

Running on multiple datasets and datasets staging

Dario Berzano

Laboratoire de Physique Corpusculaire - Clermont-Ferrand, France

ALICE Offline Week - Genève, July 8th, 2010

Outline

- * How to run a PROOF analysis on multiple datasets
- * The dataset manager
- * Staging policies: static and dynamic datasets
- * How to register and repair your personal datasets

How to run a PROOF analysis on multiple datasets

A simple example

```
AliAnalysisManager *mgr = AliAnalysisManager::GetAnalysisManager();  
/* some initialization here */  
mgr->StartAnalysis("proof", <datasets>, nEvents, nEventsSkip);
```

* A single dataset name

`"/alice/sim/LHC10a12_104800"`

* Several datasets separated by spaces, commas, pipes

`"/alice/sim/LHC10a12_104824 /alice/sim/LHC10a12_104825 ..."`

* A text file with one dataset per line

`"dslist.txt"`



dslist.txt
/alice/sim/LHC10a12_104800
/alice/sim/LHC10a12_104824
/alice/sim/LHC10a12_104825

Caveats about ROOT and AliRoot versions

- * This method is only supported in PROOF since a special ROOT version, v5-26-00-proof-04

Instructions here: <http://root.cern.ch/drupal/content/root-version-v5-26-00-proof>

- * A little modification required in AliAnalysisManager

Not yet in trunk but it will be available soon

Multiple datasets tutorial

- * For the impatient: if you install the «special» ROOT version you can try the multiple datasets tutorial, which already contains the modified AliAnalysisManager:

<http://skafinfo.saske.sk/alice/tutorial/aaf-tutorial-new.tar.gz>

- * Unpack, launch ROOT, then test it:

this file contains
the list of datasets

```
root[0] .x runAAF.C("dberzano@skaf-test.saske.sk", "ds.txt",  
kFALSE, "VO_ALICE@AliRoot::v4-19-18-AN", 1e7)
```

Two ways of running on multiple datasets

Grand dataset

- * Several datasets processed as if they were **merged** into one big dataset
- * In a string: pipe-separated
"ds1|ds2|ds3..."
- * List in a file: one by line
"dataset_list.txt"

Keep-separated

- * PROOF **notifies** when the processing of a dataset finishes and a new one starts
- * In a string: commas, spaces
"ds1,ds2,ds3..."
- * List in a file: one by line
"dataset_list.txt,"

See also: <http://root.cern.ch/drupal/content/working-data-sets#processmulti>

comma appended

The dataset manager

Brief introduction to datasets

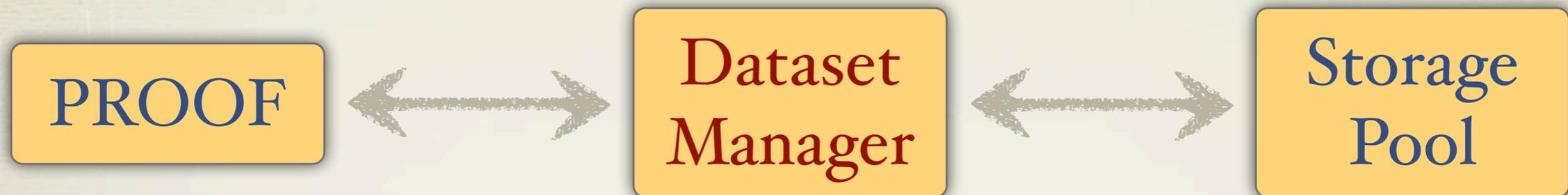
- * A dataset is a **list of files** with some «metadata»
- * Each entry has **several URLs** which all point to different copies of the same file
- * Each entry can be marked **(un)staged** if it has (not) been copied on the disks of the Analysis Facility
- * A file may also be **(un)corrupted**

Example of a file entry

has 10 000 evts - is Staged and Not Corrupted

1. root://alz80xl.to.infn.it:1095//alien/alice/data...
2. root://alz118wx:1095//alien/alice/data/2010/...
3. alien:///alice/data/2010/LHC10b/000117113/E...

What is the dataset manager?



- * Bidirectional **synchronization** between PROOF and the storage pool (usually xrootd)
- * It can be considered an interface to enable **storage management** from inside PROOF
- * It is **invisible** both for the sysadmin and the user

When the user registers a dataset...

