



# dCache in May CCRC08

**Ron Trompert**

CCRC post mortem workshop  
june 12th-13th 2008

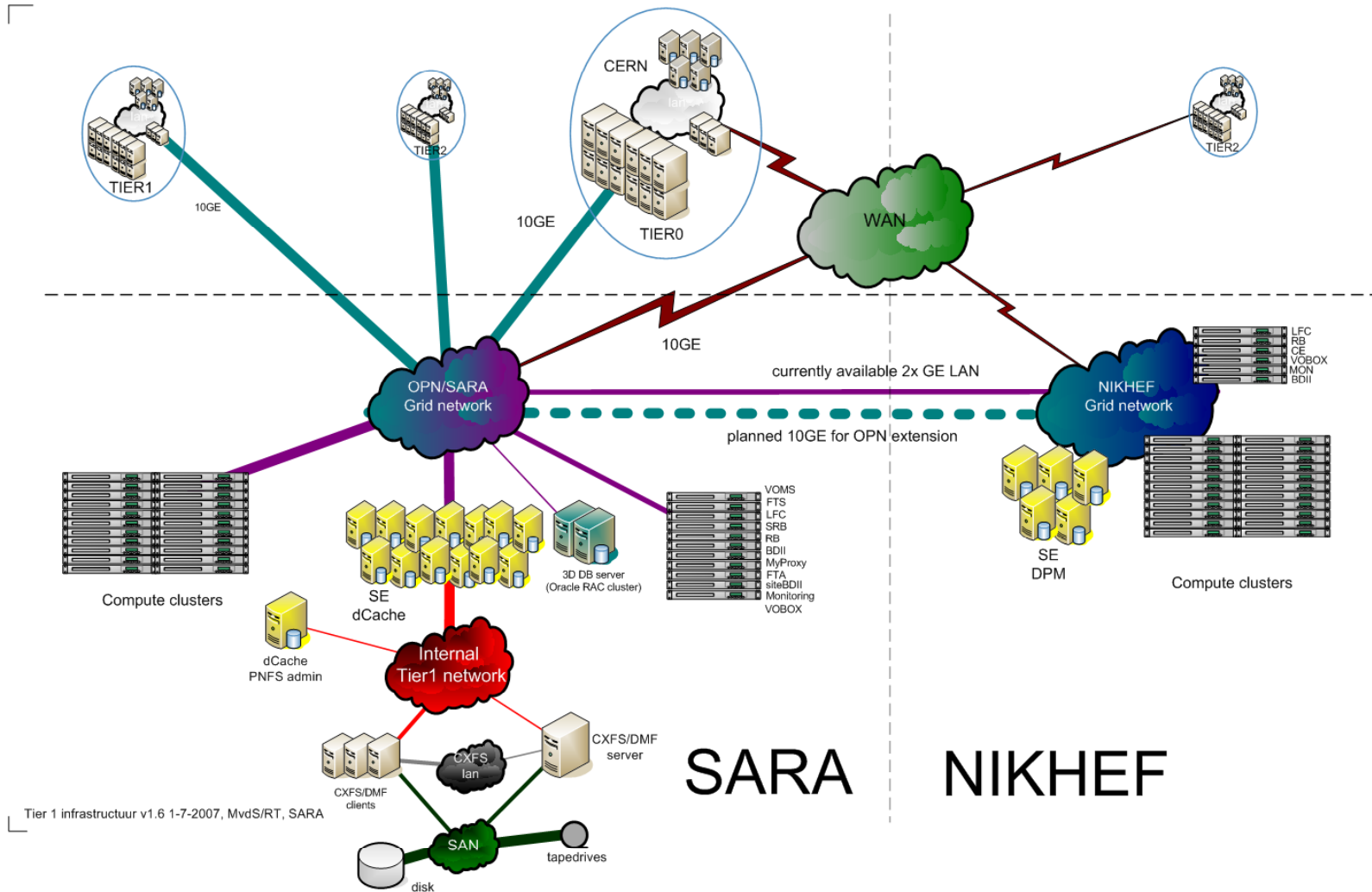


# Contents

- ▀ Storage Infrastructure
- ▀ dCache Setup
- ▀ Issues during May CCRC
- ▀ Questions for developers
- ▀ Stage tests

