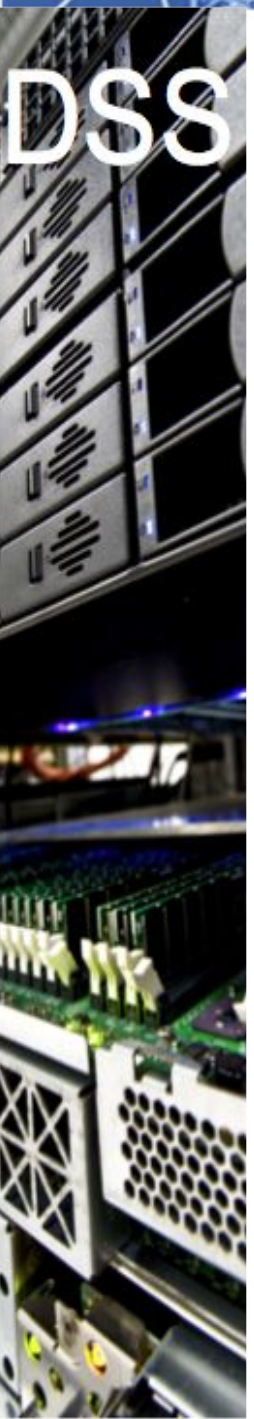


LHC data for Tier0 and Analysis

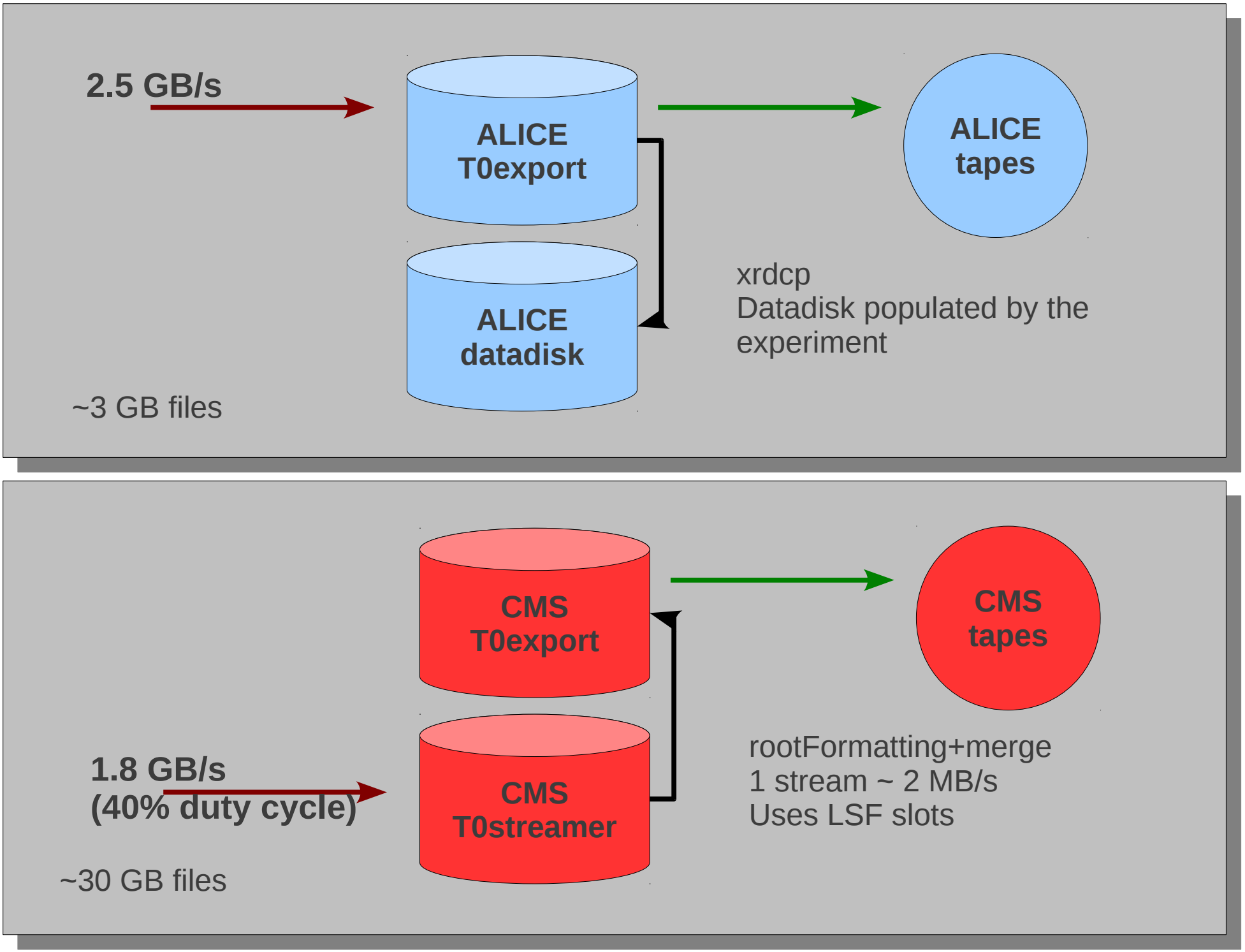
Massimo Lamanna / CERN
(for the IT-DSS group)

Introduction

- IT DSS is responsible of the data management for physics data at CERN
  - Mainly but not exclusively the LHC experiments
  - Physics data (disk and tapes): notably AFS and CASTOR
- **Production mode** but **not** a steady-state world
  - Technology evolves
  - New ideas are being discussed within HEP
  - Running LHC data management: experience
  - Data are coming! Lots of data
    - 10+ M files per month
    - Times 3 in total size in the next few years (40 PB → 120 PB in 2015)
    - Real data are more interesting than MonteCarlo (users, users, users)
    - Master operational load!

CASTOR: examples of recent activity

- Heavy-ion run (Fall 2010)
- Verify the compatibility of running ALICE and CMS concurrently
  - ALICE 2.5 GB/s continuous (with LHC duty cycle 100%)
  - CMS 1.8 GB/s assuming a ~50% LHC duty cycle)
    - With 100% duty cycle CMS would alternate data collection to data export
  - Substantially higher than expected (cfr. LCG TDR)
    - At these rates: collect data and export “after the run” (always the case for ALICE).
    - CMS: in January data compression step (factor 5-10) and then an export
- Actions:
  - Put on the floor 2011 resources
    - Provide a “local” copy (since there is no Tier1 export during the run)
  - Prepare a CASTOR update to improve especially tape handling
  - Perform a test!
- Coexist with ATLAS (~300 MB/s) during heavy ions + reprocessing, LHCb (2010 reprocessing), COMPASS, LHC user analysis (pp data)

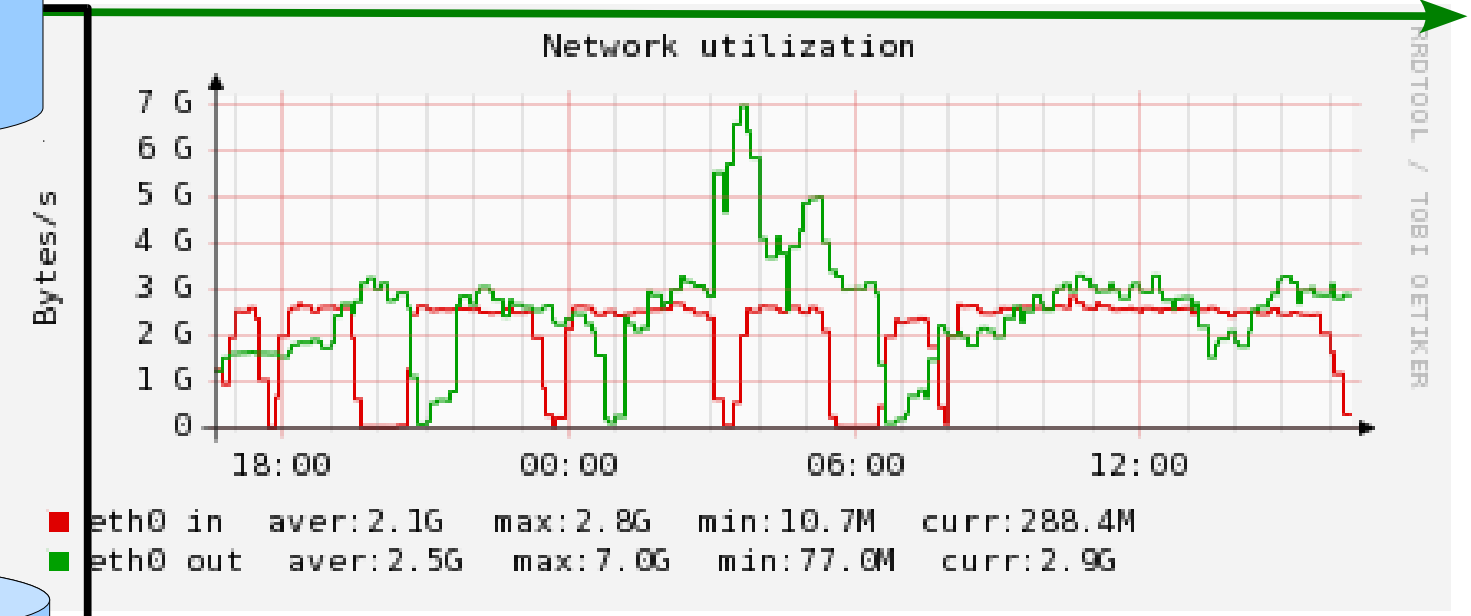

2.5 GB/s
ALICE T0export
ALICE tapes
ALICE datadisk
xrdcp
Datadisk populated by the experiment
~3 GB files
CMS T0export
CMS tapes
1.8 GB/s (40% duty cycle)
CMS T0streamer
rootFormatting+merge
1 stream ~ 2 MB/s
Uses LSF slots
~30 GB files

