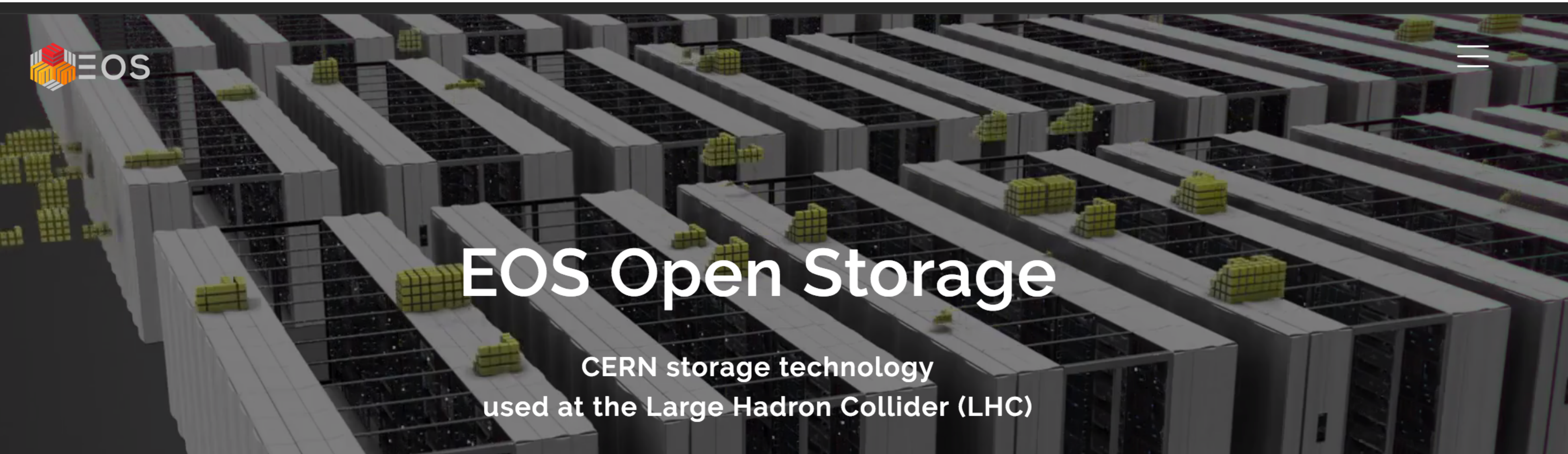
eosxd

# A new FUSE based file system client for EOS

Andreas-Joachim Peters
for the EOS team



EOS Open Storage

CERN storage technology
used at the Large Hadron Collider (LHC)

# Contents

- Introduction
- Architecture
- Implementation
- Performance
- Known issues
- Outlook

# Contents

- Introduction
- Architecture
- Implementation
- Performance
- Known issues
- Outlook

parts of this presentation
are going to be very technical!

Introduction

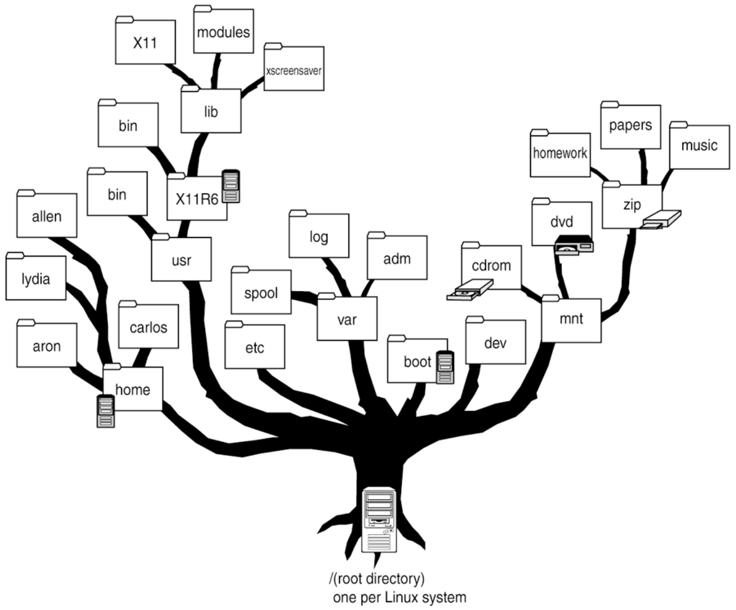
# Background

- Background to `/eos`
  - a filesystem mount is standard API supported by every application - not always the most efficient for physics analysis
  - a filesystem mount is very delicate interface
    - any failure translates into applications failures, job inefficiencies etc.
  - FUSE is a simple (not always) but not the most efficient way to implement a filesystem
  - implementing a filesystem in general is challenging
  - this is the 3rd generation of a FUSE based client for EOS

## Introduction

# Files vs Inodes

- EOS native implementation (XRootD) is
  - access by URL towards meta data server
  - access by inode towards storage server