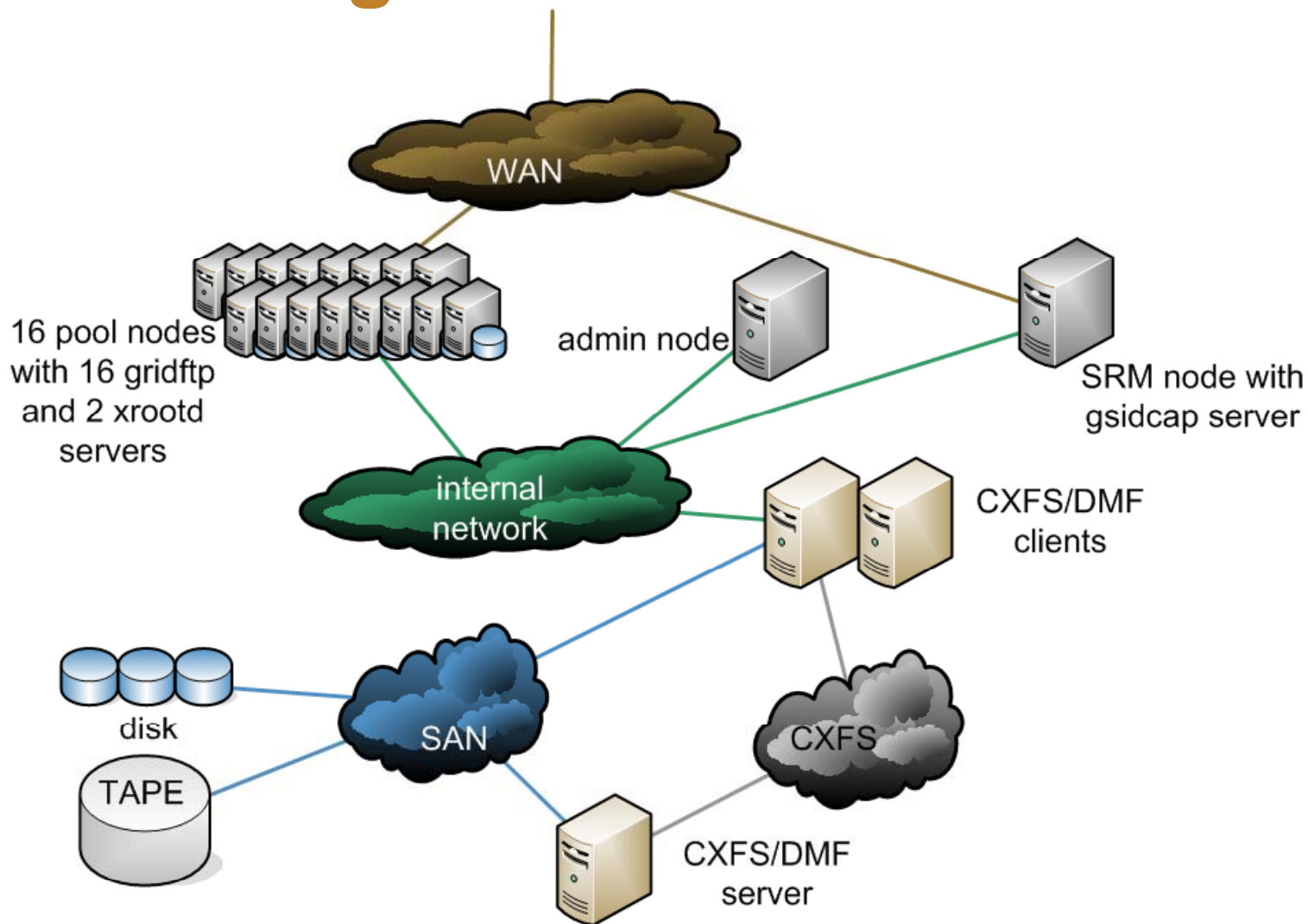


# Storage Infrastructure

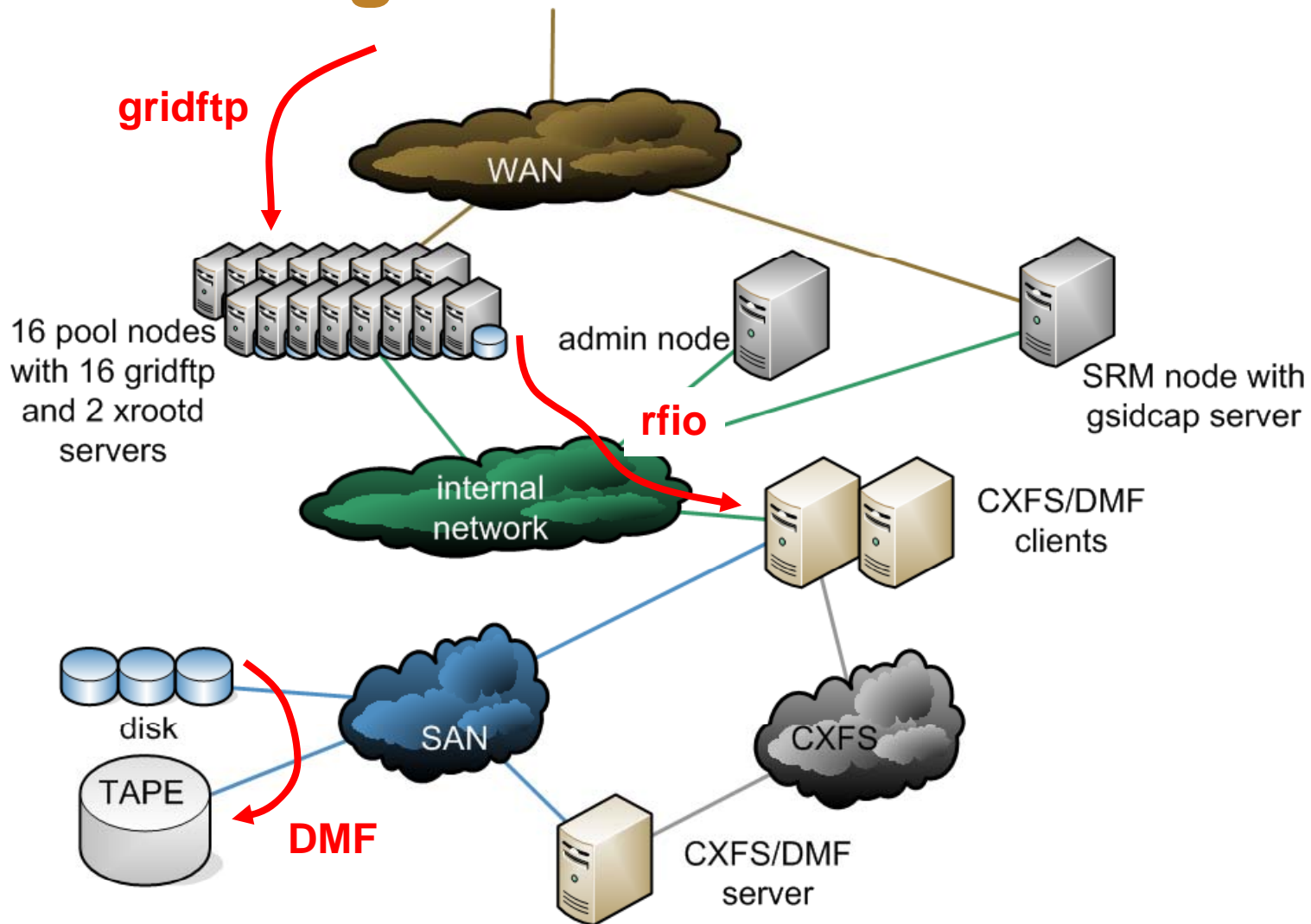


Tier 1 infrastructuur v1.6 1-7-2007, MvdS/RT, SARA

# Storage Infrastructure



# Storage Infrastructure





# dCache Setup

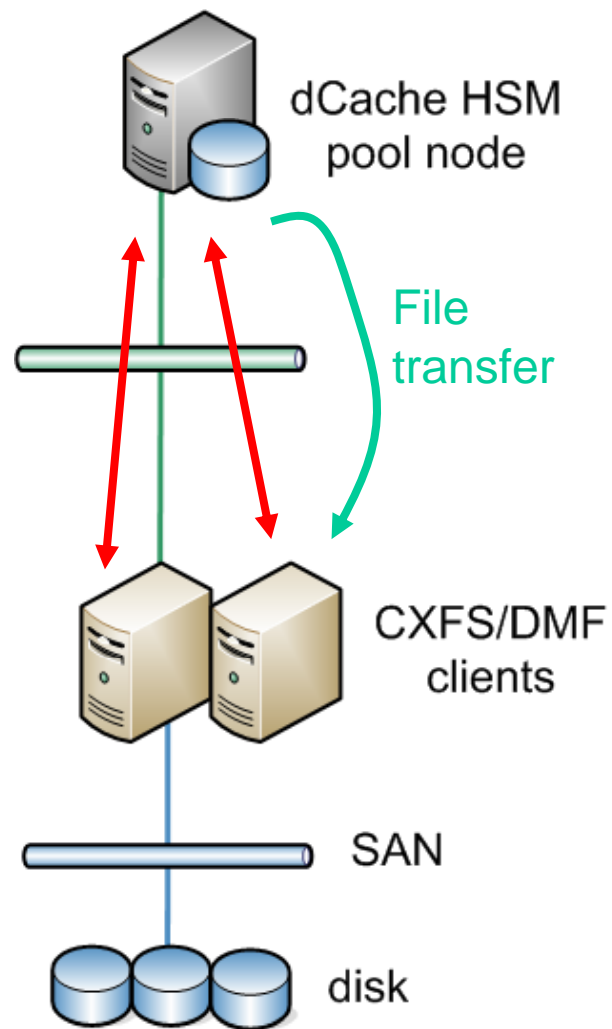
- Dedicated t1d0, t0d1,(t1d1 for LHCb) write pools
- Dedicated read pools (for t0d1 and t1d0 data)
- Dedicated cache pools
  
- Separate queues for gridftp and gsidcap transfers
- Movers:
  - Typically a pool has a maximum of 20 gridftp movers on our SUN x4500 servers and 6 on our Linux boxes and a maximum of 100 gsidcap movers
  - 2-4 store processes which store data to HSM on our tape write pools
  - 10 restore processes on our cache pools

# dCache Setup

## Load balancing

- dCache HSM pool node asks for load to both CXFS/DMF clients
- The node with the least load gets the transfer
- The maximum load is 3/core. If a node has a load of 3/core it will not get a transfer

current load?







