DSS

2011+

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

ATLAS retreat Napoli 2-4 Jan 2011

Introduction

- IT DSS: responsible of the data management for physics at CERN
  - Mainly but not exclusively the LHC experiments
  - Physics data (on disk and tape): notably AFS, CASTOR and EOS
- **Production services** but **not** a steady-state situation
  - Technology evolves
  - New ideas are being discussed within HEP
    - Includes rediscovering of “old” ideas
      - “All relevant data on disk” part of the computing model back in 1999 (Hoffmann review)
  - Experience from the LHC data taking
    - 10+ M files per month
    - Times 3 in total size in the next few years (40 PB → 120 PB in 2015)
      - **Learn more about your 2011 plans (rates, data model, etc...)**
    - Real data are more interesting than MonteCarlo: users, users, users, users
    - Master operational load!

How we proceeded so far

- Bringing EOS alive
  - Integrating best ideas, technology and experience
  - Refocus on experiments requirements
  - Expose developments at early stage
  - Operations on board at early stage
- First experience: LST (with ATLAS)
  - Similar activity starting with CMS
  - And interest from others
  - Some similarities with Tier3/Analysis Facilities approaches

Next steps (2011)

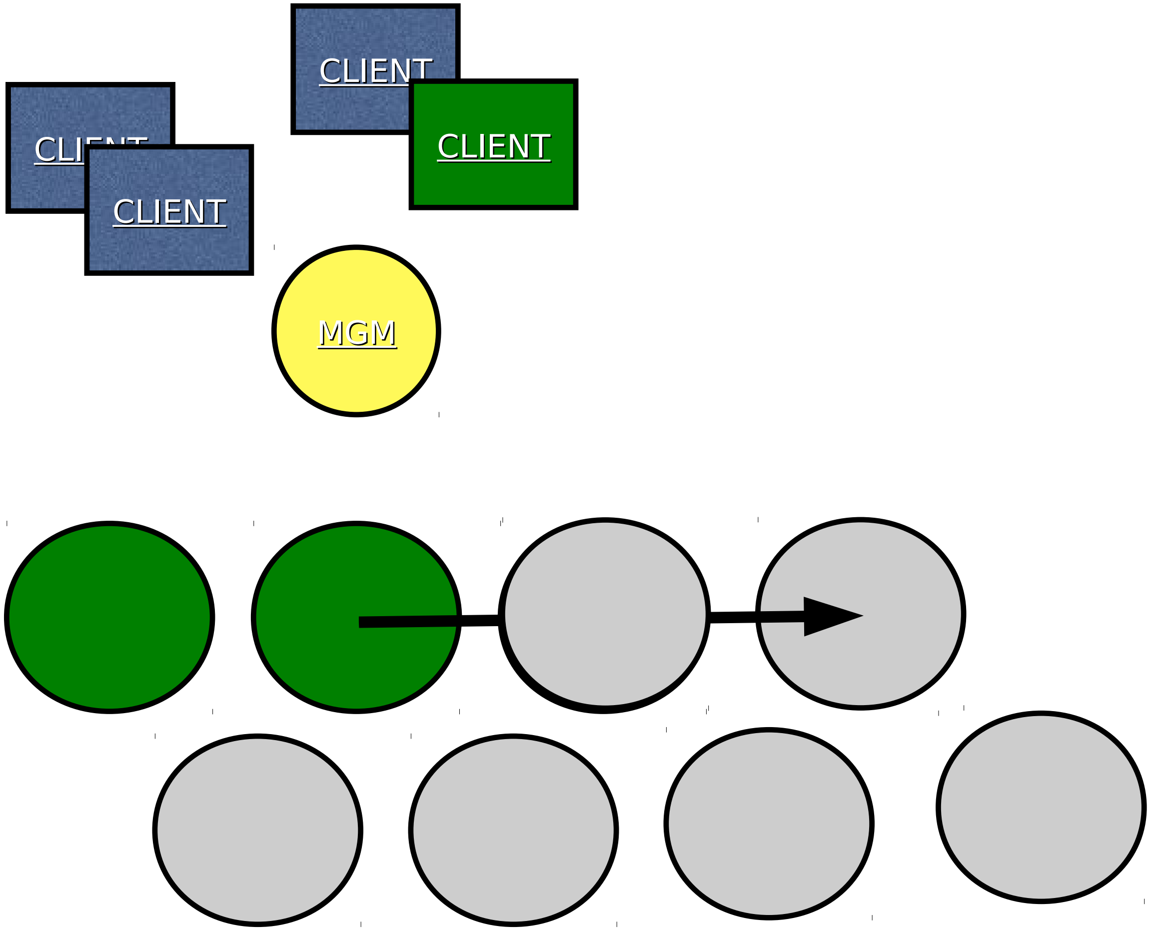
- Software
  - EOS consolidation (April)
- Operations
  - Set up EOS as a service (streamlined ops effort)
  - Steady process:
    - Continue to build on CASTOR experience (procedures, etc...)
    - Simplifying the overall model and procedures
      - **In a nutshell: no need of synchronous (human) intervention**
  - More monitoring (including user monitoring)
- Deployment
  - **D1 area**
  - Dependable Serviceable High-Performance High-Concurrency Multi-User Data Access
    - Analysis use cases

