DIRAC Data Management: consistency, integrity and coherence of data

Marianne Bargiotti

CERN

CHEP'07
VICTORIA, BC

International Conference on Computing in High Energy and Nuclear Physics

2-7 Sept 2007 Victoria BC Canada

Outline

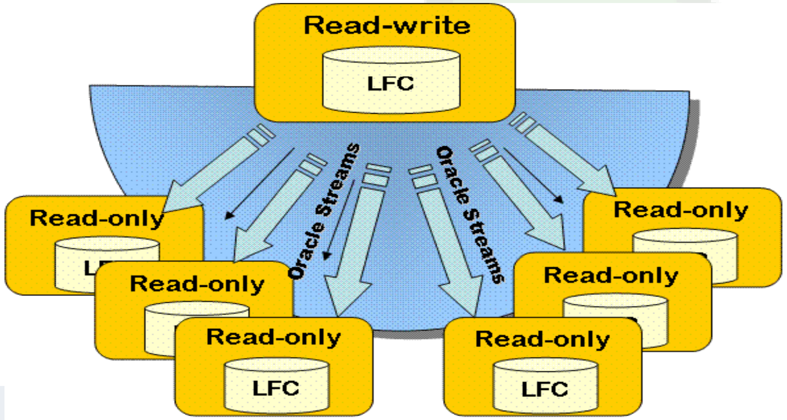
- DIRAC Data Management System (DMS)
- LHCb catalogues and physical storage
- DMS integrity checking
- Conclusions

DIRAC Data Management System

- DIRAC project (Distributed Infrastructure with Remote Agent Control) is the LHCb Workload and Data Management System
  - ❑ DIRAC architecture based on Services and Agents
    - see A. Tsaregorodsev poster [189]
- The DIRAC Data Management System deals with three components:
  - ❑ Storage Elements: files are stored in Grid Storage Elements (SE)
  - ❑ File Catalog: allows to know where files are stored
  - ❑ Bookkeeping Meta Data Data Base (BK): allows to know what are the contents of the files
  - consistency between these catalogues and Storage Elements is fundamental for a reliable Data Management
    - see A.C Smith poster [195]

LHCb File Catalogue

- The LCG File Catalogue (LFC) allows registering and retrieving the location of physical replicas in the grid infrastructure. It stores:
  - ❑ file information (lfn, size, guid)
  - ❑ replica information
- DIRAC WMS uses LFC information to decide where jobs can be scheduled
  - Fundamental to avoid any kind of inconsistencies both with storages and with related catalogues (BK Meta Data Data Base)
- Baseline choice for DIRAC: central LFC
  - ❑ one single master (R/W) and many RO mirrors
  - ❑ coherence ensured by single write endpoint



Registration of replicas

- Before the registration in the LCG File Catalogue, at the beginning of transfer phase, the existence of file GUID to be transferred is checked
  - ❑ to avoid 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 the dedicated table of lfn, guid and size
  - ❑ replica registration

→ **if either step fails:**

- → possible source of errors and inconsistencies
- → e.g the file is registered without any replica or with zero size

LHCb Bookkeeping data base

- The Bookkeeping (BK) is the system that stores data provenience information.
- It contains information about jobs and files and their relations:
  - ❑ 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.
- The Bookkeeping DB represents the main gateway for users to select the available data and datasets.
- All the data stored in the BK and flagged as '**having replica**' on the catalog, must be correctly registered and available in LFC.

Storage Elements

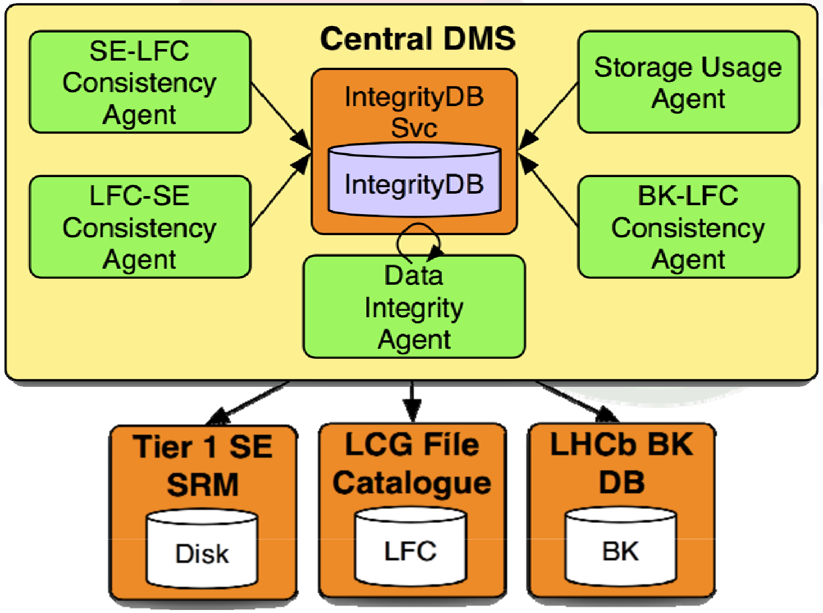
- **DIRAC Storage Element Client**
  - ❑ provides uniform access to GRID Storage Elements
  - ❑ implemented with plug-in modules for access protocols
    - srm, gridftp, bbftp, sftp, http supported
- SRM is the standard interface to grid storage
- LHCb has 14 SRM endpoints
  - ❑ disk and tape storage for each T1 site
- SRM allows browsing the storage namespace (since SRM v2)
- Functionalities is exposed to users through GFAL Library API
  - ❑ python binding of GFAL Library is used to develop the DIRAC tools