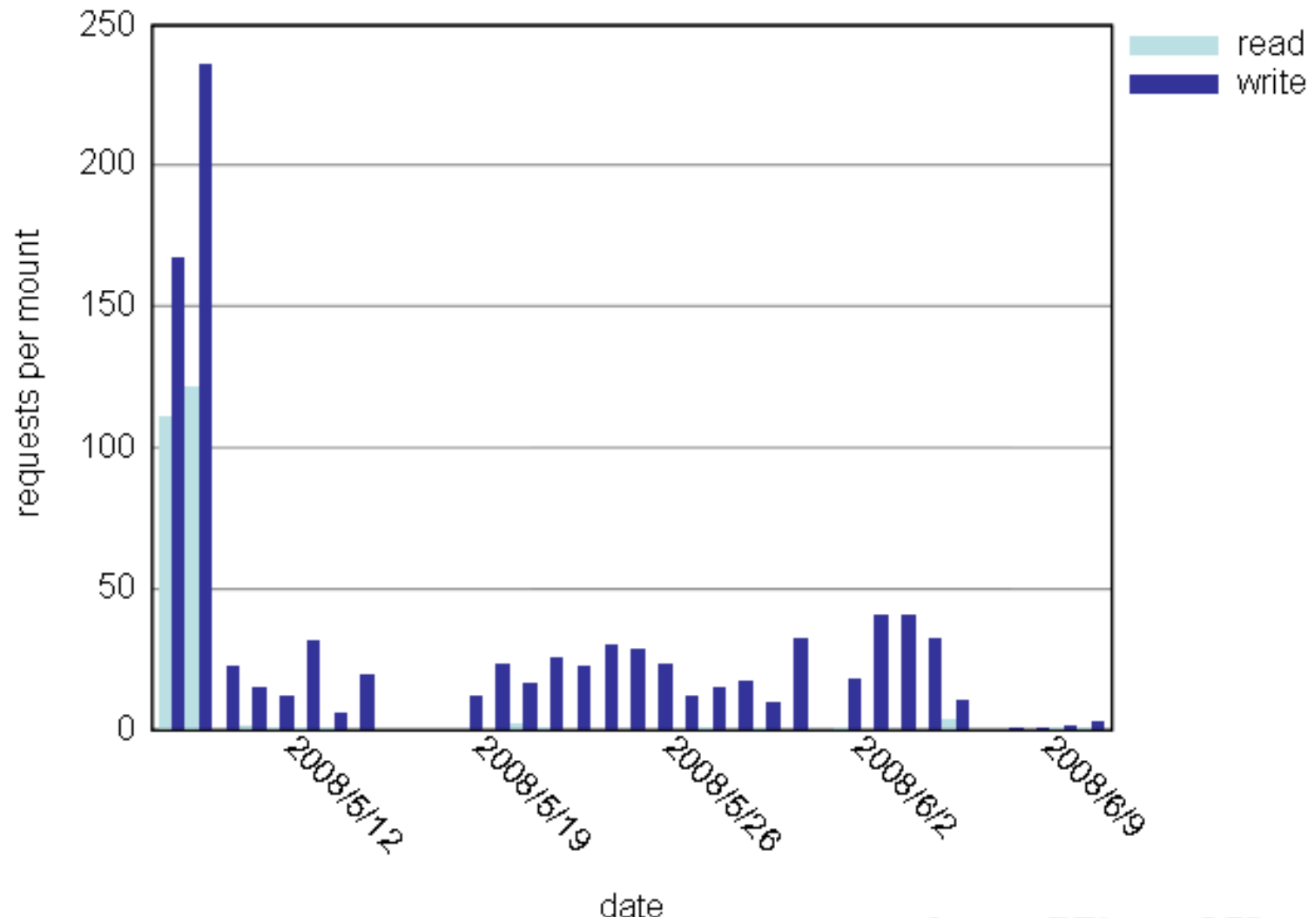
# Tape metrics

- file size per tape mount
- Rate per tape mount
- data transferred per tape mount
- repeat mounts
- requests per tape mount
- mounts
- request duration
- total amount of data transferred





ATLAS tape metrics: requests per mount



# Issues during May CCRC

- Full HSM pools during ATLAS T1-T1 tests
  - Turned out to be caused by orphan files due to failed transfers
    - ▶ When a file is removed from the pnfs namespace, dCache does not always cleanup the disk copy from the pool. This is a orphan file.
  - Transfers failed due to a transfer timeout in the FTS
  - Solved by:
    - ▶ Setting the transfer timeout to 1 hour
    - ▶ Cron job to check for orphan files and clean them up

# Issues during May CCRC

## ■ Bad throughput during ATLAS T0-T1 throughput tests to ATLASDATATAPE

- Number of movers on pools receiving the data was rather low
- During the T0-T1 tests there were also ATLAS M5 T1-T1 transfers to ATLASDATATAPE
- Solved by:
  - ▶ Doubling the number of mover processes on the ATLAS tape pools
  - ▶ Deploy an extra node for ATLASDATATAPE
  - ▶ ATLAS halted the M5 subscriptions