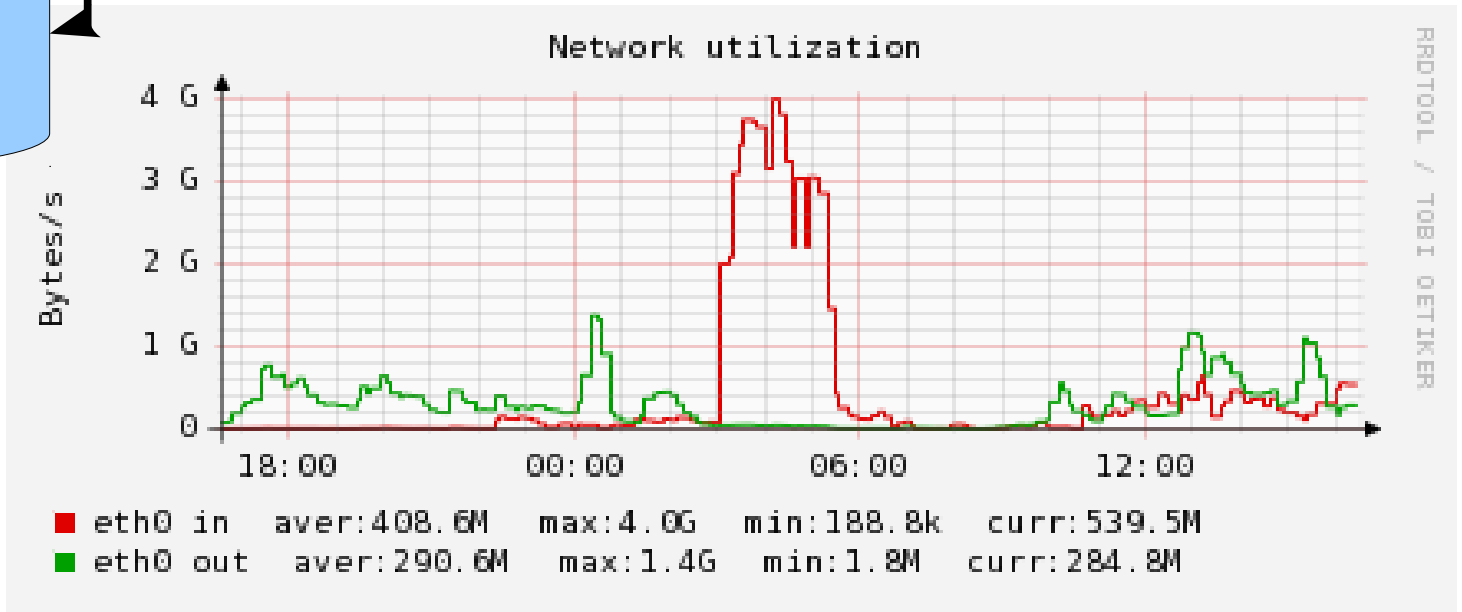
ALICE HI test

ALICE T0export

ALICE tapes

ALICE datadisk

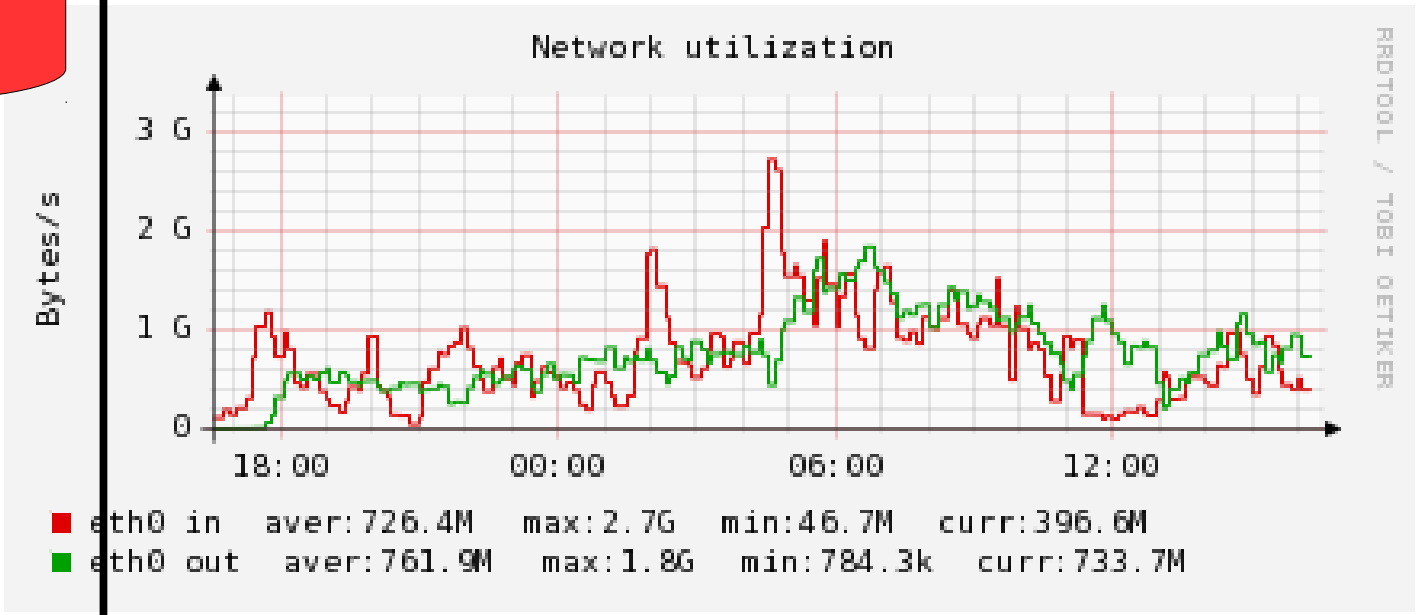




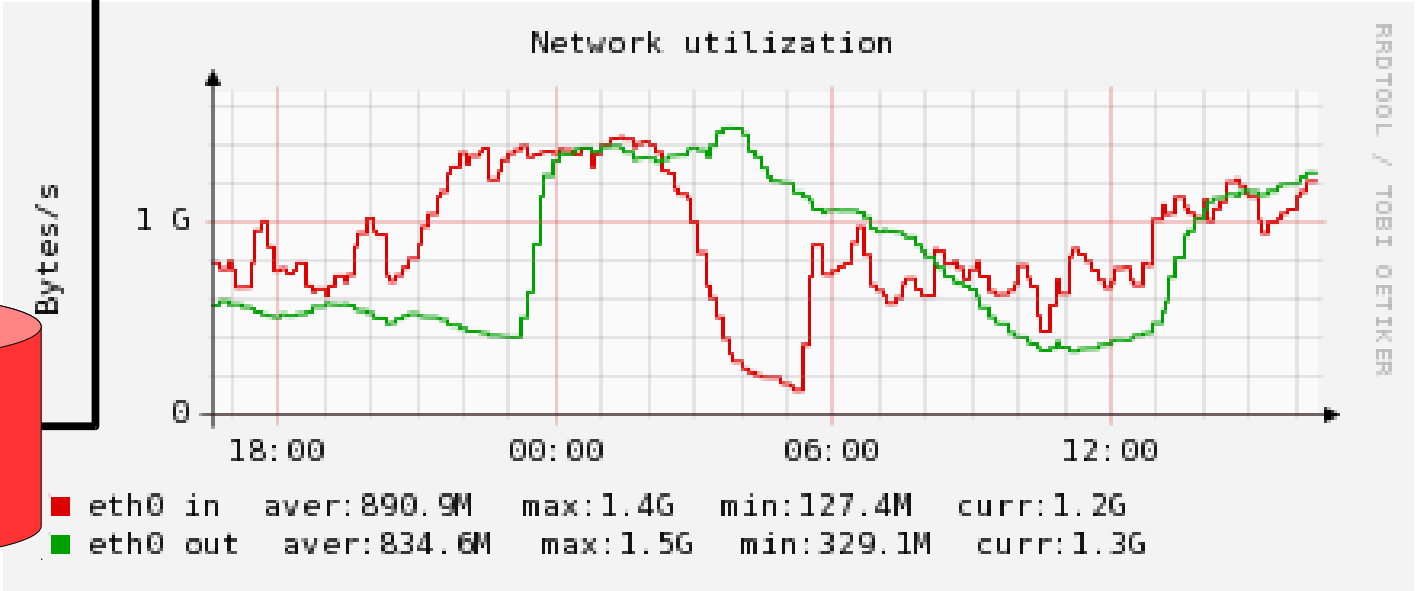
CMS HI test

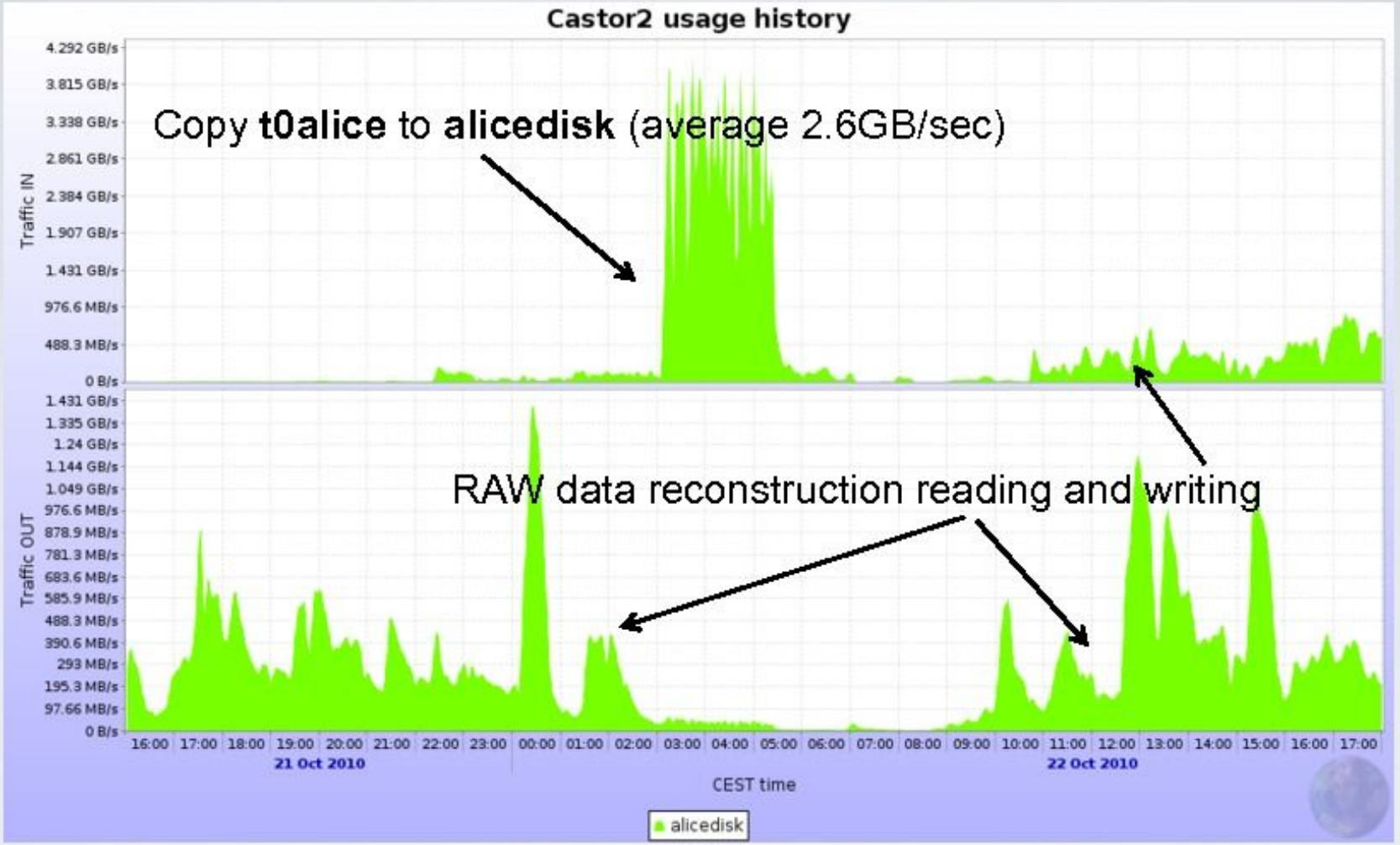
CMS T0export

CMS tapes



CMS T0streamer



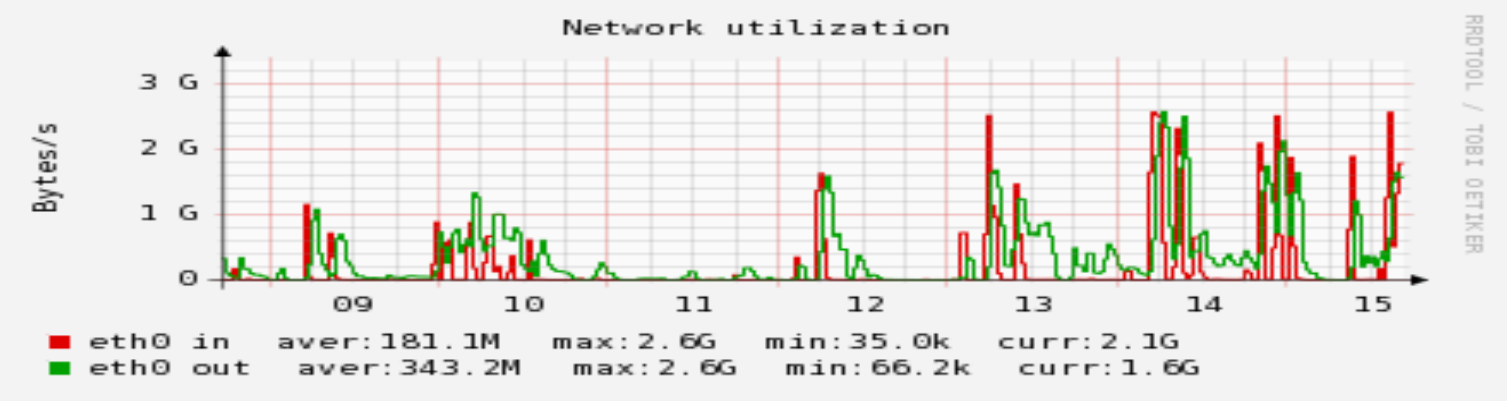


COPY t0alice → alicedisk + reco

Last week in CASTOR

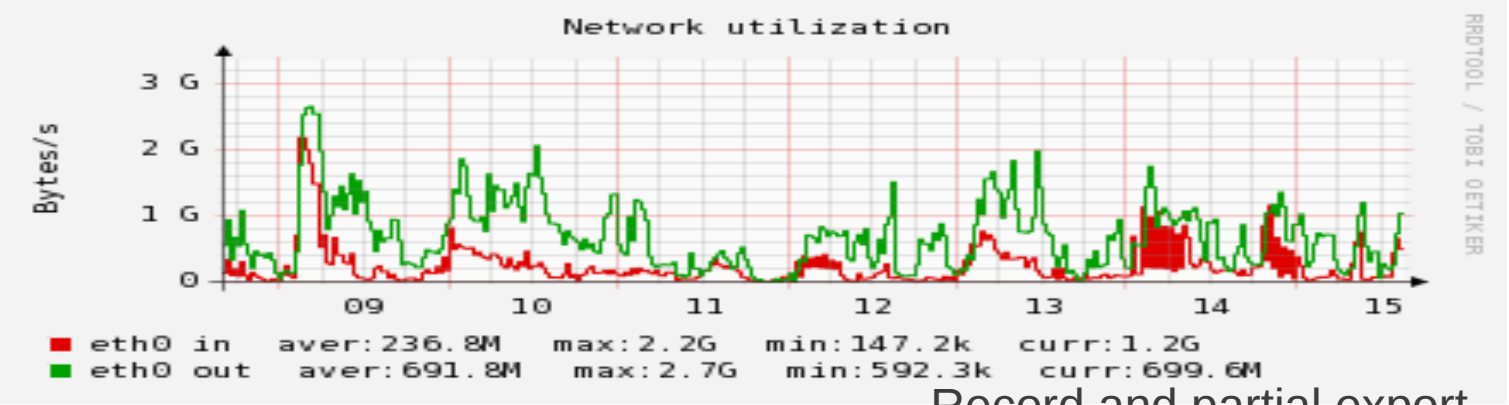
ALICE

Record and partial export



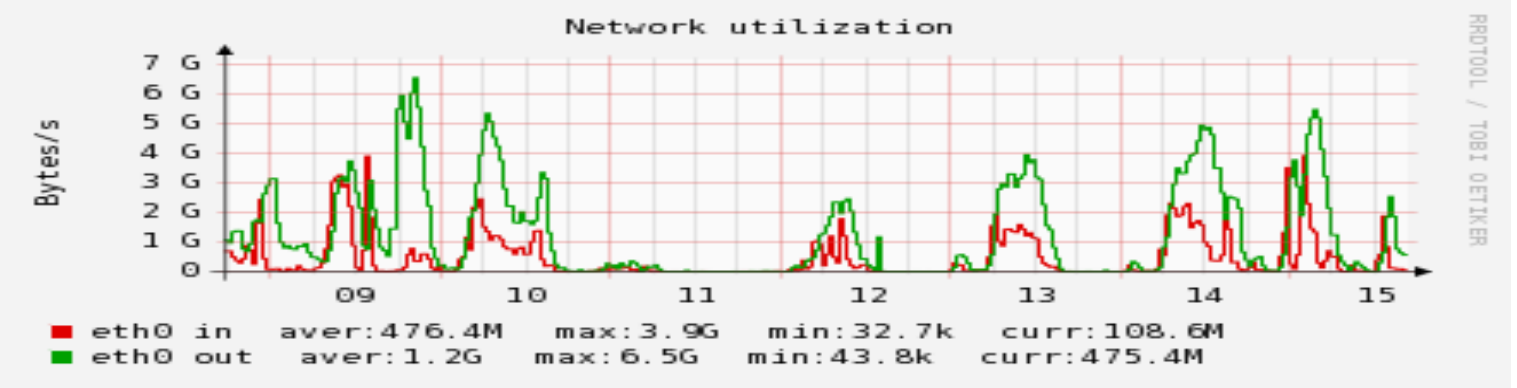
ATLAS

Record and export

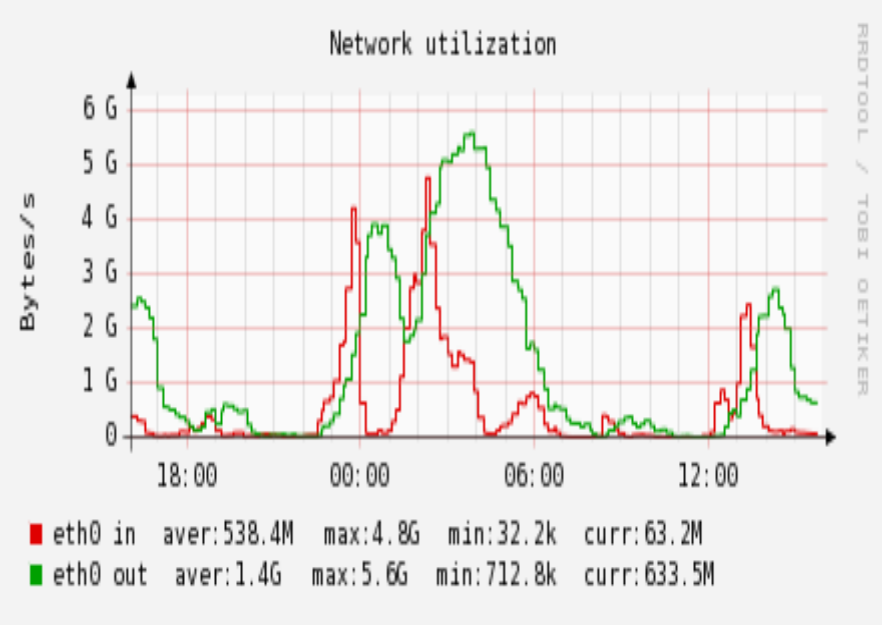


CMS

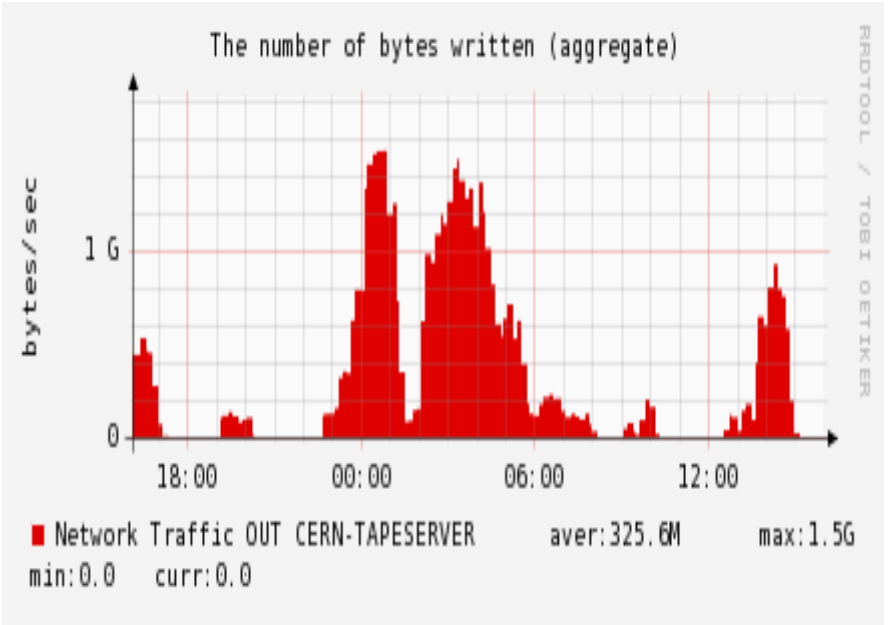
Record and partial export



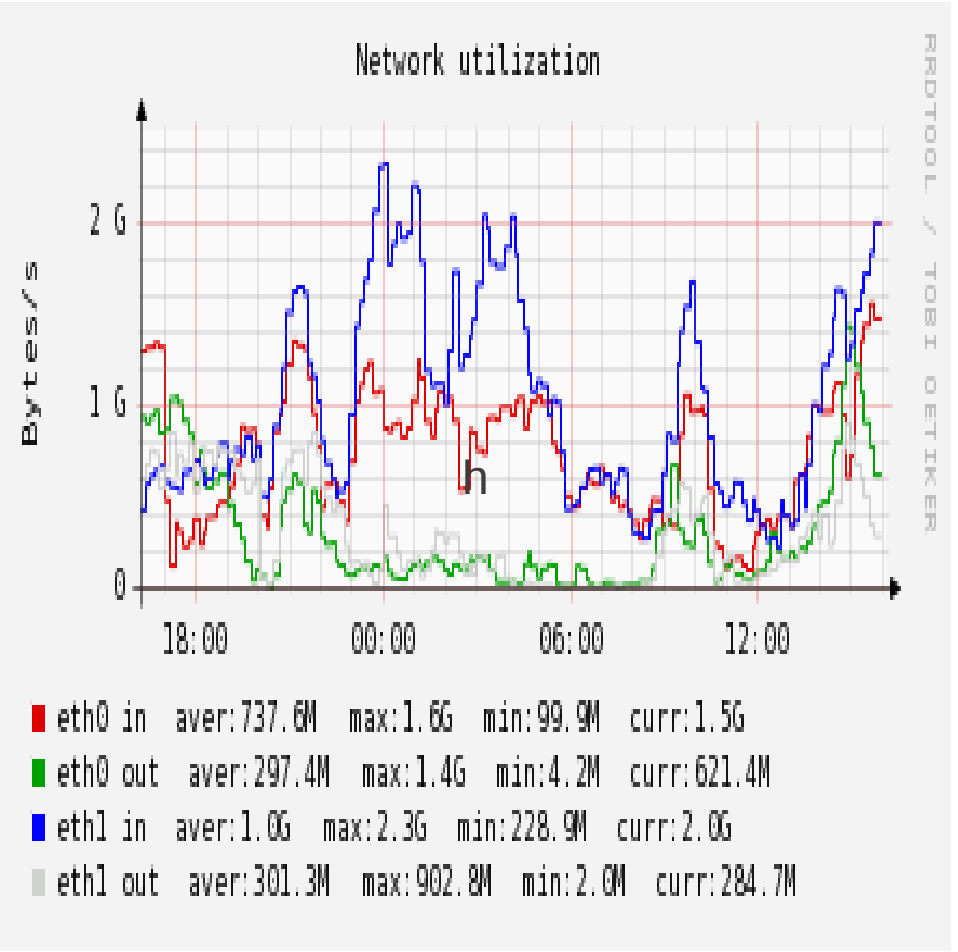
CMS pool activity (t0export)



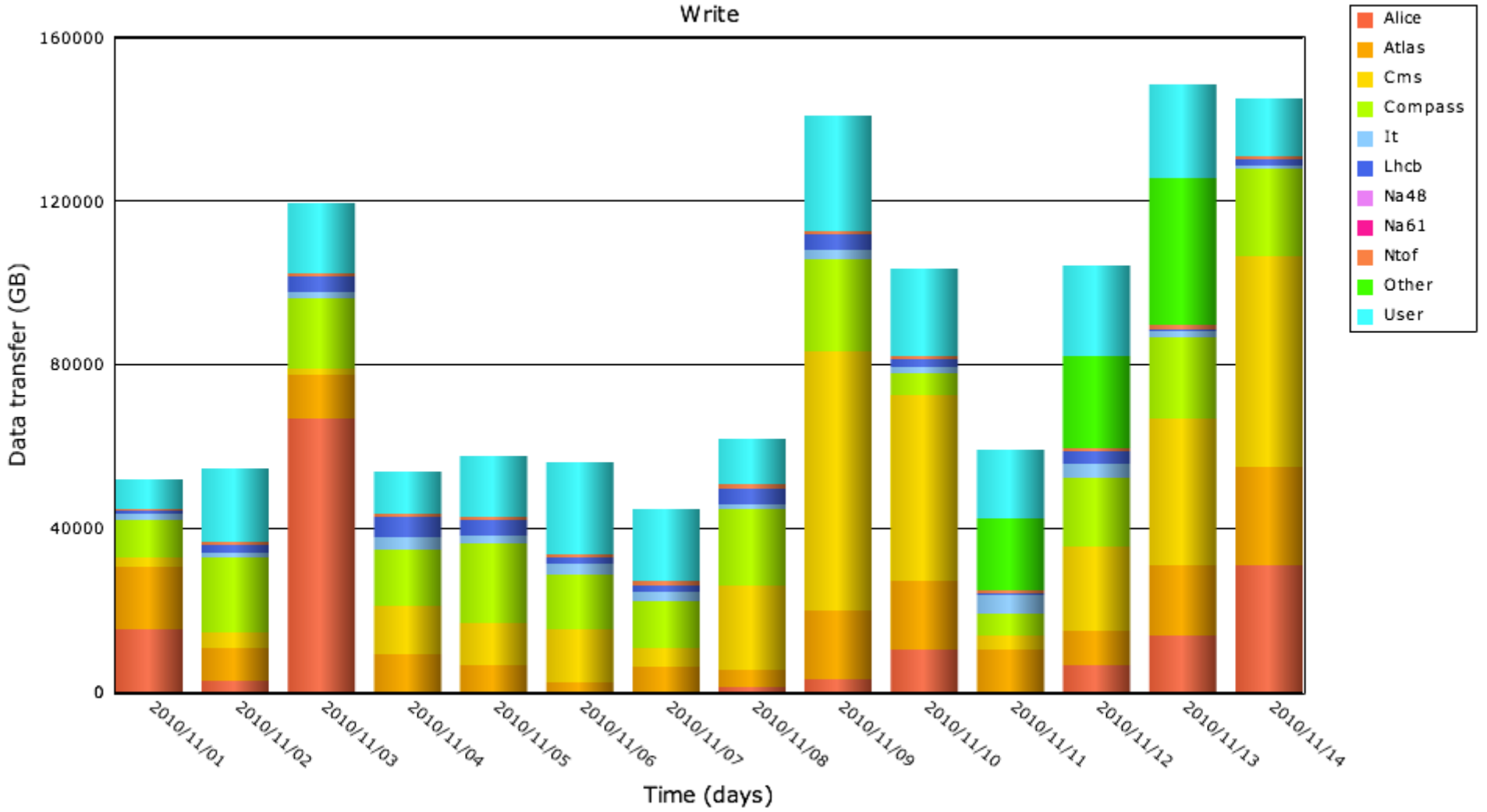
CMS pool activity (t0 to tape)



CASTOR tape infrastructure



HI Test → HI Run



From Tier0 to Analysis

- CASTOR is coping with the LHC data rate
  - Often substantially higher than expected!
  - And the machine is steadily improving!
