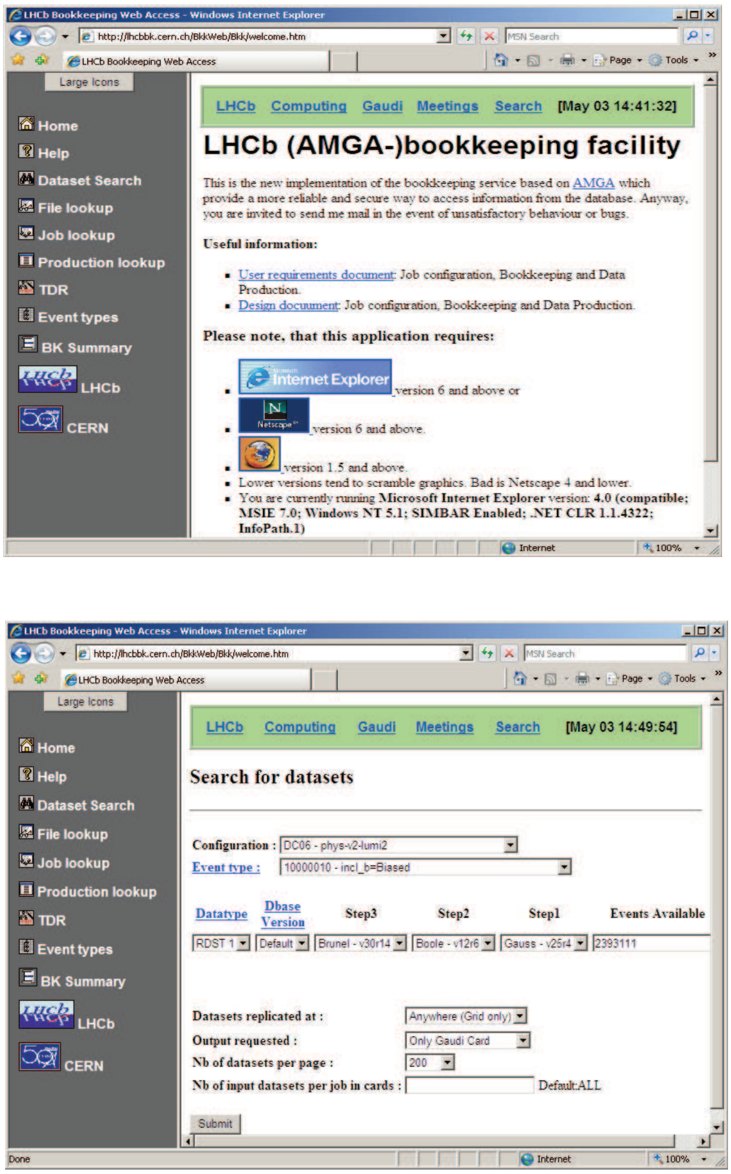
Servlets service for BK web browsing and selection of data files

AMGA Server for remote application use.

The Bookkeeping service represents the main gateway for users to select all the available data and datasets. Data is exposed to users trough a web interface (see Fig.2) that allows browsing the bookkeeping content and get information about file and their provenance. The interface exposes many functionalities, including the possibility to generate Gaudi [8] Card, the list of files to be processed by a job, and a wide choice of visualization for files (including replica information), jobs and productions. For its versatility the interface is used extensively by physicists for selecting events for analysis.
All data visible to users is flagged as `'has replica'`: in this way, in order to have coherent sets of data available for grid jobs, all the data stored in the BK and flagged as "having replica" have to be correctly registered and available in LFC. The coherence between these two databases represents one of the key feature of integrity checks.

5. Storage Elements

The DIRAC Storage Element (SE) client provides a standard API to various storage implementations.
The DIRAC SE is an abstraction of grid storage resources: it actually provides access granted by specific plug-ins, supports different access protocols (srm, gridftp, bbftp, sftp, http) and allows namespace management, file up/download, deletion etc.


LHCb Bookkeeping Web Access - Windows Internet Explorer
http://lhcbbk.cern.ch/BkkWeb/Bkk/welcome.htm
MSN Search
LHCb Bookkeeping Web Access
Page
Tools
Large Icons
Home
Help
Dataset Search
File lookup
Job lookup
Production lookup
TDR
Event types
BK Summary
LHCb
CERN
LHCb Computing Gaudi Meetings Search [May 03 14:41:32]
Internet
100%

LHCb (AMGA-)bookkeeping facility

This is the new implementation of the bookkeeping service based on AMGA which provide a more reliable and secure way to access information from the database. Anyway, you are invited to send me mail in the event of unsatisfactory behaviour or bugs.