# Issues during May CCRC

## ■ Bad performance file staging for LHCb

- During these tests ATLAS was pushing hard with transfers to ATLASDATATAPE. But this is the first C in CCRC. Need to expand h/w infrastructure
- During these tests some non CCRC DIRAC jobs came along and did a lot of stage request for small files (150 MB) which killed the performance of our tape systems



# Issues during May CCRC

- **Space token issue.** From dCache 1.8.0-14 to 1.8.0-15 VOMS FQANs were checked against (existing) space reservations in the srm space table. The contents of this table depends on how reservations are done. This caused space not to be available anymore. “Fixed” this with a manual update on the srmspace table, was fixed in dCache in 1.8.0-15p4.
- **Srm returns wrong information.** File is copied to dcache SRM. The SRM returns NEARLINE even when the file still has a disk copy. This is also the case for data in t0d1 space.
- **Crashing gsidcap.** In dCache the gsidcap cell still was there but nothing listened on port 22128. Increased the logging but still no clue as to what causes this.
- **Crashing lfcdemon.** On three different LFC instances we have seen since the last upgrade that the lfcdemon crashed without any error message. A fix is underway. A temporary fix has been put in place to restart the daemon when it crashes.
- **Successful failures.** Files were correctly transferred to our site and registered in the catalog but ATLAS DDM thought that the transfer had failed.

# Questions for developers

- **What is the roadmap regarding srm implementation?**
  - Will d1<->d0 storage class transitions be implemented or are we going for file pinning?
- **We would like to be informed in advance when there are changes in de software requiring configuration changes**
- **dCache went from 1.8.0-15 at the start to 1.8.0-15p5 at the end. This is a lot. Can this not be prevented?**



# Stage tests

- So far reading from tape has never really been tested
- Experiments have tried to do some stage testing. But it is difficult for them to control the circumstances
  - Interference from other experiments and other users from the same experiment
  - Uncertainty about whether files are really only on tape or not
- We were just curious





# Stage tests

## ■ Preliminaries

- Wrote 500 2 GB files to tape
- Wrote another 50 2GB files to tape two days later with transfers in between to make sure that these files are on different tapes
- Files contained random data to avoid compression to influence the results

## ■ Tests

- Recall original 500 files from tape
- Recall 450 of the original 500 file from tape and the 50 new ones
- Test the prestager

## ■ Used dccp -P to stage file

- Srm-bring-online crashes



# Stage tests

## ▶ Prestager

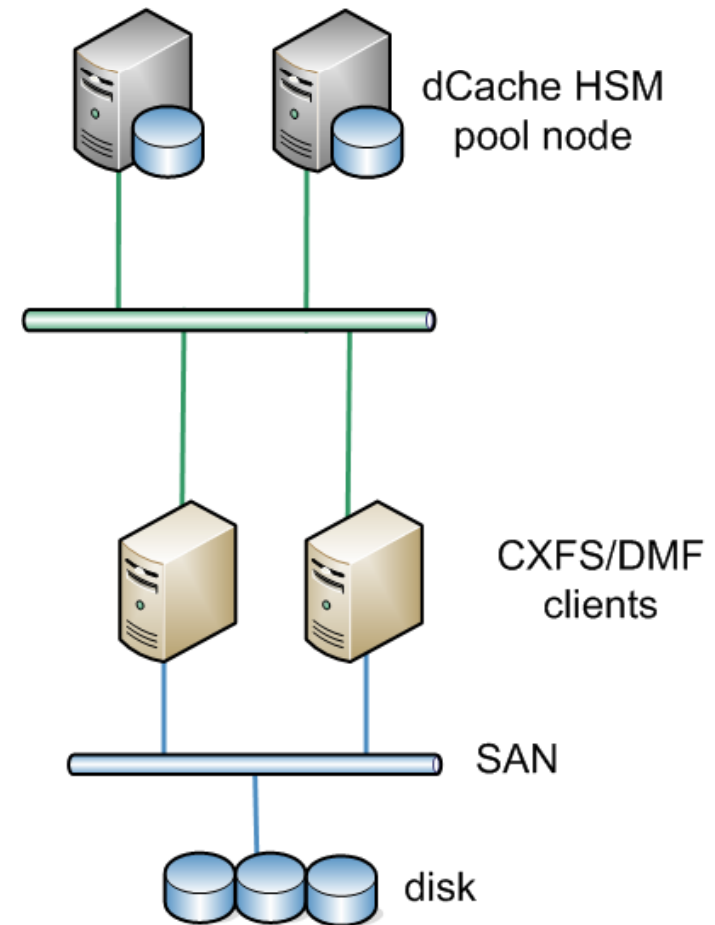
CellName	DomainName	Movers			Restores		
		Active	Max	Queued	Active	Max	Queued
bee12_5	bee12Domain	0	106	0	20	20	73