- As expected, user analysis is increasing
  - Lot of the load on the Tier1/2/3 infrastructure
  - CERN?
- Analysis at CERN
  - Sizeable! (and growing!!!)
  - Potentially interfering with data taking (and other “organised” activities)
  - Moving from RFIO to xroot in some areas (especially analysis)

Requirements for analysis

- Multi PB facility
- RW file access (random and sequential reads, updates)
- Fast Hierarchical Namespace
  - Target capacity: 10^9 files, 10^6 containers (directories)
- Strong authorization
- Quotas
- Checksums
- Distributed redundancy of services and data
  - Dependability and durability
- Dynamic hardware pool scaling and replacement without downtimes
  - Operability

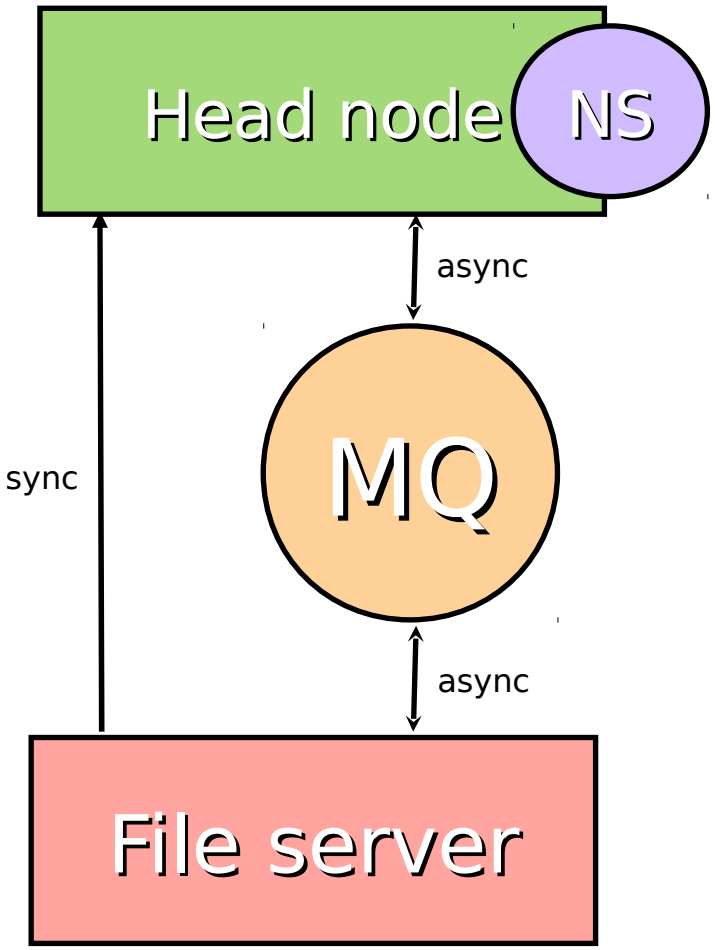
Starting points

- April 2010: storage discussions within the IT-DSS group
  - Prototyping/development started in May
- Input/discussion at the Daam workshop (June 17/18)
  - Demonstrators
  - Build on xroot strengths and know-how
- Prototype is under evaluation since August
  - Pilot user: ATLAS
  - Input from the CASTOR team (notably operations)
  - ATLAS Large Scale Test (pool of ~1.5 PB)
- Now being opened to ATLAS users
  - Ran by the CERN DSS-FDO operations team
- Still much work left to do
  - Good points:
    - **Early involvement of the users**
    - **Operations in the project from the beginning**
- This activity is what we call EOS

Selected features of EOS

- Is a set of XRootd plug-ins
  - And speaks XRoot protocol with you
- Just a Bunch Of Disks...
  - JBOD - no hardware RAID arrays
  - “Network RAID” within node groups
- Per-directory settings
  - Operations (and users) decide availability/performance (n. of replicas by directory – not physical placement)
    - One pool of disk – different classes of service
- Dependable and durable
  - Self-healing
  - “Asynchronous” operations (e.g. replace broken disks when “convenient” while the system keeps on running)