Immediate steps

- Streamlining of the space token (disk pools)
  - Less (but larger) entities (“disk pools”)
  - Share different approaches/experiences
- Reduction of duplication/optimisation of resources
  - Analysis areas (CASTOR → EOS)
  - Hot-disk areas
    - Interplay with CERNVMF data will be understood
  - Replication per directory
    - Fine-grained control of reliability, availability and performance
- Continuing to “protect” tape resources from users :)
  - CMS is looking into the ATLAS model
  - Successful implementation depends on experiment support
- More transparency (monitoring and user-supporting activities)
  - service-now migration
  - Keep users aware of what is going on (not only the ADC experts) will help our 1st/2nd/3rd lines support

Replica healing (3-replica case)

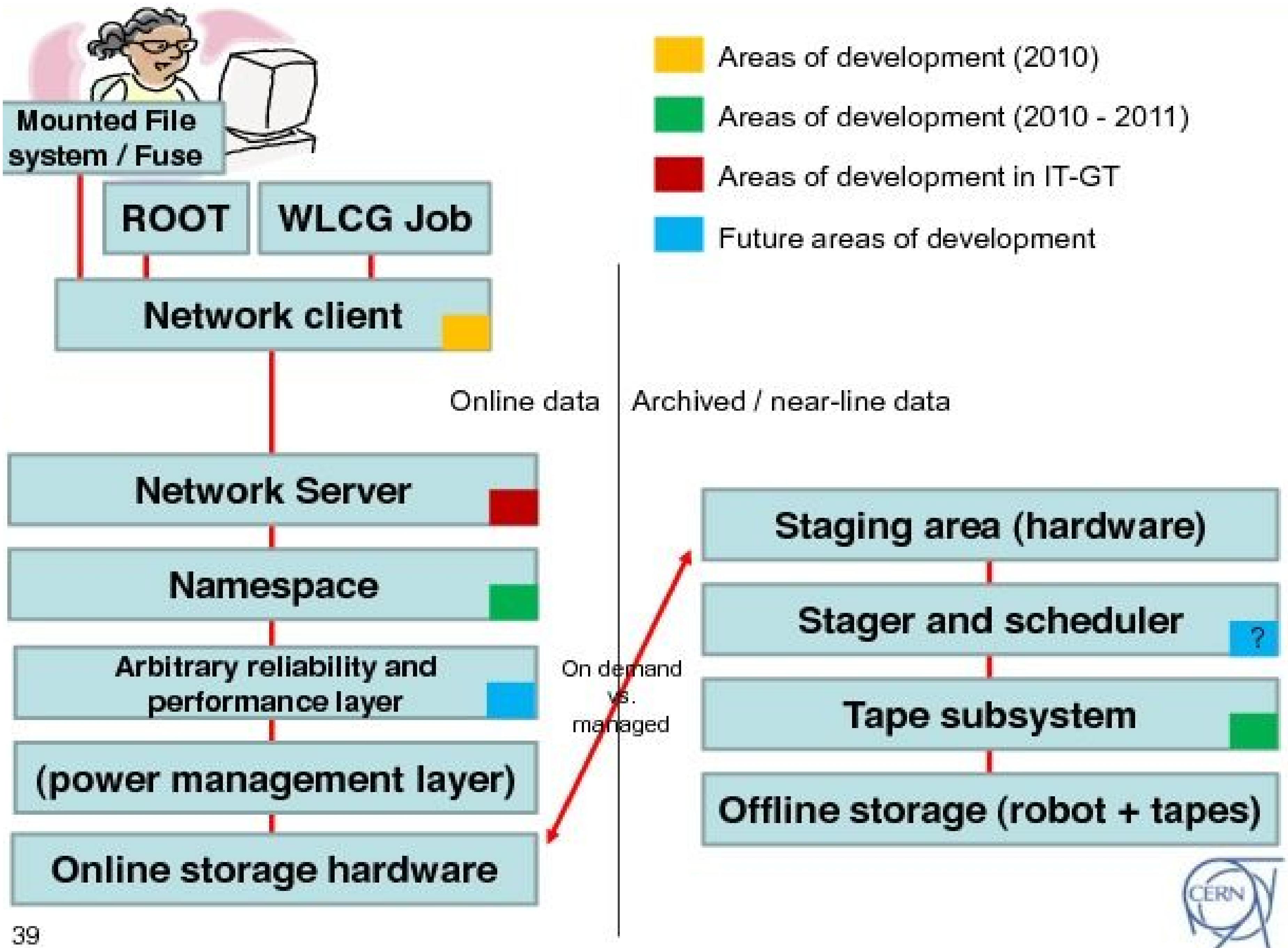
asynchronous operations (failure/draining/rebalancing/...)



The intervention on the failing box can be done when appropriate (asynchronously) because the system re-establishes the foreseen number of copies. The machine carries no state (it can be reinstalled from scratch and the disk cleaned as in the batch node case). All circles represent a machine in production

Future developments (example) availability and reliability