Data integrity checks

- Considering the high number of interactions among DM system components, integrity checking is part of the DIRAC Data Management system.
- Two ways of performing checks:
  - those running as Agents within the DIRAC framework
  - those launched by the Data Manager to address specific situations.
- The Agent type of checks can be broken into two further distinct types.
  - ❑ Those solely based on the information found on SE/LFC/BK
    - BK->LFC,
    - LFC->SE,
    - SE->LFC
    - Storage Usage Agent
  - ❑ those based on a priori knowledge of where files should exist based on the Computing Model
    - i.e DST always present at all T1's disks

DMS Integrity Agents overview

- The complete suite for integrity checking includes an assortment of agents:
  - ❑ Agents providing independent integrity checks on catalog and storages and reporting to IntegrityDB
  - ❑ Further agent (Data Integrity Agent) processes, where possible, the files contained in the IntegrityDB by correcting, registering or replicating files as needed



Data integrity checks & DM Console

- The **Data Management Console** is the interface for the Data Manager.
  - ❑ the DM Console allows data integrity checks to be launched.
- The development of these tools has been driven by experience
  - ❑ many catalog operations (fixes)
    - ➢ 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

BK - LFC Consistency Agent

- **Known problems: many lfns registred in the BK but failed to be registred on LFC**
  - **missing files in the LFC: users trying to select LFNs in the BK can't find any replica in the LFC**
    - Possible causes: Failing of registration on the LFC due to failure on copy, temporary lack of service..
- **BK→LFC: massive check on productions:**
  - ❑ checking from BK dumps of different productions against same directories on LFC
  - ❑ for each production:
    - checking for the existence of each entry from BK against LFC
    - check on file sizes

LFC Pathologies

- Many different possible inconsistencies arising in a complex computing model:
  - ❑ zero size files:
    - ❑ file metadata registred on LFC but missing information on size (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 GRIDKA-tape
  - ❑ wrong SE host:
    - ❑ CERN_Castor, wrong info in the LHCb Configuration Service
  - ❑ wrong protocol
    - ❑ sfn, rfio, bbftp…
  - ❑ mistakes in files registration
    - ❑ blank spaces on the surl path, carriage returns, presence of port number in the surl path..

LFC – SE Consistency Agent

- LFC replicas need perfect coherence with storage replicas both in path, protocol and size:
  - ❑ **Replication issue**: check whether the LFC replicas are really resident on Physical storages (check the existence and the size of files)
    - ➡ if file is not existing is stored in the IntegrityDB
  - ❑ **Registration issues**: LFC->SE agent stores problematic files in central IntegrityDB according to different pathologies:
    - ➡ zero size files
    - ➡ missing replica information
    - ➡ wrong SA Path
    - ➡ wrong protocol

SE – LFC Consistency Agent

- Checks the SE contents against LCG File Catalogue:
  - ❑ lists the contents of the SE
  - ❑ checks against the catalogue for corresponding replicas
    - ➢ if missing (due to any kind of incorrect registration) → ***Insert into Integrity DB***
  - ❑ missing efficient Storage Interface for bulk list operations and for getting meta data
    - ➢ extraction meta-data infos (even in bulk op, with prior knowledge of surls)
    - ➢ not possible to list the content of remote directories and getting associated meta-data (lcg-ls)
  - ❑ Further implementations to be put in place through SRM v2!!

