DIRAC Data Management System

A. Tsaregorodtsev,
CPPM-IN2P3-CNRS
Plan

- DIRAC Project brief overview
- Data Management System problem
- DIRAC Data Management Model
- DMS Basic Components
- Managing Large Data Flows
- Extending DIRAC DMS
- Conclusions
DIRAC provides all the necessary components to build ad-hoc grid infrastructures **interconnecting** computing resources of different types, allowing **interoperability** and simplifying **interfaces**. This allows to speak about the DIRAC **interware**.
Interware: WMS

- Pilot based Workload Management
  - High user job efficiency
  - Suitable for usage with heterogeneous resources
  - Allowing application of community policies
DIRAC forms an abstraction of a simple computer which has the power of thousands of CPUs and petabytes of storage behind the scene coming from various sources and various technologies (grids, clouds, etc).

DIRAC Web Portal is following the computer desktop paradigm

- Natural for a non-expert user
Data is partitioned in files
- 0 to 10-20 GB

File replicas are distributed over a number of Storage Elements worldwide
- Distribution according to a Computing Model of the experiment

Files are registered in a single logical name space
- With metadata and ACL info

For most of applications the file access as simple as in a File System
- With additional upload/download if necessary
DMS objects

- Files
- Replicas
- Datasets

- Storage Elements
  - Access protocols
- Transfer channels
  - Transfer services

- Catalogs
- File Systems
DMS tasks

- Initial File upload
- Catalog registration
- File replication
- File access/download
- Integrity checking
- File removal

Often we are working with 10K to 1M files at a time
  - Make sure that ALL the elementary operations are accomplished
Data Management Model

- Storage element abstraction with a client implementation for each access protocol (SRM, XROOTD, gfal2 based, etc.)
- Each SE is seen by the clients as a logical entity
  - With some specific operational properties
  - New SE technologies, e.g. Federated Cloud, EOS are available after the proper configuration
  - SE’s can be configured with multiple protocols
Storage Element Proxy

- SE Proxy Service translates the DIRAC data transfer protocol to a particular storage protocol
  - Using DIRAC authentication
  - Using credentials specific to the target storage system

- SE Proxy Service allows access to storages not having access libraries on a given client machine
  - DIRAC or HTTP protocol

- Allows third party like transfers between incompatible storages
- Pilot submitted to the batch system through the SSH tunnel
- Pilot communicates with the DIRAC service through the Gateway proxy service
- Output upload to the target SE through the SE proxy
Central File Catalog (DFC, LFC, ...) is maintaining a single global logical name space.

Several catalogs can be used together:
- The mechanism is used to send messages to “pseudocatalog” services.

DataManager is a single client interface for logical data operations.
Replica Catalog

- File standard metadata
  - Size, ownership, time stamps, ACL, checksum

- Standard Replica Catalog functionality
  - Optimized for bulk queries

- On the fly PFN construction
  - Small database footprint
  - Full PFN can be stored if necessary

- Ancestor-descendent relations
  - Basic provenance information
  - Possibility to select ancestors in a given generation
Efficient Storage Usage reports

Necessary for
- quota policy management
- storage management

Using special prefilled tables
- Updated at each new file or replica insertion
  - More efficient with bulk insertion
- Instant reports for any directory
- Possibility of instant “du” command
### Storage Usage

**Report of storage usage for any directory**

- Whole community data
- Per user data
- “Logical” storage
  - LFNs, sum of the LFN sizes
- “Physical” storage
  - Physical replicas, total volume per Storage Element

```bash
FC:/> size -l /lhcb/user/a/atsareg/1
directory: /lhcb/user/a/atsareg/1
Logical Size: 134,756,946 Files: 498 Directories: 500

<table>
<thead>
<tr>
<th>StorageElement</th>
<th>Size</th>
<th>Replicas</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 IN2P3-USER</td>
<td>20,254,050</td>
<td>75</td>
</tr>
<tr>
<td>2 CNAF-USER</td>
<td>18,363,672</td>
<td>68</td>
</tr>
<tr>
<td>3 RAL-USER</td>
<td>16,473,294</td>
<td>61</td>
</tr>
<tr>
<td>4 CERN-USER</td>
<td>19,443,888</td>
<td>72</td>
</tr>
<tr>
<td>5 GRIDKA-USER</td>
<td>21,064,212</td>
<td>78</td>
</tr>
<tr>
<td>6 SARA-USER</td>
<td>20,254,050</td>
<td>75</td>
</tr>
<tr>
<td>7 PIC-USER</td>
<td>18,903,780</td>
<td>70</td>
</tr>
</tbody>
</table>

Total: 134,756,946 Files: 499

Query time 0.98 sec
```
Asynchronous operations