- **Plain (reliability of the service = reliability of the hardware)**
  - Example: a CASTOR disk server running in RAID1 (Mirroring: n=1, m=1 S=1+m/n)
  - Since the two copies belong to the same PC box, availability can be a problem
- **Replication**
  - Reliable, maximum performance, but heavy storage overhead
  - Example: n=1, m=2; S = 1+ 200% (EOS)
- **Reed-Solomon, double, triple parity, NetRaid5, NetRaid6**
  - Maximum reliability, minimum storage overhead
  - Example (n+m) = 10+3, can lose any 3 out of 13
  - S = 1 + 30%
  - EOS/RAID5 prototype being evaluated internally
- **Low Density Parity Check (LDPC) / Fountain Codes**
  - Excellent performance, more storage overhead but better than
  - Example: (n+m)=8+6, can lose any 3 out 14 (S=1+75%)


Mounted File system / Fuse
ROOT
WLCG Job
Network client
Online data
Archived / near-line data
Network Server
Namespace
Arbitrary reliability and performance layer
(power management layer)
Online storage hardware
On demand vs. managed
Staging area (hardware)
Stager and scheduler
?
Tape subsystem
Offline storage (robot + tapes)
Areas of development (2010)
Areas of development (2010 - 2011)
Areas of development in IT-GT
Future areas of development
39

EOS **CASTOR**

Spare (for discussion)