EOS Architecture



Head node

Namespace, Quota
Strong Authentication
Capability Engine
File Placement
File Location

Message Queue

Service State Messages
File Transaction Reports

File Server

File & File Meta Data Store
Capability Authorization
Checksumming & Verification
Disk Error Detection (Scrubbing)

EOS Namespace

Version 1 (current)

In-memory hierarchical namespace using Google hash

Stored on disk as a changelog file

Rebuilt in memory on startup

Two views:

- hierarchical view (directory view)
- view storage location (filesystem view)

very fast, but limited by the size of memory

- 1GB = ~1M files

Version 2 (under development)

Only view index in memory

Metadata read from disk/buffer cache

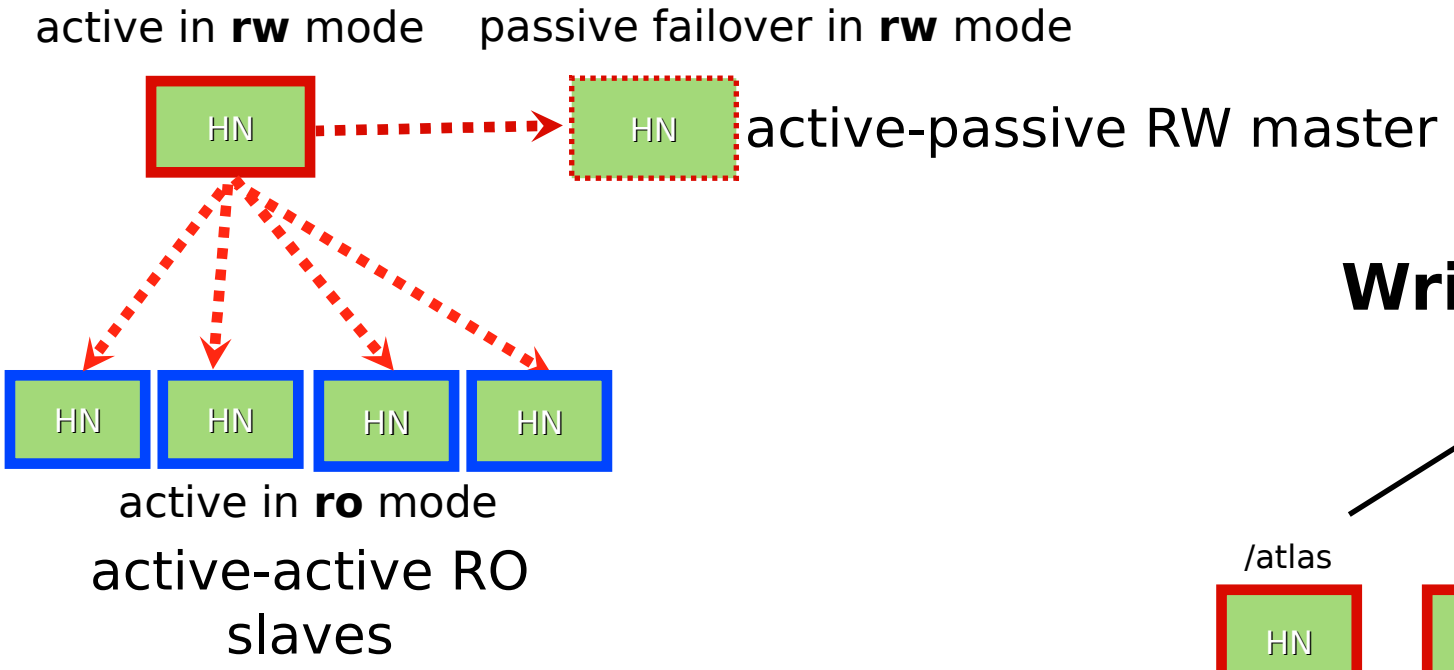
Perfect use case for SSDs (need random IOPS)

10^9 files = ~20GB per view

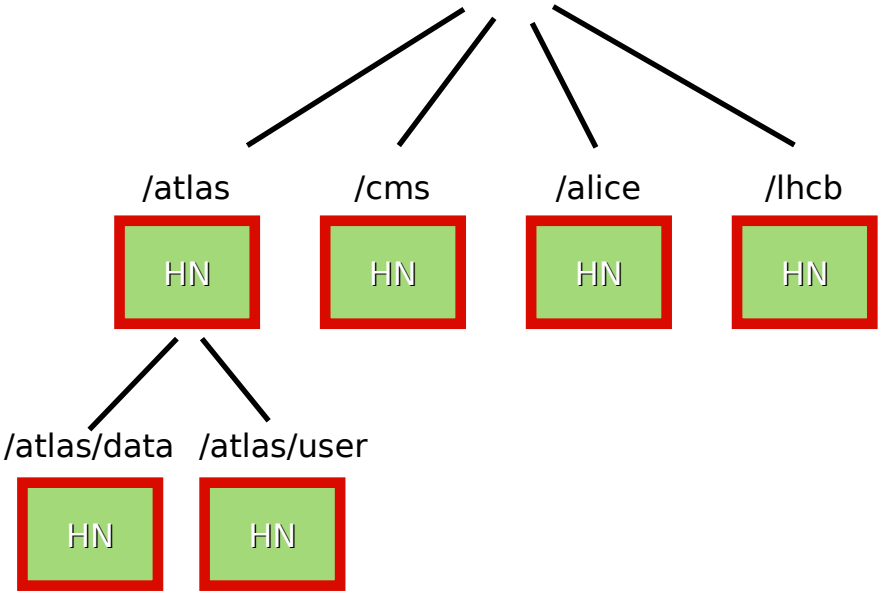
Namespace	V1	V2*
Inode Scale	100 M inodes	1000 M inodes
In-Memory Size	80-100 GB (replicas have minor space contribution)	20 GB x n(replica)
Boot Time	~520 s **	15-30 min ** (difficult to guess)
Pool size assuming avg. 10 Mb/file + 2 replicas	2 PB	20 PB
Pool Nodes assuming 40 TB/node	50	500
File Systems assuming 20 / node	1.000	10.000

High Availability - Namespace scaling

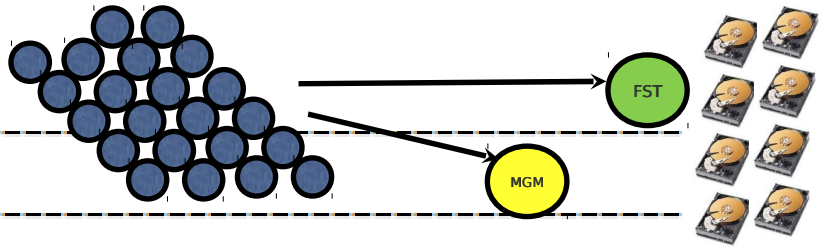
HA & Read Scale out



Write Scale out



File creation test



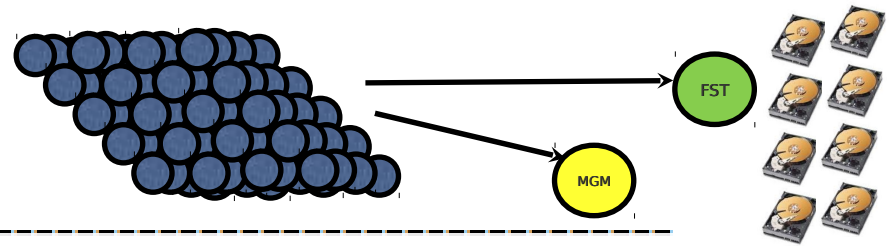

```
EOS Console [root://localhost] |/> ns stat
# ------------------------------------------------------------------------------------
# Namespace Statistic
# ------------------------------------------------------------------------------------
ALL      Files                            5008898
ALL      Directories                      5073
# ------------------------------------------------------------------------------------
who      command                          sum          5s      1min      5min       1h
# ------------------------------------------------------------------------------------
ALL      Commit                           5006939    926.00   1104.64   1054.88   914.63
ALL      Exists                           5007435    926.00   1104.66   1054.88   914.63
ALL      Find                             0000005      0.00      0.00      0.00     0.00
ALL      Mkdir                            0005022      1.25      1.10      1.05     0.92
ALL      Open                             5007195    926.00   1104.66   1054.88   914.63
ALL      OpenDir                          0000010      0.00      0.00      0.00     0.00
ALL      OpenFailedQuota                  0000240      0.00      0.00      0.00     0.00
ALL      OpenProc                         0000151      0.25      0.03      0.01     0.02
ALL      OpenWriteCreate                  5006955    926.00   1104.66   1054.88   914.63
ALL      OpenWriteTruncate                0000240      0.00      0.00      0.00     0.00
ALL      Rm                               0000240      0.00      0.00      0.00     0.00
ALL      Stat                             0000413      0.00      0.00      0.00     0.11
```

