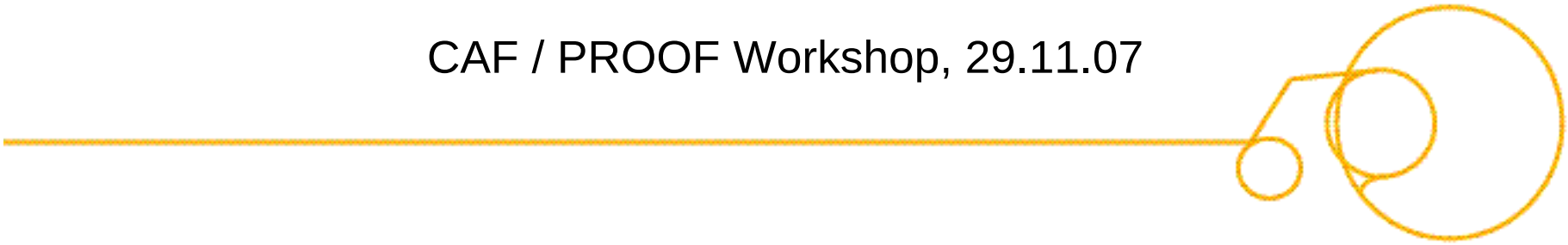
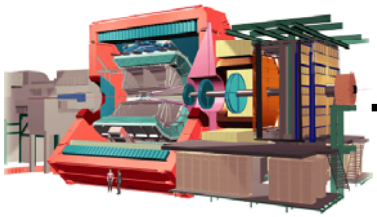
Dynamic staging to a CAF cluster

Jan Fiete Grosse-Oetringhaus, CERN PH/ALICE

CAF / PROOF Workshop, 29.11.07



CAF Schema



AliEn SE

CASTOR MSS

Disk Buffer

Tape

Tier-1 data export

Staging

CAF computing cluster

Proof master, xrootd redirector

Proof worker

local disk

Proof worker

local disk

Proof worker

local disk

...

Staging

- Files are produced in ALICE's PDC and stored in AliEn SEs (for CERN: CASTOR)
- Step 1 (first months): Manual
  - Files copied by a shell script to redirector that balances between disk servers
  - To allow user staging the nodes were open for writing
  - Complicated for users, no control over quotas, difficult garbage collection
- Step 2 (until mid 2007): Semi-automatic
  - Staging script plugged into xrootd
    - Prepare request with stage flag or open request to a file triggered staging
  - User gets list of files from the AliEn FC and triggers staging for all files
  - Convenient for users, no quotas, difficult garbage collection

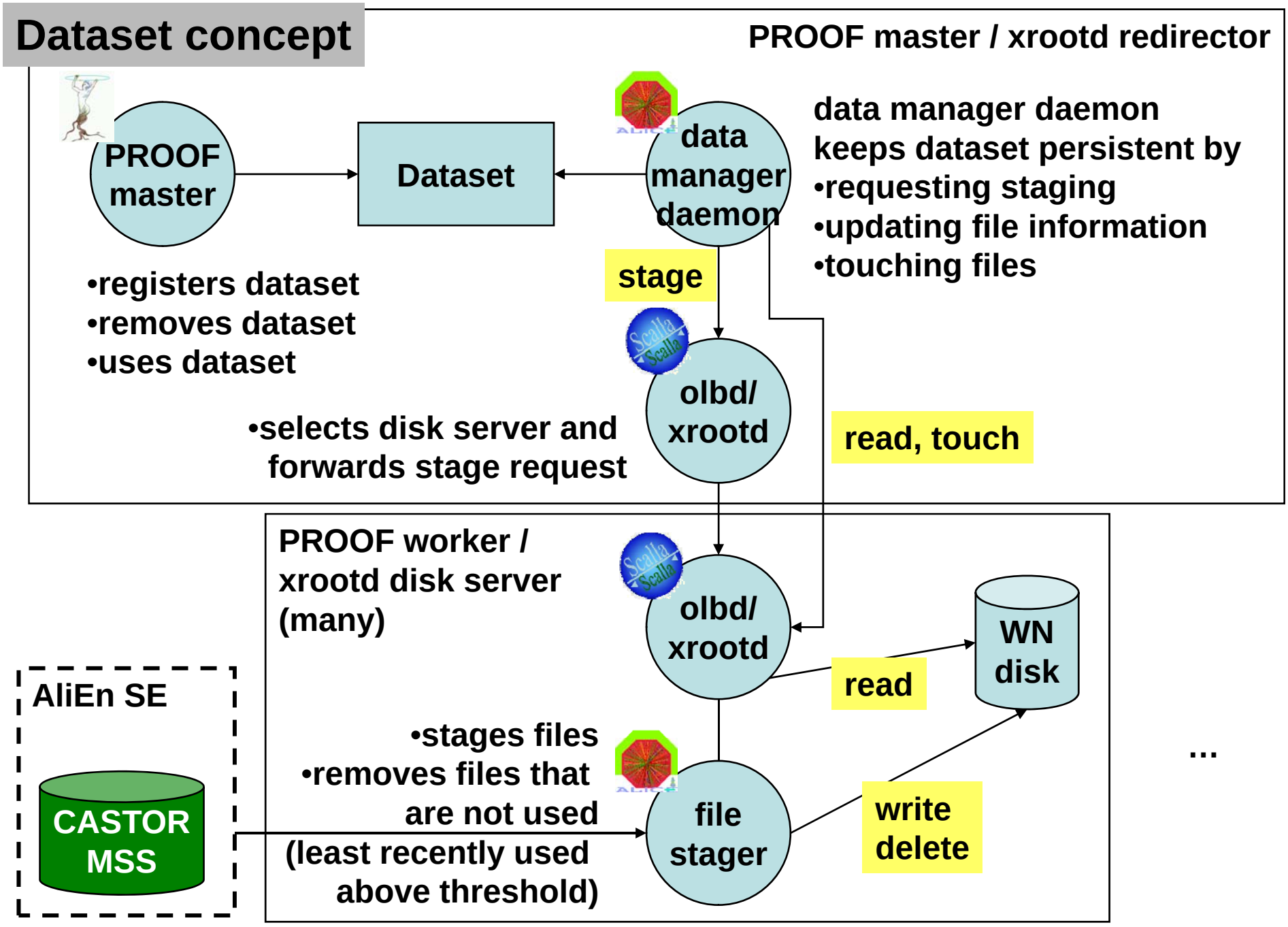
Staging (2)