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

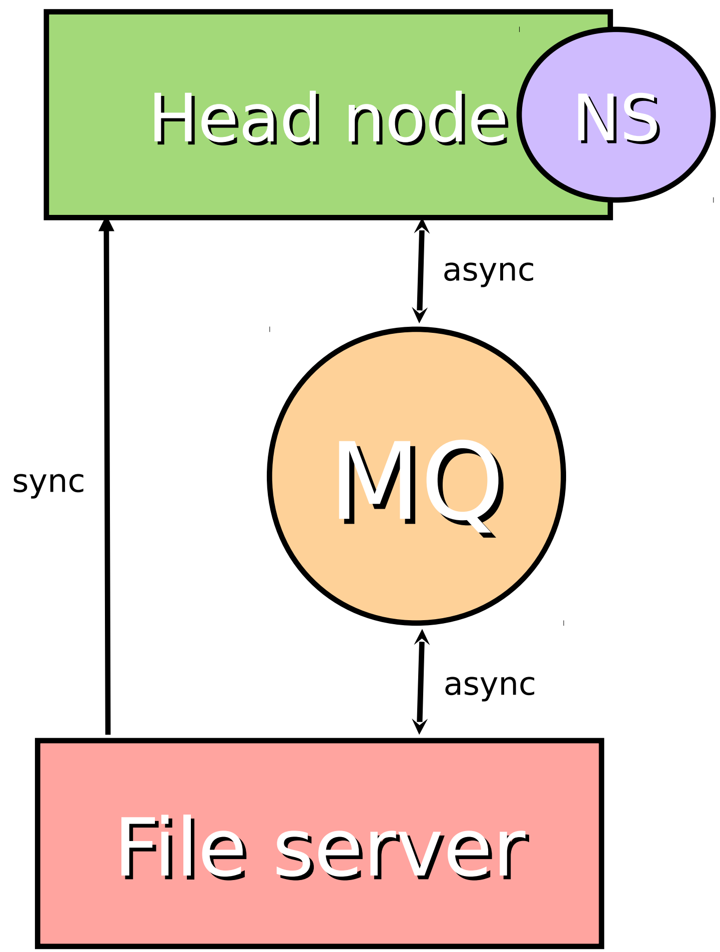
Starting points

- 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 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

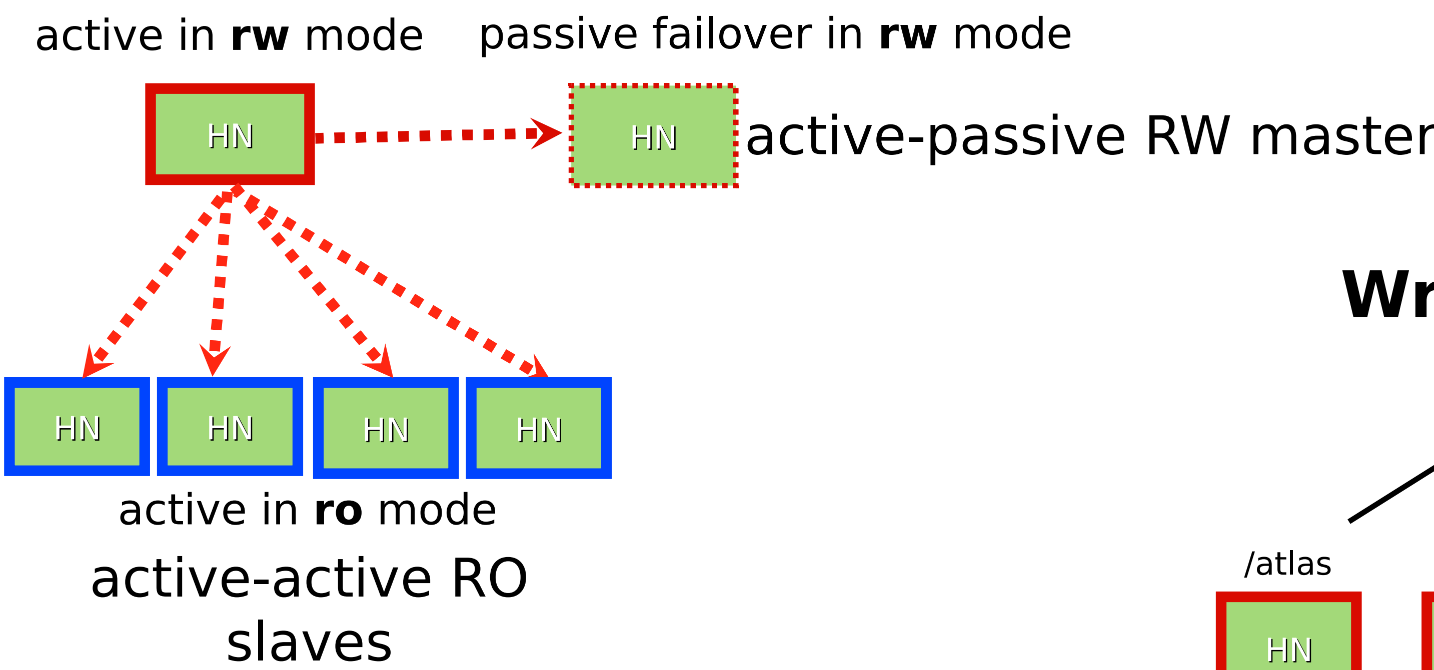
Perfect use case for SSDs (need random IOPS)