Storage Usage Agent

- Using the registered replicas and their sizes on the LFC, this agent constructs an exhaustive picture of current LHCb storage usage:
  - ❑ works through breakdown by directories
  - ❑ produce a full picture of disk and tape occupancy on each storage
  - ❑ loops on LFC extracting files sizes according to different storages by directories
  - ❑ stores information on central IntegrityDB
  - ❑ provides an up-to-dated picture of LHCb's usage of resources in almost real time
- Foreseen development: using LFC accounting interface to have a global picture per site

Data Integrity Agent

- The Integrity agent spans over a wide number of pathologies stored by agents in the IntegrityDB.
- Action taken:
  - ❑ LFC – SE:
    - ➢ in case of missing replica on LFC: produce SURL paths starting from LFN, according to DIRAC Configuration System for all the defined storage elements;
      - extensive search throughout all T1 SEs
      - if search successful, registration of missing replicas.
    - ➢ same action in case of zero-size files, wrong SA-Path,..
  - ❑ BK - LFC:
    - ➢ if file not present on LFC:
      - extensive research performed on all SEs
      - if file is not found anywhere →removal of flag 'has replica': no more visible to users
      - if file is found:→ update of LFC with missing file infos extracted from storages
  - ❑ SE – LFC:
    - ➢ files missing from the catalogue can be:
      - registered in catalogue if LFN is present
      - deleted from SE if LFN is missing on the catalogue

Prevention of Inconsistencies

- **Failover mechanism:**
  - ❑ each operation that can fail is wrapped in a XML record as a request which can be stored in a Request DB.
  - ❑ Request DBs are sitting in one of the LHCb VO Boxes, which ensures that these records will never be lost
  - ❑ these requests are executed by dedicated agents running on VO Boxes, and are retried as many times as needed until they succeed
  - ❑ examples: files registration operation, data transfer operation, BK registration…
- **Many other internal checks are also implemented within the DIRAC system to avoid data inconsistencies as much as possible. They include for example:**
  - ❑ checking on file transfers based on file size or checksum, etc..

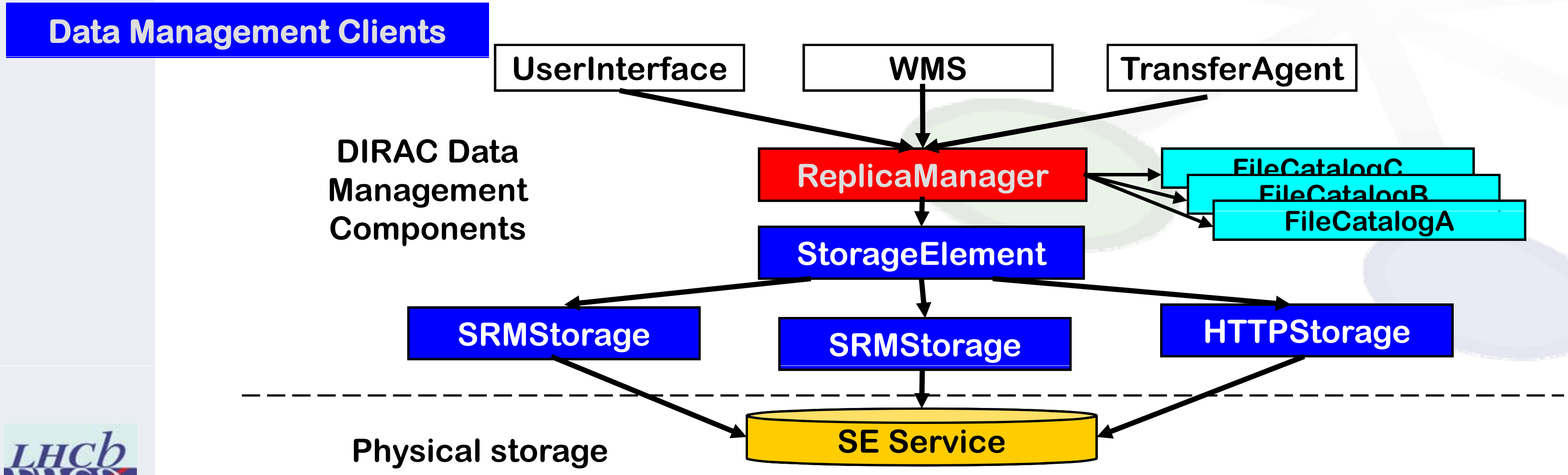
Conclusions

- Integrity checks suite is an important part of Data Management activity
- Further development will be possible with SRM v2 (SE vs LFC Agent)
- Most effort now in the prevention of inconsistencies (checksums, failover mechanisms…)
- Final target: minimizing the number of occurrences of frustrated users looking for non-existing data.

Backup

DIRAC DM System

- **Main components of the DIRAC Data Management System:**
  - □ **Replica Manager**
    - provides an API for the available data management operations
    - uploading/downloading file to/from GRID SE, replication of files, file registration, file removal
  - □ **File Catalog**
    - standard API exposed for variety of available catalogs
    - allows redundancy across several catalogs
  - □ **Storage Element**
    - abstraction of GRID storage resources: Grid SE (also Storage Element) is the underlying resource used
    - srm, gridftp, bbftp, sftp, http supported



Data Management Clients

UserInterface

WMS

TransferAgent

DIRAC Data Management Components

ReplicaManager

FileCatalogC

FileCatalogB

FileCatalogA

StorageElement

SRMStorage

SRMStorage

HTTPStorage

Physical storage

SE Service