- ▶ A script looks at the files (73) which are queued to be restored and tells DMF to stage the files
- ▶ If the queued restore is due to become active, the file is already staged on the CXFS file system so you only have a disk2disk copy

# Stage tests

## Test setup

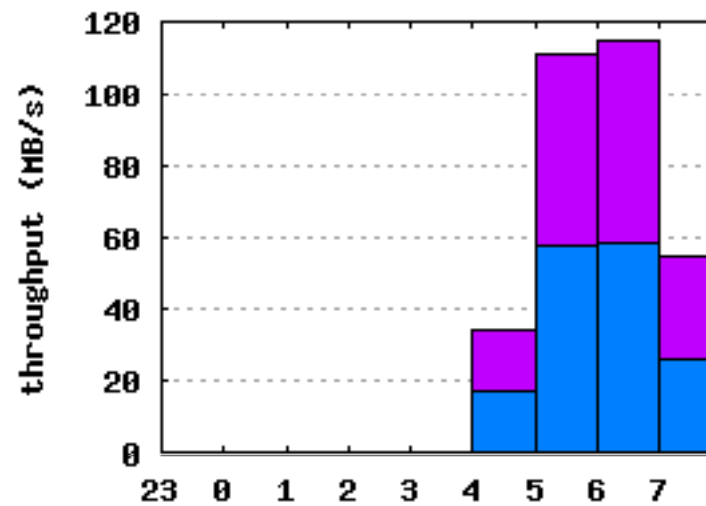
- 1.7 TB cxfs filesystem
- Two dcache pool nodes
- 10 restore processes per node
- 6 tape drives (9940B)





# Stage tests

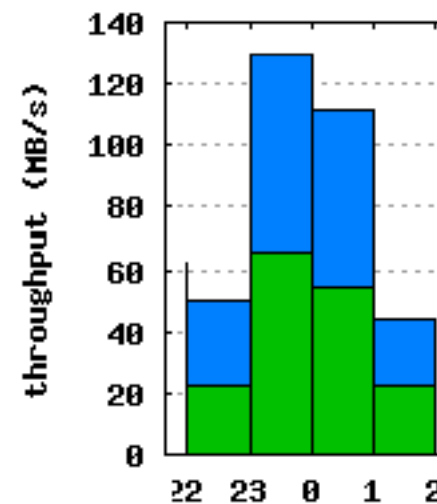
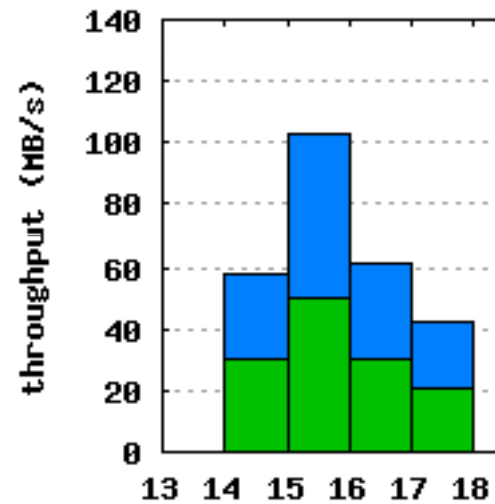
Recall 500 2GB from tape



Data on 8 tapes

# Stage tests

- Recall 450+50 2GB from tape without prestaging (left) and with prestaging (right)



- Data on 14 tapes