Useful information:

- User requirements document: Job configuration, Bookkeeping and Data Production.
- Design document: Job configuration, Bookkeeping and Data Production.

Please note, that this application requires:

- Internet Explorer version 6 and above or
- Netscape version 6 and above.
- version 1.5 and above.
- Lower versions tend to scramble graphics. Bad is Netscape 4 and lower.
- You are currently running **Microsoft Internet Explorer** version: **4.0 (compatible; MSIE 7.0; Windows NT 5.1; SIMBAR Enabled; .NET CLR 1.1.4322; InfoPath.1)**

LHCb Bookkeeping Web Access - Windows Internet Explorer
http://lhcbbk.cern.ch/BkkWeb/Bkk/welcome.htm
MSN Search
LHCb Bookkeeping Web Access
Page
Tools
Large Icons
Home
Help
Dataset Search
File lookup
Job lookup
Production lookup
TDR
Event types
BK Summary
LHCb
CERN
LHCb Computing Gaudi Meetings Search [May 03 14:49:54]
Done
Internet
100%

Search for datasets

Configuration : DC06 - phys-v2-lumi2
Event type : 10000010 - incl_b=Biased

Datatype	Dbase Version	Step3	Step2	Step1	Events Available
RDST 1	Default	Brunel - v30r14	Boole - v12r6	Gauss - v25r4	2393111

Datasets replicated at : Anywhere (Grid only)
Output requested : Only Gaudi Card
Nb of datasets per page : 200
Nb of input datasets per job in cards : Default:ALL

Submit

Figure 2. Bookkeeping web interface

The Storage Resource Manager (SRM)[9] is the standard interface to grid storage, providing a unique interface hiding heterogeneity of storages in the WLCG. LHCb has 14 SRM endpoints, including disk and tape storage for each T1 site.
Among the many SRM functionalities (like extraction of meta data information, staging files, removing files and tURL retrieval), it will be possible, starting from SRM v2.2 [10], to browse the storage namespace. This will represent an essential resource for some intensive checks that LHCb is going to perform between SEs and catalogues.

All the SRM functionalities are exposed to users through Grid File Access Library (GFAL) [11]. The API python binding of GFAL Library is used to develop the DIRAC tools.

6. Data Integrity Checks

Considering the high number of interactions among DM system components, integrity checking is part of the DIRAC Data Management system. We have identified two different ways of performing checks:

- those launched by Data Manager to address specific situations
- those running as **Agents** on DIRAC framework

The Agent type of checks, that will be our main focus in this paper, can be broken into two further distinct types:

Check solely based on the information found on SE/LFC/BK : here we can identify three different kind of checks and a smart way of extracting the LHCb storage resource usage (Storage Resource Agent):

- Bookkeeping vs LFC
- LFC vs SE
- SE vs LFC
- Storage Usage Agent

Checks based on file distributions according to the Computing Model: i.e DST distribution at T1s disks

7. Data Integrity Checks and Data Management Console

The Data Management Console (DMC) is the fundamental interface for the Data Manager. The role of the Data Management Console is the portal through which the Data Manager performs routine activities. Currently the Console supports many catalog operations and fixes like:

- bulk extraction of replica information
- deletion of replicas according to sites
- extraction of replicas through LFC directories
- change of replicas SE name in the catalogue
- creations of bulk transfer/removal jobs

The Console will also allow the launch of integrity checks.
The Data Manager launched checks are developed in a rolling fashion (as per request) and accessible through the Console interface. Although these checks would be inherently specific, generalizations could be made by making use of regular expressions in LFN checks. These types of checks could be used to check for problematic files in a newly finished production or to answer questions from users regarding jobs crashing due to certain files.

8. DMS Integrity Agents overview

The consistency of DIRACs metadata catalogues and Storage Elements is performed with an integrity checking suite. This suite consists of a series of agents that check the mutual consistency of DIRACs three main Data Management resources: Storage Elements (SE), LFC, LHCbs Bookkeeping DB. These agents report all inconsistencies to the IntegrityDB. The Data Integrity Agent then resolves, where possible, the files contained in the IntegrityDB by correcting/registering/replicating files as needed.

The agents reporting to the central DB are those reported on the previous section: BK vs LFC, LFC vs SE, SE vs LFC and the Storage Usage Agent. The layout of the Integrity Agents suite can be seen in Fig.3.
A more detailed description of all these components is given in the following subsections.