1 kHz

NS Size: 10 Mio Files
* 22 ROOT clients 1 kHz
* 1 ROOT client 220 Hz

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+  COMMAND
23681 daemon    20   0 8173m 7.9g 4356 S     0   6.3   0:48.55 xrootd
```

File read test

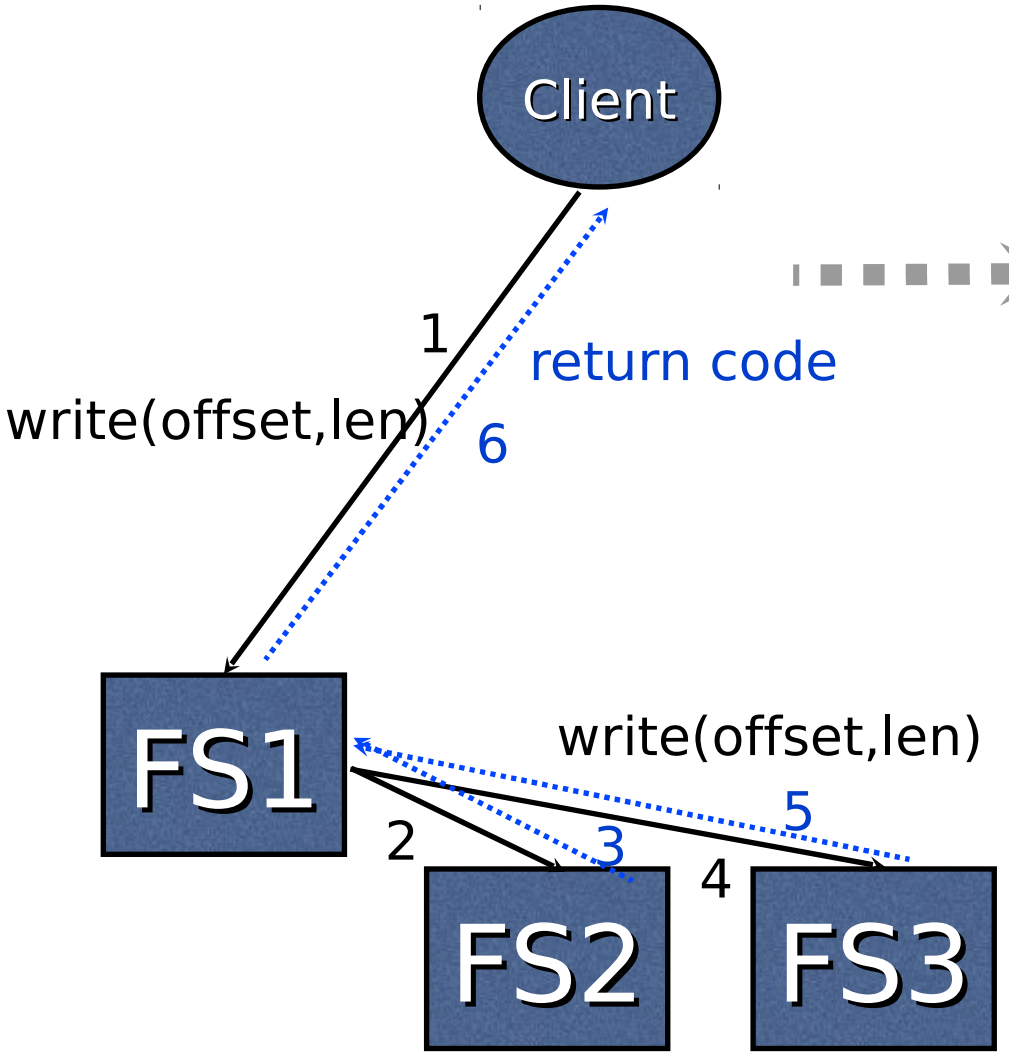