- * The dataset manager **notices** the new dataset
- * It performs proper alien://-to-root:// URL conversions and tells the underlying storage to **fetch** the data
- * As long as transfer proceeds it **updates** the status of the files in the PROOF dataset list available to the user:
 - * how many files are **staged** or **corrupted**
 - * how many **events** are already available for analysis

Staging queue

- * When a file needs to be staged it is placed in a **queue**
- * The staging queue is completely held by the dataset manager
- * Several download requests are performed in **parallel** using threads

Corrupted files

- * A file can be marked as corrupted in **two cases**:
 - * The download fails several times
 - * The file can't be seen as a good ROOT file
- * When a download fails it is **requeued** at the end of the queue for a certain (configurable) number of times, to give opportunities to other files to be staged

Managing deleted datasets (upcoming feature)

- * The manager will **check** if the files in that dataset are not in other datasets: in this case it will tell the storage (xrootd) to **remove the files**
- * It is a way to keep the storage **clean** automatically
- * **Not yet available**, it will be implemented soon: it requires some modifications in the PROOF code
- * For this reason, **don't delete your datasets for now**, instead ask the system administrators

Staging policies

Static datasets

- * As in the old CAF
- * Only **staged** files can be analyzed
- * Staging occurs **asynchronously**: the dataset manager gets the files in the background
- * Needed at least for the most analyzed official datasets for **performance** reasons

Dynamic datasets / 1

- * The files are not really on the local disks
- * When doing the first analysis files are **downloaded**: it will be slow, but further analyses will be faster because files are **cached**
- * This is made possible via the **vMSS** (Virtual Mass Storage System) interface of xrootd

Dynamic datasets/2

- * The dataset manager **translates** URLs from alien:// to root:// and **fills** event information without staging files
- * xrootd will **remove** the files after «some time» since the last access, starting from **oldest** files, only when disk space is running out
- * This is why for now **quota** is not enabled (but it may change in the future: only 25 out of 80 TB used now)

Official and personal datasets

Official datasets

- * They can be found under:
 - `/alice/sim/*`
 - `/alice/data/*`
- * Sysadmins will take care of them (i.e. create them and «repair» corrupted files)

Personal datasets

- * If you are inside a group:
 - `/<group>/<user>/`
- * If you are not in a group:
 - `/default/<user>/`
- * **Ask to be added to a group!**
- * **You** take care of the «health» of your datasets with some provided utilities

How to register and repair your custom datasets

Utilities to manage datasets

* Some utility functions to manage **your own** datasets

* It is in global packages, so if you have the newest ROOT it will be downloaded for you:

only master needed

```
TProof::Open("dberzano@skaf.saske.sk", "masteronly");
```

```
gProof->EnablePackage("VO_ALICE@AFDSUtils::0.2.1");
```

* Full documentation available on AAF pages:

<http://aaf.cern.ch/node/117>

How to register your own datasets directly from AliEn/1

/* Example: search for ROOT files inside a ZIP archive */

afCreateGenericDsFromAliEn(

"/alice/data/2010/LHC10b/000117113/ESDs/pass2",  base path

"root_archive.zip",  the archive name

"/esdTree",  name of the default tree - **important!**

kFALSE,  if **kTRUE** we mark the files as staged to prevent staging

"AliESDs.root"  the «anchor»: file inside the archive that contains data

)

How to register your own datasets directly from AliEn/2

```
root [1] afCreateGenericDsFromAliEn(  
"/alice/data/2010/LHC10b/000117113/ESDs/pass2",  
"root_archive.zip", "/esdTree", kFALSE, "AliESDs.root");
```

```
=> Trying to connect to Server [0] root://pcapiserv01.cern.ch:  
10000 as User dberzano
```

```
Found 393 files (19.5 GiB total).
```

```
Dataset name (leave empty if you don't want to save it)?
```

```
/default/dberzano/foo_bar
```

```
Dataset saved
```

Monitor the status of the datasets

SKAF test datasets

What is this about?

Data Sets					
Data Set	Name	N events	Staged	Corrupted	Size
1. 1	/alice/sim/LHC10a12_104157	284100	96.24	0.508	709.7 MB
2. 2	/alice/sim/LHC10a12_104799	277500	94.2	5.092	1.358 GB
3. 3	/alice/sim/LHC10a12_104825	276300	94.27	2.149	1.123 GB
4. 4	/alice/sim/LHC10a12_104824	138000	46.7	0	1.892 GB
5. 5	/alice/sim/LHC10a12_104793	0	0	0	1.101 GB
6. 6	/alice/sim/LHC10a12_104792	0	0	0	68.44 MB
7. 7	/alice/sim/LHC10a12_104800	0	0	0	1.814 GB
8. 8	/alice/sim/LHC10a12_104316	0	0	0	675.4 MB

- * Via **MonALISA**: test it at <http://wiki.saske.sk:8080/stats?page=SKAF%2Fdatasets>
- * Soon available on **CAF** too

How to repair datasets/ 1

- * The function `afRepairDs()` allows you to list which files are corrupted, on a `text file` too, and take some actions:
 - * `uncorrupt` - mark as uncorrupted to tell the dataset manager to retry the download
 - * `unlist` - remove them from the dataset
 - * `unstage` - delete the corrupted files from the disks
- * These actions can be properly `combined`

How to repair datasets/2

- * List corrupted files on all datasets and save on `bad.txt`:

```
afDsRepair("/*/*", "", "bad.txt");
```

- * Remove bad files from all datasets of group MUON and delete them from the AF storage:

```
afDsRepair("/MUON/*", "unlist:unstage");
```

this can be a single dataset or a mask

combine actions with colons

Thank you!