10^9 files = ~20GB per view

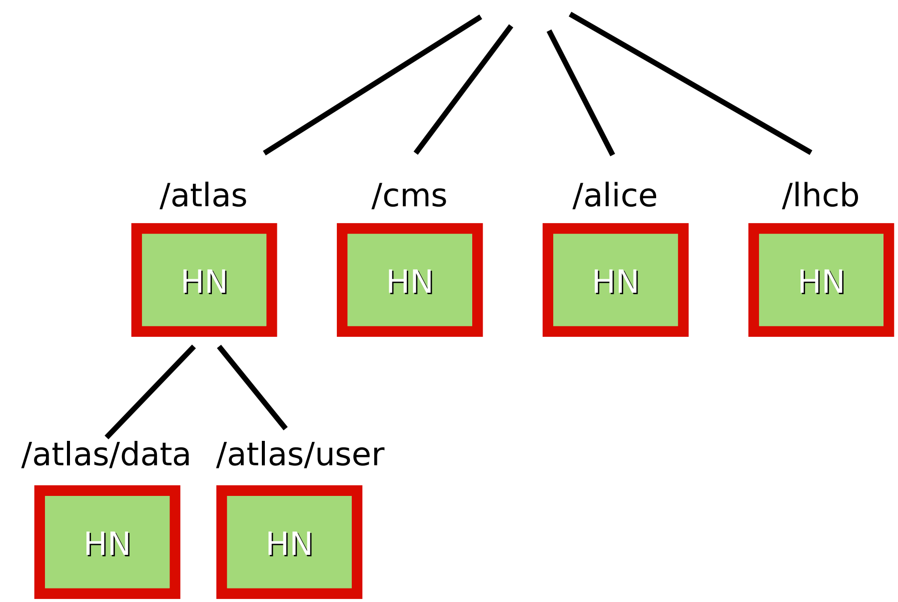
Namespace	V1	V2*
Inode Scale	100 M inodes	1000 M inodes
In-Memory Size	80-100 GB (replicas have minor space contribution)	20 GB x n(replica)
Boot Time	~520 s **	15-30 min ** (difficult to guess)
Pool size assuming avg. 10 Mb/file + 2 replicas	2 PB	20 PB
Pool Nodes assuming 40 TB/node	50	500
File Systems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