```
EOS Console [root://localhost] |/> ns stat
# ------------------------------------------------------------------------------------
# Namespace Statistic
# ------------------------------------------------------------------------------------
ALL      Files                            10079235
ALL      Directories                      10139
# ------------------------------------------------------------------------------------
who      command                          sum          5s      1min      5min       1h
# ------------------------------------------------------------------------------------
ALL      Commit                           0001742      0.00      0.00      0.00      0.00
ALL      Exists                           0003220      0.00      0.00      0.00      0.00
ALL      Open                             100069124    6785.00   6764.90   6697.50   7096.28
ALL      OpenFailedQuota                  16645985       0.00      0.00      0.00    751.41
ALL      OpenProc                         0000527      0.50      0.19      0.05      0.02
ALL      OpenRead                         100066929    6784.75   6764.90   6697.50   7096.28
ALL      OpenWriteCreate                  0000262      0.00      0.00      0.00      0.00
ALL      OpenWriteTruncate                0001479      0.00      0.00      0.00      0.00
ALL      Rm                               0001479      0.00      0.00      0.00      0.00
```

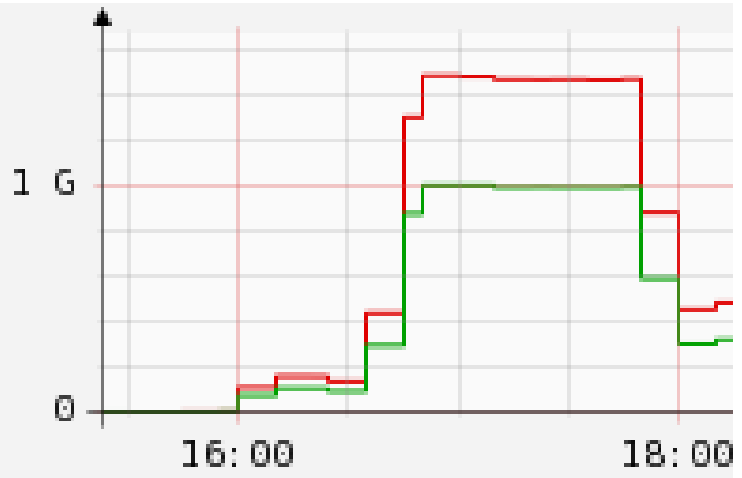
7 kHz

NS Size: 10 Mio Files
* 100 Million read open
* 350 ROOT clients 7 kHz
* CPU usage 20%

Replica layout



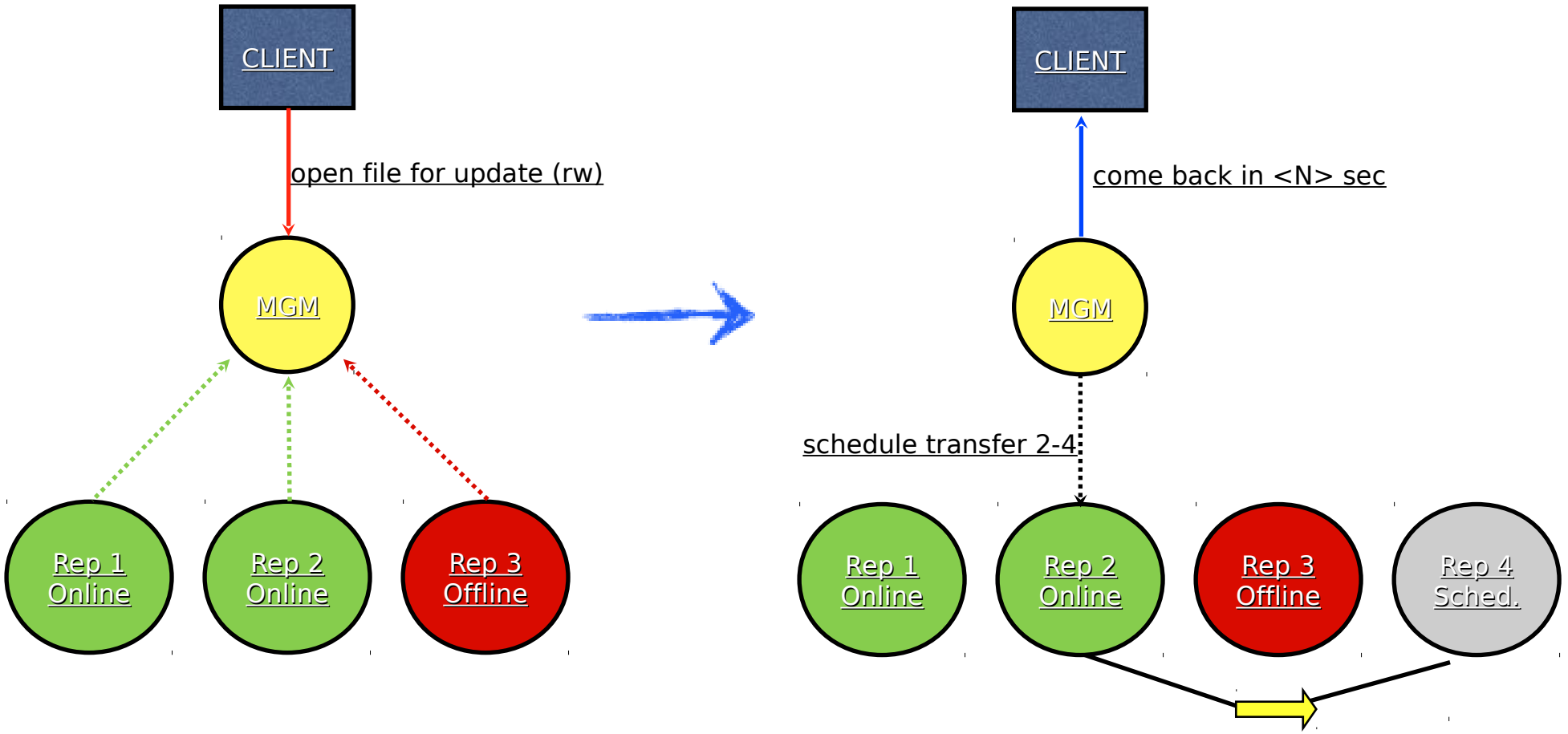
Network IO for file creations with 3 replicas:



500 MB/s injection result in
- 1 GB/s output on eth0 of all disk servers
- 1.5 GB/s input on eth0 of all disk servers

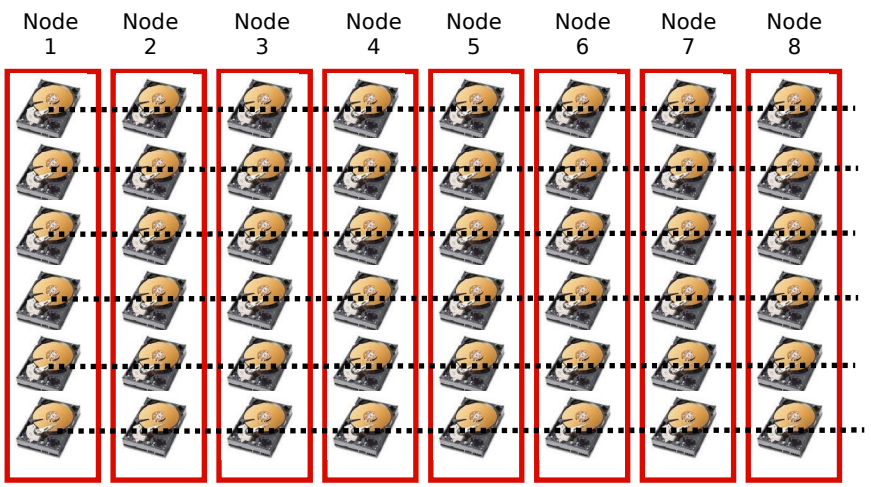
Plain (no replica)
Replica (here 3 replicas)
More sophisticated redundant storage (RAID5, LDPC)

Replica healing



Client **RW** reopen of an existing file triggers
- creation of a new replica
- dropping of offline replica

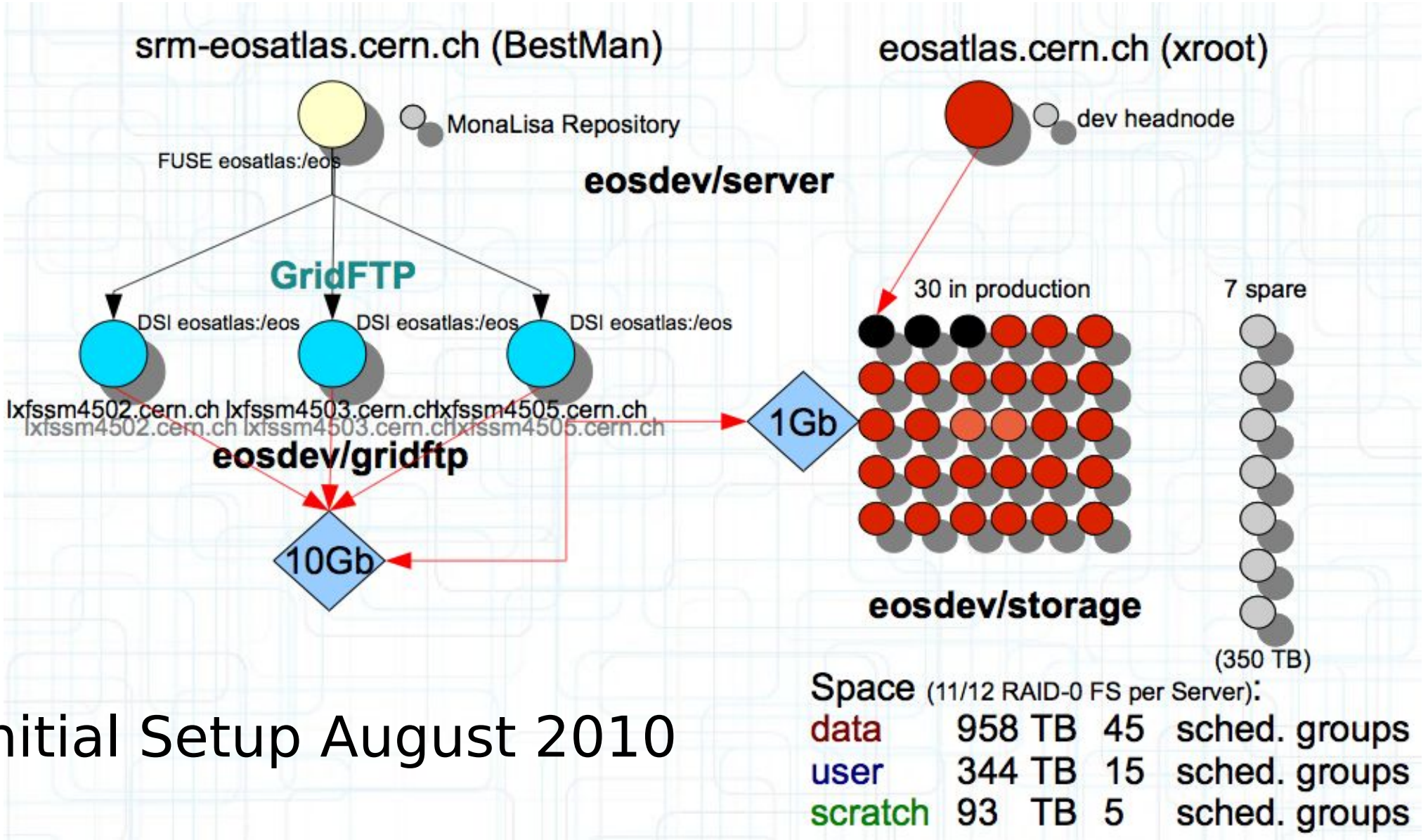