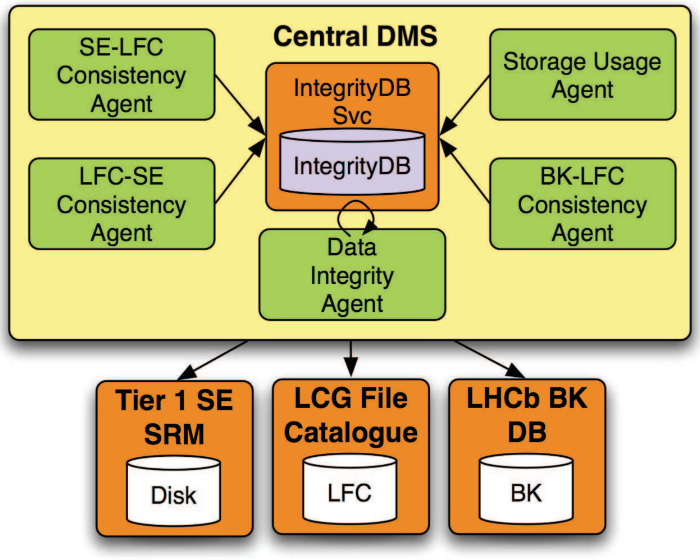


Figure 3. Central DMS and Integrity Agents overview

8.1. Bookkeeping vs. LFC Consistency Agent

The main issue affecting the BK service is the possibility of exposing wrongly flagged LFNs to users: these files have been correctly registered in the BK and flagged as 'having replica', but the corresponding replica information are missing in the LFC. In this way, user jobs selecting these files in the BK will not find a corresponding replica in the LFC, causing the failure of the job.

The Bookkeeping vs. LFC consistency agent is then able to verify that the visible files in the Bookkeeping exist in the LFC: it performs massive checks on LHCb production checking the BK dumps of different productions against the same directories on LFC.

For each production a check for the existence of each entry in the BK against the LFC is performed. Moreover a check on file size is done. Each missing or problematic file is reported to the central IntegrityDB.

8.2. LFC Pathologies

Many different possible inconsistencies can arise in a complex computing model: some of them have been introduced by bugs in previous versions of DIRAC (now fixed) and some others are due to failures in registering replicas in the catalogue.

zero size files : file metadata is registered in LFC but size information is missing (set to 0);

missing replica information : missing replica field in the Replica Information Table on the DB;

wrong SAPath : `srm://gridka-dCache.fzk.de:8443/castor/cern.ch/grid/lhcb/production /DC06/v1-lumi2/00001354/DIGI/0000/00001354_00000027_9.digi` registered at `GRIDKA-tape`: mismatch between the namespace and the SE: a file registered at FZK.de site can not contain a the string "cern.ch" in the path, as defined in LHCb naming convention;

wrong SE host : wrong or old information stored in the LHCb Configuration Service;

wrong protocol : replica registered in the catalogue with protocol other than SRM (sfn, rfio, bbftp,..);

mistakes in files registration : full assortment of mistakes in file registration like blank spaces on the surl path, carriage returns, presence of port number in the surl path, etc..

8.3. LFC vs. SE Consistency Agent

All the replicas stored in the LFC, need perfect coherence with storage replicas both in path, protocol and size. We can identify two different issues according to the specific operation performed:

Replication issue : the agent performs a check to verify that LFC replicas are really resident on physical storages: the existence and the size of files are checked. If files are not existing, they are recorded as such in the Integrity DB;

Registration issues : LFC - SE consistency agent stores problematic files in the central Integrity DB according to different pathologies:

- zero size files
- missing replica information
- wrong SA Path
- wrong protocol

8.4. SE vs. LFC Consistency Agent

The physical files registered on storage resources without a corresponding registered replica on the LFC, can never be accessed by users and represent a source of resource waste. This agent performs a complete check on the SE contents against LCG File Catalogue: it lists the contents of the SE namespace and verifies the physical files registration in the LFC. If files are missing (due to any kind of incorrect registration), they are recorded as such in the IntegrityDB.

What is actually missing is an efficient Storage Interface for bulk metadata queries (for example for directory listings); at the moment is not possible to list the content of remote directories and getting associated meta-data (lcg-ls). Further implementations on this side will be put in place through SRM v2.

8.5. Storage Usage Agent

This agent is able to construct an exhaustive picture of current LHCb storage usage, looping over the namespace of the LFC and generating a high granularity picture of storage usage using the registered replicas and their sizes. The extraction of the relevant information is obtained through breakdown by directory.

The retrieved information is stored on the central IntegrityDB indexed by LFC directories. This agent is therefore able to produce a full picture of disk and tape occupancy on each storage system and provides an up-to-date picture of LHCbs usage of resources in almost real time (looping time on catalog is 12 hours).