High Availability - Namespace scaling

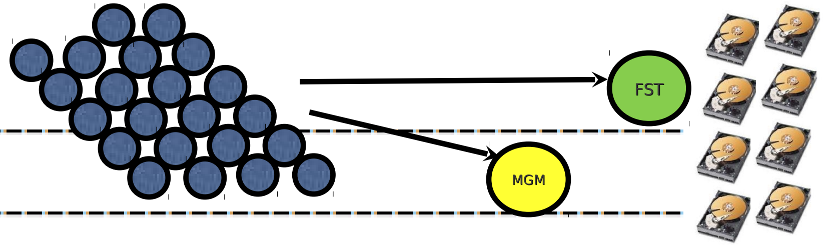
HA & Read Scale out



Write Scale out



File creation test



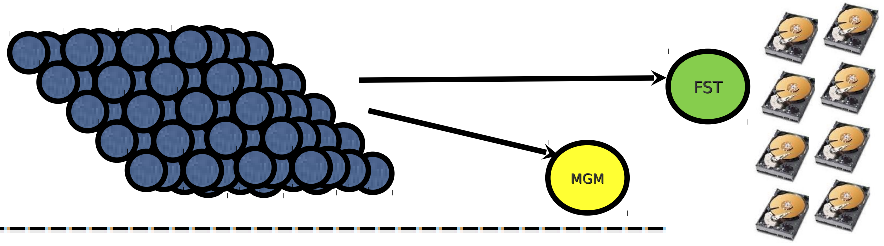

```
EOS Console [root://localhost] |/> ns stat
# ------------------------------------------------------------------------------------
# Namespace Statistic
# ------------------------------------------------------------------------------------
ALL      Files                            5008898
ALL      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

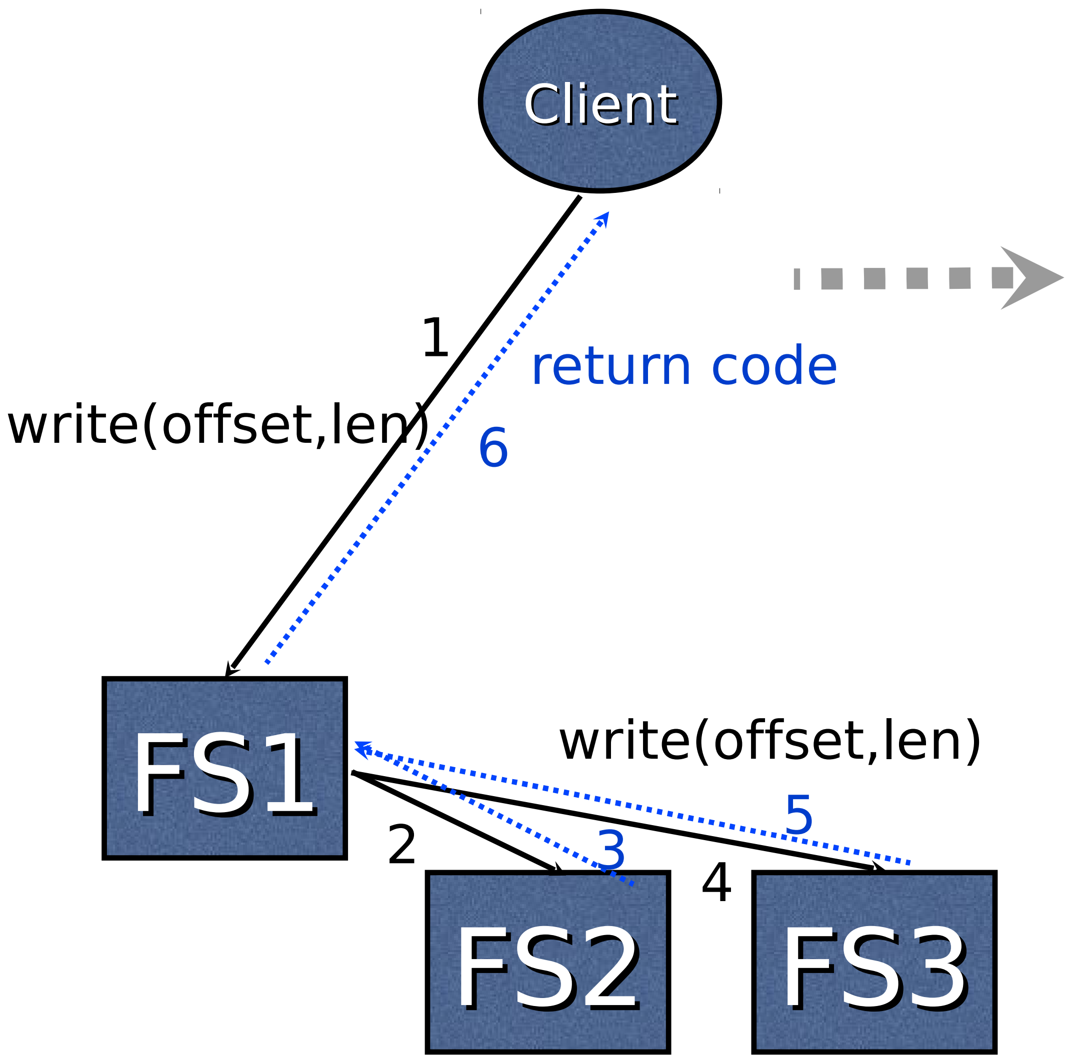