Replica placement



In order to minimize the risk of data loss we couple disks into scheduling groups (current default is 8 disks per group)

- The system selects a scheduling group to store a file in in a round-robin
- All the other replicas of this file are stored within the same group
  - Data placement optimised vs hardware layout (PC boxes, network infrastructure, etc...)

EOS ATLAS test (LST)

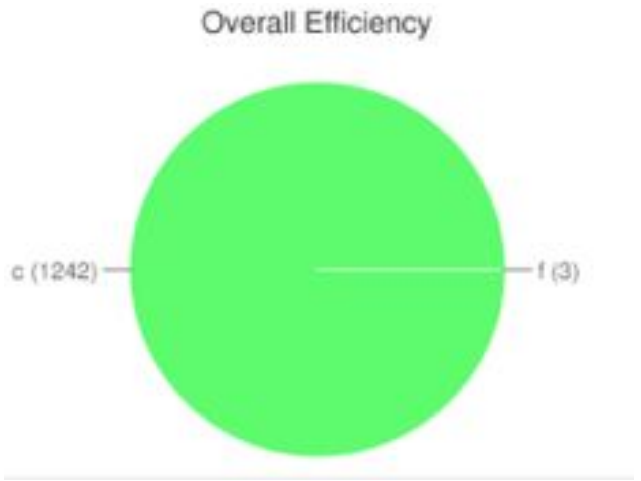


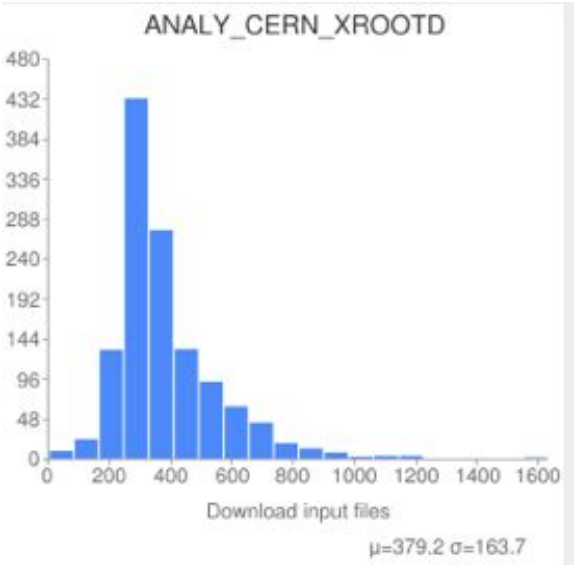
Initial Setup August 2010

Hammer Cloud Test

HC 10001181

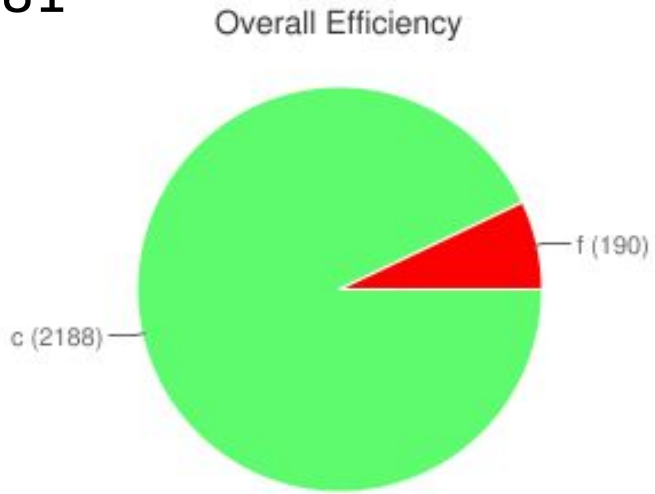
EOSATLAS POOL 27 Disk Server RAID0

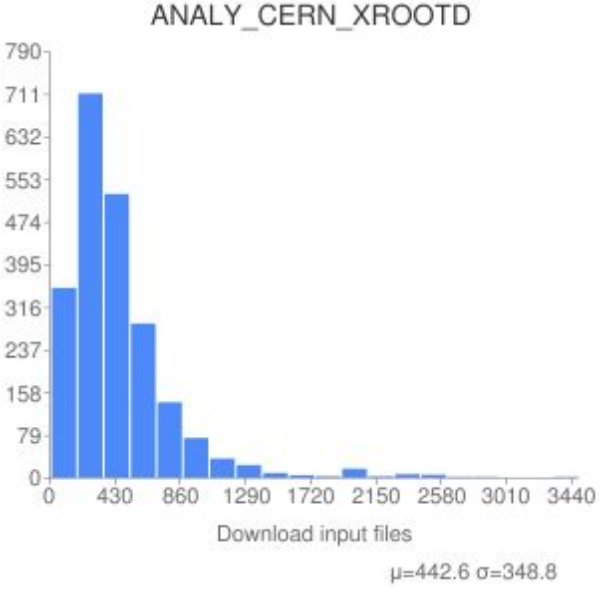




HC 10001381

EOSATLAS POOL reconfigured to 8 Disk Server JBOD

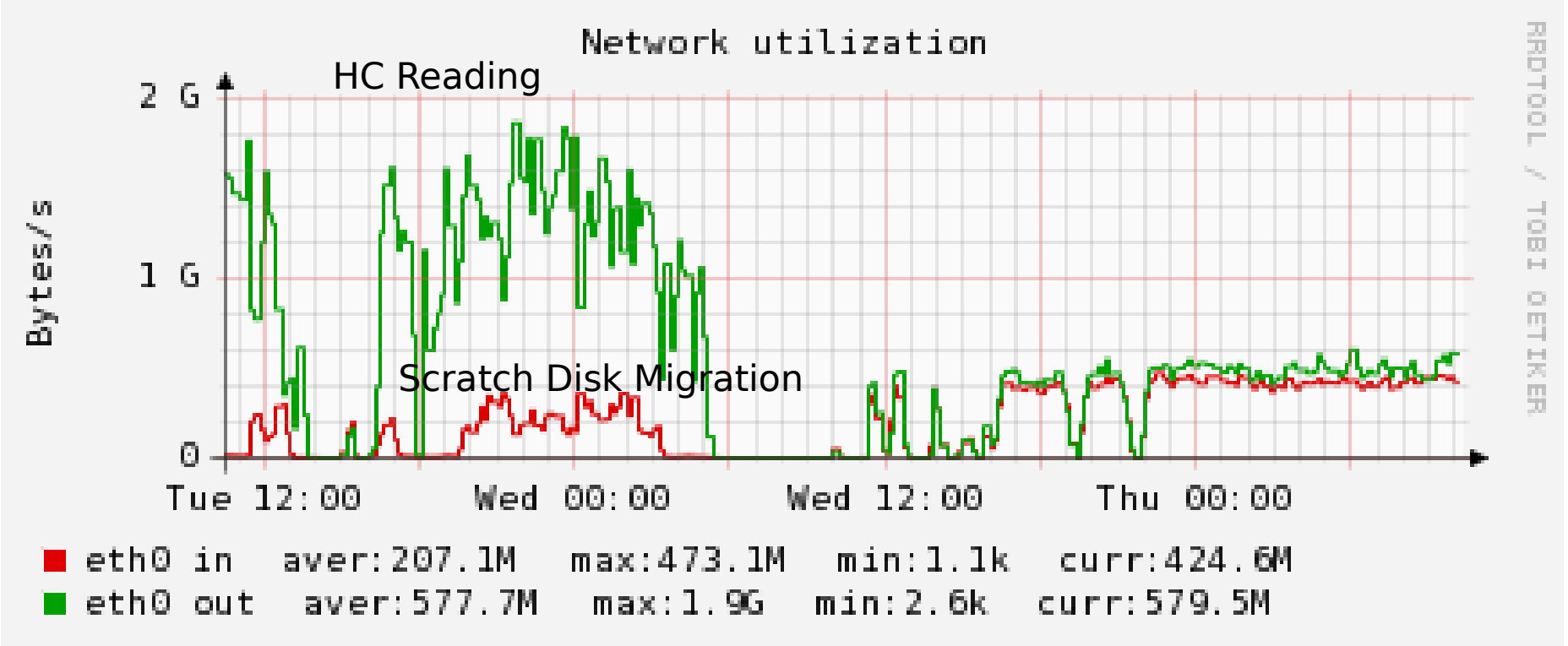




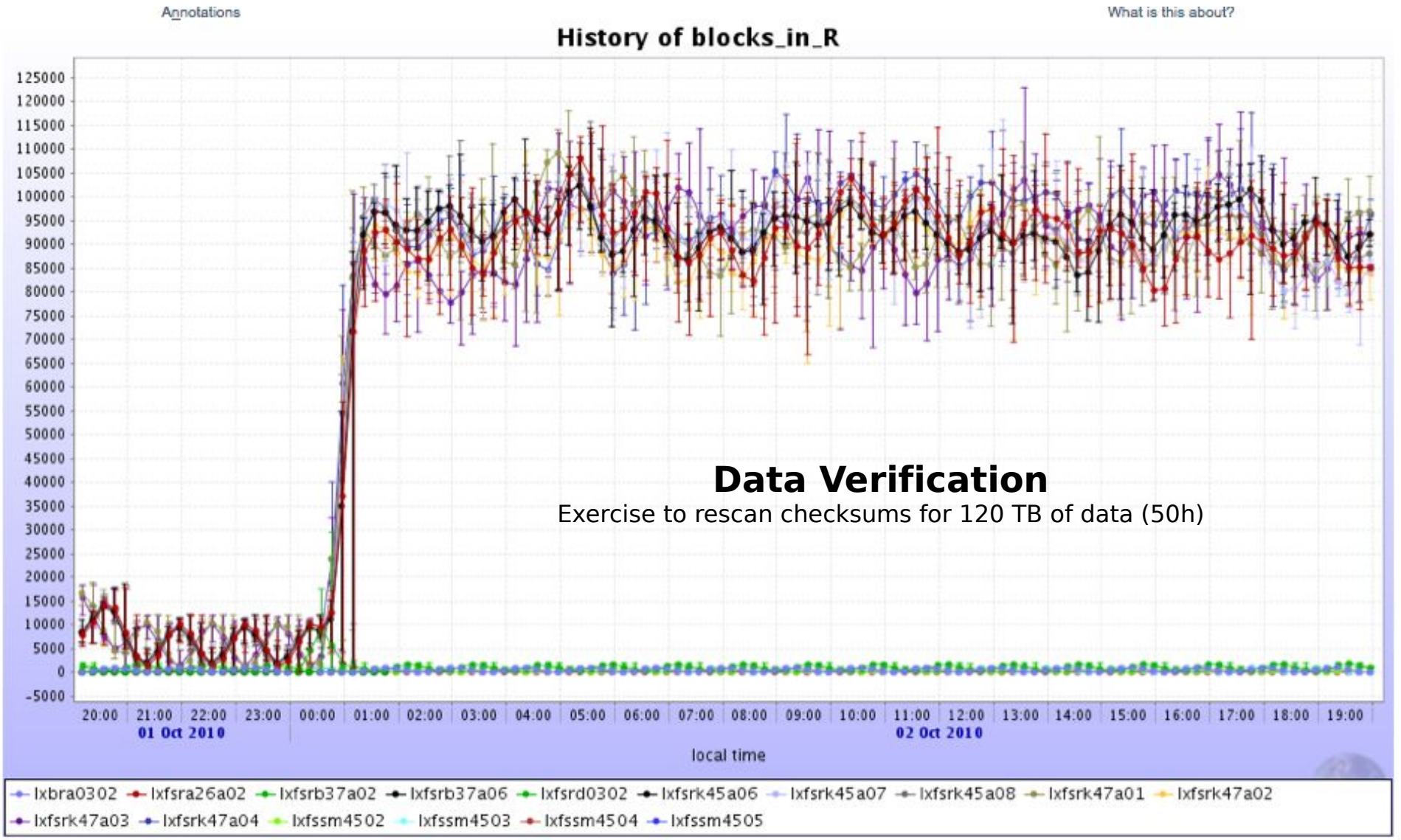
Hardware replacement test

Life Cycle Management

Exercise to migrate
27 disk server with 10 Raid-0 FS to 8 new with 20 JBOD FS
[partially overlapped with HC Tests]



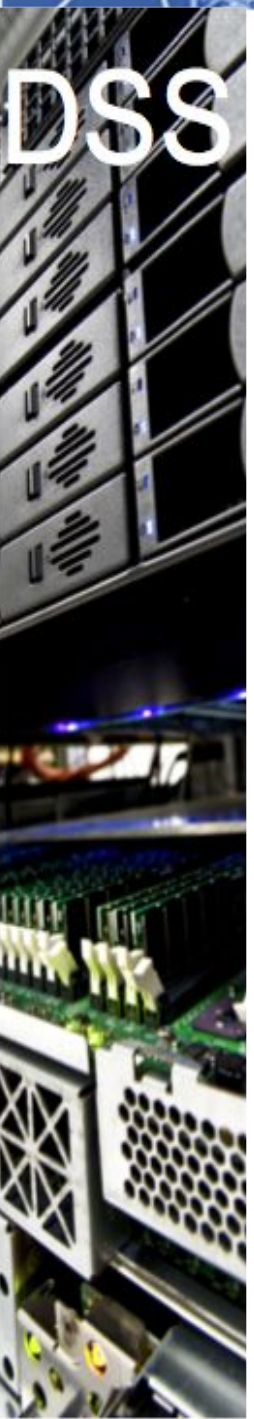
Data verification test

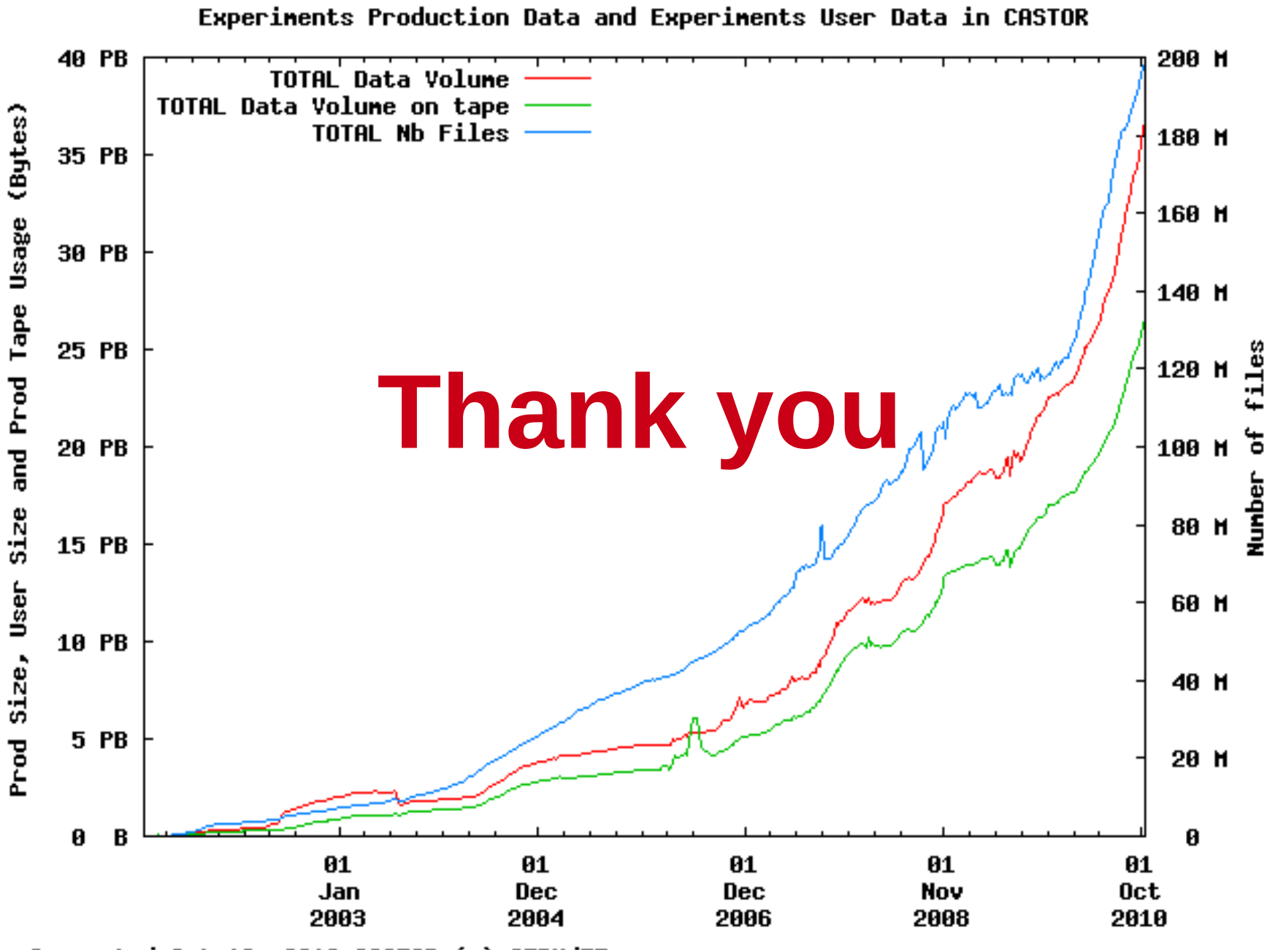


Plans

- Operation
  - Opening EOS to ATLAS users on Nov 15
  - Run by IT DSS operations team
- Development
  - Implement Version 2 of the namespace
- Field testing
  - EOS is one end point in ATLAS
    - Managed by IT operations
    - Used by ATLAS users and looked after by ATLAS shifts (as any Tier1/2/(3))
- Larger Testbed (if available)
  - Scale the instance from todays 600 disks to 2,000-8,000 disks (4 – 16 PB)

Remark: could also support NFS4.1 protocol - implementation is running inside xrootd servers but files can be accessed with any protocol supporting client stalling & redirection


DSS


Experiments Production Data and Experiments User Data in CASTOR
TOTAL Data Volume
TOTAL Data Volume on tape
TOTAL Nb Files
Prod Size, User Size and Prod Tape Usage (Bytes)
Number of files
40 PB
35 PB
30 PB
25 PB
20 PB
15 PB
10 PB
5 PB
0 B
200 M
180 M
160 M
140 M
120 M
100 M
80 M
60 M
40 M
20 M
0
01 Jan 2003
01 Dec 2004
01 Dec 2006
01 Nov 2008
01 Oct 2010

Thank you