8.6. Data Integrity Agent

The Data Integrity agent takes actions on a wide number of pathologies stored by agents in the IntegrityDB. It's possible to differentiate the measures taken according to the different consistency checks performed by the previously described agents:

LFC vs. SE Consistency Checks In case of a missing replica in the LFC, a SURL path is produced starting from LFN, according to DIRAC Configuration System for all the defined storage elements. An extensive search is then performed throughout all T1 SEs: if the search is successful, the agent proceeds with the registration of missing replica, otherwise the LFN is removed from the LCG File Catalogue.

Bookkeeping vs. LFC Consistency Checks if a file is not present on the LFC, the agent performs an extensive search on all SEs. In case the file is not physically accessible on any of them, it proceeds with the removal of the tag "has replica", making it no longerr visible to user analysis. On the other hand, in case the file is available on one (or more) SEs, the agent updates the LFC by inserting the missing replica information.

SE vs. LFC Consistency Checks In case of missing files in the catalogue the agent can either register in the catalogue if the LFN is already present, or delete the file from SE if the LFN is missing on the catalogue.

This resolution can then take the form of re-replication, re-registration, physical removal or replica removal of files. Some pathologies may not be resolved automatically and remain in the DB for manual intervention.

8.7. Scalability Issues

DIRACs current approach assumes the information provided by the underlying resources is an accurate representation of the system. The limitation of this approach is in this assumption. A clear example is given by the problem of ensuring the consistency of the catalogues (either LFC or BK) with the storage resource, that is reflected within the SEs themselves. The contents of the SE namespace, like the contents of replica catalogues (and Grid resources in general), are not 100% reliable. An a-priori solution of the data integrity issues would require continual data access of all files and disk, that represents an physically impossible approach with the growing amount of experimental data.

The future approach within DIRAC will include detecting integrity issues as they arise and resolving them automatically post-priori. This approach is scalable and can be extended as new pathologies are observed.

9. Prevention of inconsistencies

A lot of effort has also been spent in the prevention of inconsistencies, besides the cure as developed with the DMS Integrity Agents.

The most important mean for preventing inconsistencies is the "failover mechanism", that uses the VOBoxes installation as a 'failover' for uploading of files that failed to go to their destination SEs. In the failover mechanism all failing operations on data (file transfers, file catalogue registration) are persistently cumulated as special XML records into Request DB sitting in the LHCb VO-Boxes. All the requests are then executed by dedicated agents running on VO-Boxes and retried as many times as needed until they succeed.

A "multiple VO-boxes" system ensures the persistency of the request to move output data to its true destination, while a centralized database of replication requests allows aggregation of requests into bulk transfers managed by FTS with access to dedicated high bandwidth network connectivity.

Other internal checks are also implemented within the DIRAC system to avoid data inconsistencies as much as possible. Among them, for example, the checking on file transfers based on file size or checksum.

10. Conclusions

The integrity check suite represents a vital part of Data Management activity in the provision of reliable data management. Further development will be possible with SRM v2 (for example the SE vs. LFC Consistency Agent). Most efforts are now focused on the prevention of inconsistencies through the use of the failover mechanisms and other internal checks. The final target is always the reliability of the service and the highest availability of data in order to minimize the number of occurrences of frustrated users looking for non-existing data.

References

[1] Andrei Tsaregorodsev 'DIRAC, community grid solution', Victoria: CHEP 2007 (id 189)

[2] R. Graciani and A. Casajus, 'DIRAC: Agents and Services', Victoria: CHEP 2007 (id 177)

[3] L.Abadie, Recent Developments in LFC, Victoria: CHEP 2007 (id 390)

[4] A.C. Smith, DIRAC Reliable Data Management, Victoria: CHEP 2007 (id 195)

[5] S. Paterson, DIRAC Optimized Workload Management, Victoria: CHEP07 (id 174)

[6] B. Martelli, LHCb experience with LFC database replication Victoria: CHEP07 (id 213)

[7] N. Santos and B. Koblitz, Distributed Metadata with the AMGA Metadata Catalog, in Workshop on Next-Generation Distributed Data Management HPDC-15,

[8] P. Mato, GAUDI-Architecture design document, LHCb note: LHCb-98-064.

[9] Arie Shoshani et al., Storage Resource Managers: Recent International Experience on Requirements and Multiple Co-Operating Implementations. 24th IEEE Conference on Mass Storage Systems and Technologies (2007)

[10] F. Donno, Storage Resource Manager version 2.2: design, implementation, and testing experience, Victoria: CHEP07 (id 78)

[11] F.Donno, Data management in WLCG and EGEE, CERN-IT-Note-2008-002.