```
EOS Console [root://localhost] |/> ns stat
# ------------------------------------------------------------------------------------
# Namespace Statistic
# ------------------------------------------------------------------------------------
ALL      Files                        10079235
ALL      Directories                  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
```

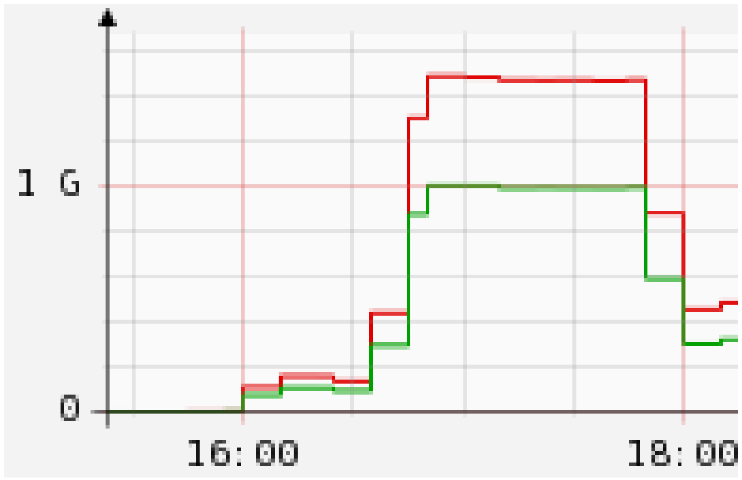