- Filesystems are implemented as trees starting at a root node 1 with name '/'. Each leave node is identified by a pair of `[parent node, name]`.
  - access is via i-nodes, not by path
  - fundamental for atomic rename operations

eosxd

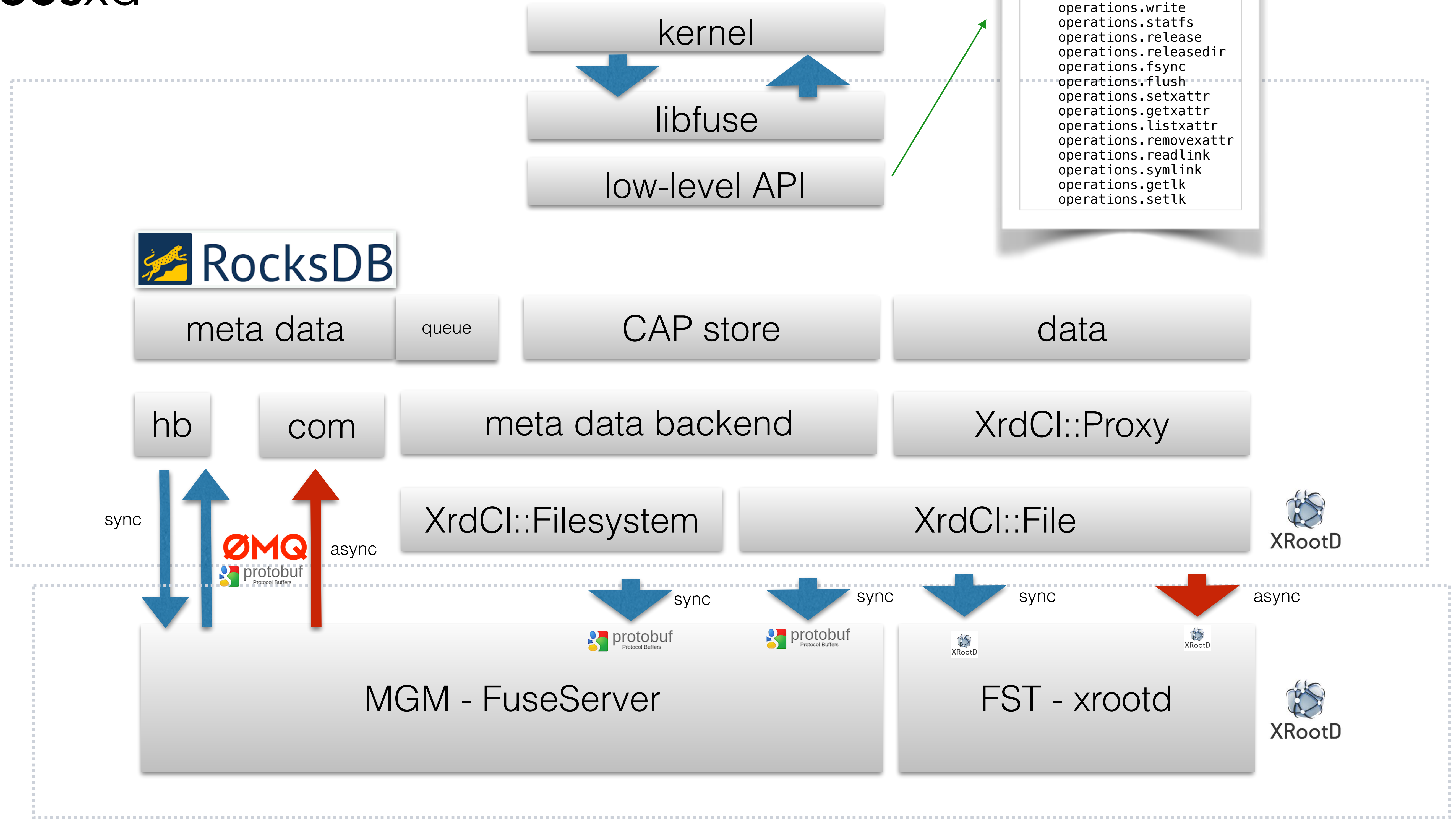
# Introduction
# Goal of reimplementation

- improve posix-ness and atomicity
  - implement file locking (byte-range) - e.g. sqlite needs that
  - implement synchronous & asynchronous IO (O_SYNC)
  - similar posix-ness as AFS - absorb some traditional use cases
  - possibility for NFS4, CIFS, S3 exports
- add more client side caching for performance but implement cross-client consistency
  - manage caches via **CAP** token
    (cache authority provider)
  - strong security mechanism e.g. don’t provide only a trust client model