- Request Management System (RMS)
  - Keeps the database of Requests

- Request is a sequence of Operations executed in a certain order
  - Operations can have associated Files

- Each Operation type has a dedicated Executor
  - Execution is done with the credentials of the Owner of the Request
    - E.g. user defined operations
  - Examples: ForwardDISET, ReplicateFile, RemoveFile

- Executors are invoked by an agent running in a background
  - Retry logic in case of failures
Replication/Removal Requests with multiple files are stored in the RMS

By users, data managers, Transformation System

The Replication Operation executor

Performs the replication itself or

Delegates replication to an external service

E.g. FTS

A dedicated FTSManager service keeps track of the submitted FTS requests

FTSMonitor Agent monitors the request progress, updates the FileCatalog with the new replicas
Data driven workflows as chains of data transformations

- Transformation: input data filter + recipe to create tasks
- Tasks are created as soon as data with required properties is registered into the system
- Tasks: jobs, data replication, etc

Transformations can be used for automatic data driven bulk data operations

- Scheduling RMS tasks
- Often as part of a more general workflow
DFC is Replica and Metadata Catalog

- User defined metadata
- The same hierarchy for metadata as for the logical name space
  - Metadata associated with files and directories
  - Allow for efficient searches
- Efficient Storage Usage reports
  - Suitable for user quotas

Example query:

```
find /lhcb/mcdata LastAccess < 01-01-2012 GaussVersion=v1,v2 SE=IN2P3,CERN Name=*.raw
```
Support for datasets

- Datasets defined as a resulting list of files from a given metaquery
  - Particular case: all the files under a given directory
- Dataset objects are stored in the same directory hierarchy as files
  - ACLs, ownership, show up in the output of \texttt{ls} command as for files
- Datasets can be frozen in order not to change from one query to another
  - Can be refreshed by an explicit command, a quick check if changed since the last update
- Datasets can be annotated
- Operations on datasets
  - Replica lookup for all the files in a dataset
  - Total size, number of files report
  - Replication, removal, etc
LHCb accomplished migration from LFC to DFC
- Needed to develop a specific ACL plugin where several DIRAC groups have same ACLs for a given data
- Not using the Metadata features of the DFC except for the Storage Usage reports
- Using Transformation System of DIRAC for bulk data driven operations (e.g. replication, processing tasks submission, etc.)

ILC, BES III, CTA use intensively DFC as both Replica and Metadata Catalog
- BES III performed a detailed performance comparison with the AMGA metadata service

Pierre Auger Observatory
- Working on complex metadata queries and dataset algebra (dataset relations, intersections, unions, etc.)
Tests with Auger data
- ~30M files
- Identical LFC and DFC server hardware

BES Collaboration made a thorough comparison of DFC vs AMGA
- Similar performance
- More suitable functionality
Command line tools
- Multiple dirac-dms-… commands

COMDIRAC
- Representing the logical DIRAC file namespace as a parallel shell
- `dls`, `dcd`, `dpwd`, `dfind`, `ddu` etc commands
- `dput`, `dget`, `drepl` for file upload/download/replication

Web Interface
- Using a standard file browser paradigm
  - Possibility to define metadata queries
- Under development
  - Better connection to other systems (WMS)
  - Better support of the DIRAC “computer” paradigm
DIRAC for CTA: DIRAC File Catalog

- In use since 2012 in parallel with LFC. Full migration to DFC in summer 2015
- More than 21 M of replicas registered
- About 10 meta-data defined to characterize MC datasets

Query example:

ccta-prod3-query --site=Paranal --particle=gamma --tel_sim_prog=simtel --array_layout=hex --phiP=180 --thetaP=20 --outputType=Data

Typical queries return several hundreds of thousands of files
FSDIRAC

- DIRAC extension to mount the DIRAC File System as a local one
- Using FUSE, fuse-python
- Needs X509 credentials to browse data
- Files can be looked up in a standard file browser on Mac, Linux
- Simple if mounted read-only
- Not simple if mounted with write access enabled
  - Dealing with multiple replicas
DIRAC includes a general purpose Accounting System