- Step 3 (now): Automatic
  - Staging script plugged into olbd
  - Implementation of PROOF datasets (by ALICE)
  - Staging daemon that runs on the cluster
  - Transparent migration from AliEn collection to PROOF datasets
  - Convenient for users, quota-enabled, garbage collection
- 3 TB (100.000 files) staged to the system

Introduction of PROOF datasets

- A dataset represents a list of files (e.g. physics run X)
  - Correspondence between AliEn collection and PROOF dataset
- Users register datasets
  - The files contained in a dataset are automatically staged from AliEn (and kept available)
  - Datasets are used for processing with PROOF
    - Contain all relevant information to start processing (location of files, abstract description of content of files)
- File-level storing by underlying xrootd infrastructure
- Datasets are public for reading (you can use datasets from anybody!)
- There are common datasets (for data of common interest)

Dataset concept



Staging script

- Two directories configured in xrootd/olbd for staging
  - /alien
  - /castor
- Staging script given with olb.prep directive
  - Perl script that consists of 3 threads
  - Front-End: Registers stage request
  - Back-End
    - Checks access privileges
    - Triggers migration from tape (CASTOR, AliEn)
    - Copies files, notifies xrootd
  - Garbage collector: Cleans up following policy file with low/high watermarks (least recently used above threshold)

Data manager daemon

- Keeps content of datasets persistent on disk
- Regularly loops over all datasets
- Sends staging requests for new files
- Extracts meta data from recently staged files
- Verifies that all files are still available on the cluster (by touch, prevents garbage collection)
  - Speed: 100 files / s

PROOF master

- Registering, removal of datasets
  - Checks quota upon registration (group level quotas)
- Display datasets, quotas
- Use datasets
  - Meta data contained in dataset allows to skip lookup and validation step

Datasets in Practice

- Create DS from AliEn collection
  - collection = TGrid::OpenCollection(lfn)
  - ds = collection->GetFileCollection()
- Upload to PROOF cluster
  - gProof->RegisterDataSet("myDS", ds)
- Check status: gProof->ShowDataSet("myDS")

```
root [9] gProof->ShowDataSet("ESD5000")
Info in <TXProofServ::GetDataSet> on master-0: uri=ESD5000

TFileCollection ESD5000 - title contains: 21899 files with a size
of 1177346326985 bytes, 100.0 % staged
The files contain the following trees:
Tree /esdTree: 2189800 events
Tree /HLTesdTree: 2189800 events
```

- Use it: gProof->Process("myDS", "mySelector.cxx+") (not completely implemented)

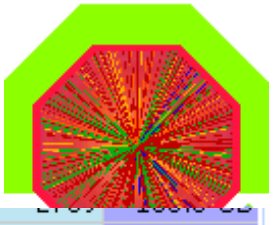
Dataset in Practice (2)

- List available datasets: gProof->ShowDataSets()

```
root [3] gProof->ShowDataSets()
Dataset URI                                |# Files|Default tree|# Events|   Disk   |Staged
/default/jgrosseo/ESD100                   |    6764|/esdTree    |   676400|  343 GB|  100 %
/default/jgrosseo/run82XX_part1            |   10000|/esdTree    |   998900|  288 GB|   99 %
/default/jgrosseo/run82XX_part2            |   10000|/esdTree    |   944700|  272 GB|   94 %
/default/jgrosseo/run82XX_part3            |   10000|/esdTree    |   987900|  285 GB|   98 %
/default/jgrosseo/ESD600                   |    1844|/esdTree    |   184400|   51 GB|  100 %
/default/jgrosseo/ESD_FullMisalignment     |     944|/esdTree    |    92100|   47 GB|   97 %
/default/jgrosseo/run12000                 |      62|/esdTree    |       49|    4 GB|   79 %
/default/jgrosseo/ESD5000                  |   21899|/esdTree    |  2189800| 1096 GB|   99 %
/default/jgrosseo/ProofSessionFiles        |   25438|/esdTree    |  4460900| 1640 GB|  100 %
```