7 kHz

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:

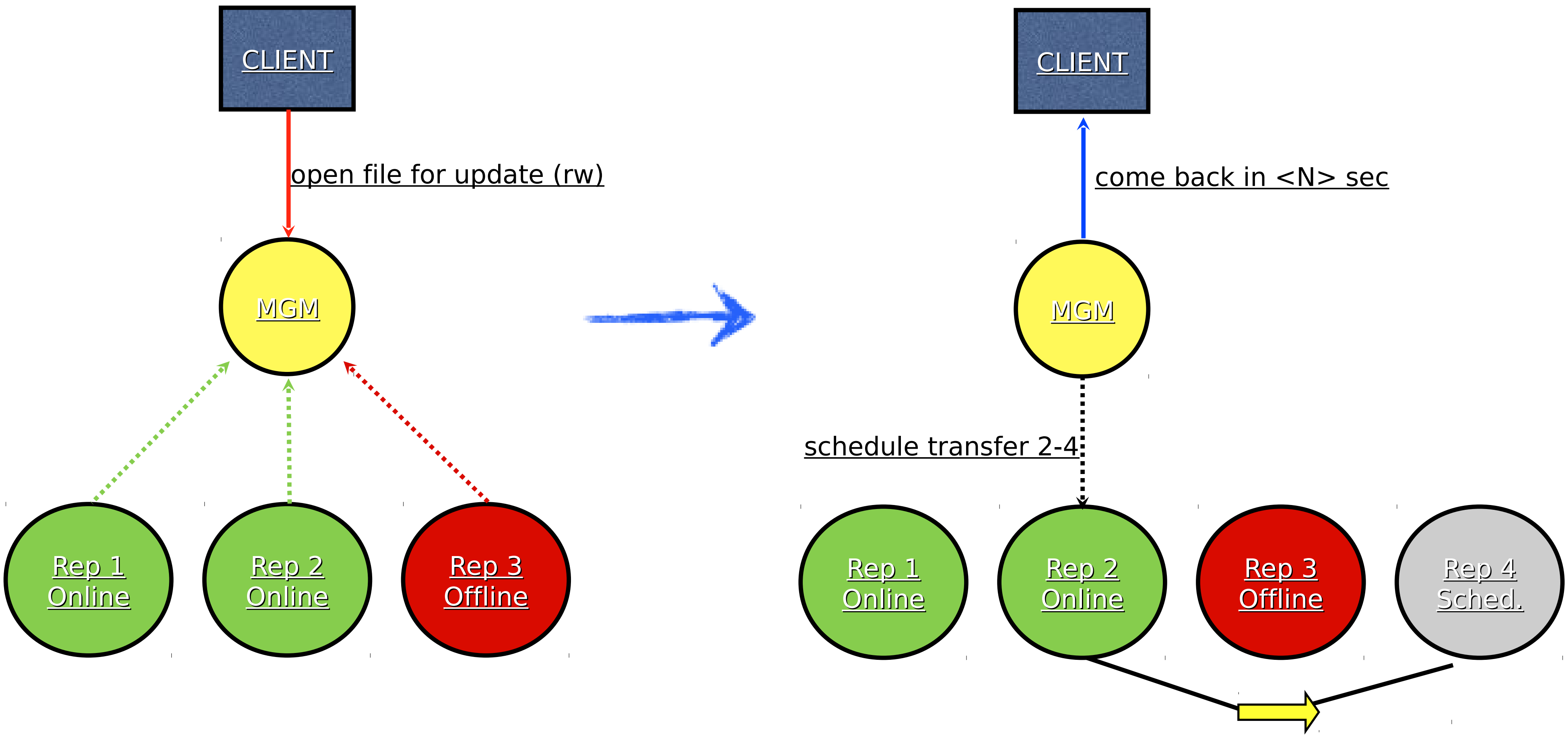


500 MB/s injection result in

- 1 GB/s output on eth0 of all disk servers
- 1.5 GB/s input on eth0 of all disk servers

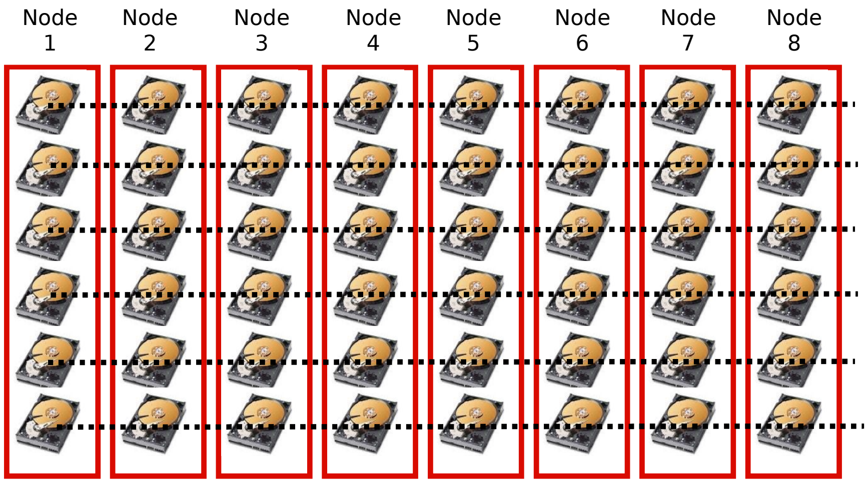
Plain (no replica)
Replica (here 3 replicas)
More sophisticated redundant storage (RAID5, LDPC)

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...)