For all the activities

Accounting reports for all the data related operations

- Transfer rates and volumes
- Storage usage
- Success/failure rates
- Etc

Plots selectable by

- Storage Elements
- Transfer channels
- Owner of the data
- Dates
- Etc, etc
Other services

- **Data Logging service**
  - Each operation on a chosen subset of name space changing the status of the file is recorded
    - Storage, identity of the operation initiator, status, etc
  - Useful in debugging problems with the data flows

- **Data Integrity service**
  - Each file access problem can be reported and accumulated in the Data Integrity database
  - Problem resolution either automatically or manually

- **Data Consistency service**
  - Comparison of catalog and storage contents
    - Usually needs a dump of the storage name space

- **Staging service**
  - Bringing data online before job submission
    - Asynchronous staging requests with polling for progress status
    - Pin time management
DIRAC has a well defined architecture and development framework
- Emphasis on modularity for easy extension
- Standard rules to create DIRAC extension
  - LHCbDIRAC, BESDIRAC, ILCDIRAC, …

Examples
- Support for datasets first added to the BESDIRAC
- LHCb has a custom Directory Tree module in the DIRAC File Catalog
  - Allows to customize the DIRAC functionality for a particular application with minimal effort
Conclusions

- DIRAC combines various distributed computing and storage resources in a coherent system seen by the user as a single large computer.

- The Data Management Model of DIRAC is organizing storage resources in a large distributed logical File System optimized for massive operations with data.

- The DMS is built in the DIRAC general framework and is naturally linked to other services – Workload Management, Transformation, etc.

- DIRAC DMS is extensible due to its modular architecture and can be easily adapted to the needs of particular applications.
Backup slides
Big Data

- Data that exceeds the boundaries and sizes of normal processing capabilities, forcing you to take a non-traditional approach to its analysis
  - Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, and information privacy
- Telecommunications/Internet traffic increase:
  - ~350 PB in 1990 ➔ ~1ZB (zettabyte = 10^6 PB) in 2015
- Google trends:
Already in 2010 the LHC experiments produced 13 PB of data
  - That rate outstripped any other scientific effort going on

<table>
<thead>
<tr>
<th>Size of data sets in terabytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business email sent per year ........................................ 2,986,100</td>
</tr>
<tr>
<td>Content uploaded to Facebook each year ................................ 182,500</td>
</tr>
<tr>
<td>Google’s search index ................................................................ 97,656</td>
</tr>
<tr>
<td>Kaiser Permanente’s digital health records ............................ 30,720</td>
</tr>
<tr>
<td>Large Hadron Collider’s annual data output ............................. 15,360</td>
</tr>
<tr>
<td>Videos uploaded to YouTube per year ..................................... 15,000</td>
</tr>
<tr>
<td>National Climactic Data Center database ................................. 6,144</td>
</tr>
<tr>
<td>Library of Congress’ digital collection ................................. 5,120</td>
</tr>
<tr>
<td>US Census Bureau data .......................................................... 3,789</td>
</tr>
<tr>
<td>Nasdaq stock market database ............................................... 3,072</td>
</tr>
<tr>
<td>Tweets sent in 2012 ................................................................ 19</td>
</tr>
<tr>
<td>Contents of every print issue of Wired .................................... 1.26</td>
</tr>
</tbody>
</table>

http://www.wired.com/magazine/2013/04/bigdata
LHC RAW data volumes are inflated by storing derived data products, replication for safety and efficient access, and by the need for storing even more simulated data than the RAW data:

~130 PB overall
LHC experiments, all developed their own middleware to address the above problems
- PanDA, AliEn, glideIn WMS, PhEDEx, …

DIRAC is developed originally for the LHCb experiment

The experience collected with a production grid system of a large HEP experiment is very valuable
- Several new experiments expressed interest in using this software relying on its proven in practice utility

In 2009 the core DIRAC development team decided to generalize the software to make it suitable for any user community.
- Consortium to develop, maintain and promote the DIRAC software
  - CERN, CNRS, University of Barcelona, University of Montpellier, IHEP

The results of this work allow to offer DIRAC as a general purpose distributed computing framework