- You always see common datasets and datasets of your group
- This method was used to stage 3 M events of PDC07

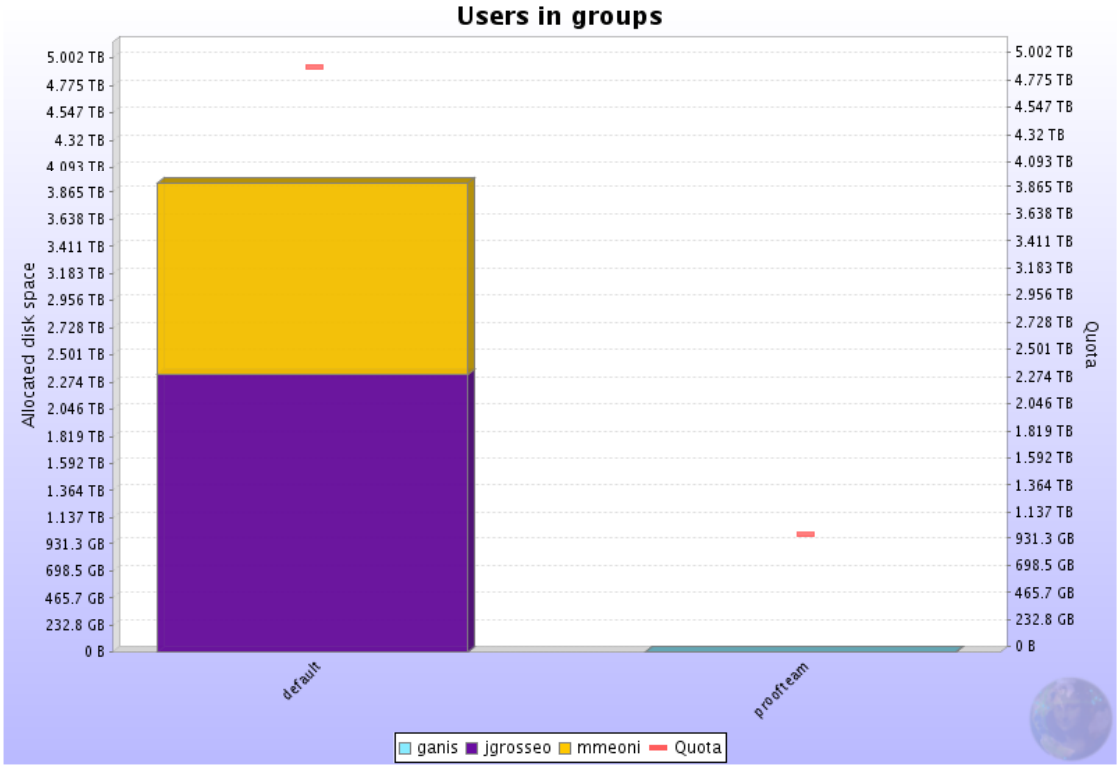
Monitoring of datasets



Number of files per host

15. lxb6055.cern.ch	2795	133.2 GB
16. lxb6056.cern.ch	2785	131.9 GB
17. lxb6057.cern.ch	2772	129.1 GB
18. lxb6058.cern.ch	1704	64.2 GB
19. lxb6059.cern.ch	1917	82.55 GB
20. lxb6060.cern.ch	2766	130.3 GB
21. lxb6061.cern.ch	1228	40.67 GB
22. lxb6062.cern.ch	2789	131.9 GB
23. lxb6063.cern.ch	2777	131.3 GB
24. lxb6064.cern.ch	2755	128.2 GB
25. lxb6065.cern.ch	2364	118.2 GB
26. lxb6066.cern.ch	2745	127.9 GB
27. lxb6067.cern.ch	2740	128.2 GB
28. lxb6068.cern.ch	2778	129.6 GB
29. lxb6069.cern.ch	2749	128.9 GB
30. lxb6070.cern.ch	2756	130.4 GB
31. lxb6071.cern.ch	2699	124.8 GB
32. lxb6072.cern.ch	2741	128 GB
33. lxb6073.cern.ch	2737	127.2 GB
34. lxb6074.cern.ch	2755	128.9 GB
35. lxb6075.cern.ch	2741	130.6 GB
36. lxb6076.cern.ch	2731	125.8 GB
37. lxb6077.cern.ch	2741	128.5 GB
38. lxb6078.cern.ch	2739	126.8 GB
39. lxb6079.cern.ch	601	16.8 GB
40. lxb6080.cern.ch	**96964**	**4.431 TB**
Total	**2620**	**122.6 GB**
Average		

Data set usage per group



Status

- Staging script implemented and in use since a year
- Daemon in place and running since 1-2 months
- Dataset handling in PROOF implemented (by ALICE), in ROOT SVN, but in an own development branch
- Processing of datasets in prototype stage (to be implemented by PROOF team)