Introduction

# Architecture

```
operations.getattr
operations.lookup
operations.setattr
operations.opendir
operations.access
operations.readdir
operations.mkdir
operations.unlink
operations.rmdir
operations.rename
operations.open
operations.create
operations.mknod
operations.read
operations.write
operations.statfs
operations.release
operations.releasedir
operations.fsync
operations.flush
operations.setxattr
operations.getxattr
operations.listxattr
operations.removexattr
operations.readlink
operations.symlink
operations.getlk
operations.setlk
```


kernel

libfuse

low-level API

RocksDB

meta data

queue

CAP store

data

hb

com

meta data backend

XrdCl::Proxy

sync

ØMQ

protobuf

async

XrdCl::Filesystem

XrdCl::File

XRootD

sync

sync

sync

async

protobuf

protobuf

MGM - FuseServer

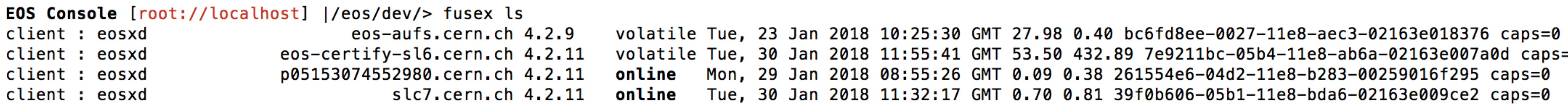
FST - xrootd

XRootD

## Architecture

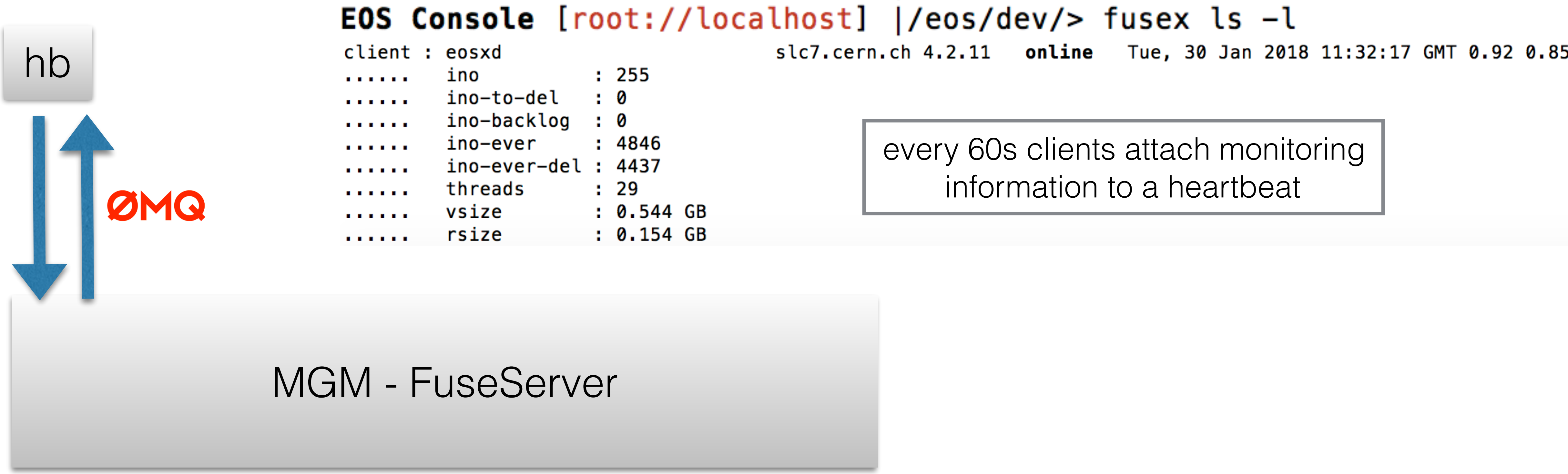
# Client Heartbeat

**eos**xd sends regular heartbeat messages. The default interval is 1s. eosxd receives the heart beat interval with the first handshake. If many clients are used the interval can be reconfigured using ‘eos fusex hb {1..15}’.



```
EOS Console [root://localhost] |/eos/dev/> fusex ls
client : eosxd               eos-aufs.cern.ch 4.2.9    volatile Tue, 23 Jan 2018 10:25:30 GMT 27.98 0.40 bc6fd8ee-0027-11e8-aec3-02163e018376 caps=0
client : eosxd            eos-certify-sl6.cern.ch 4.2.11   volatile Tue, 30 Jan 2018 11:55:41 GMT 53.50 432.89 7e9211bc-05b4-11e8-ab6a-02163e007a0d caps=
client : eosxd            p05153074552980.cern.ch 4.2.11   online   Mon, 29 Jan 2018 08:55:26 GMT 0.09 0.38 261554e6-04d2-11e8-b283-00259016f295 caps=0
client : eosxd                    slc7.cern.ch 4.2.11   online   Tue, 30 Jan 2018 11:32:17 GMT 0.70 0.81 39f0b606-05b1-11e8-bda6-02163e009ce2 caps=0
```

client name host name version status startup time last hb [s] hb flight time[ms] client uuid #caps



```
EOS Console [root://localhost] |/eos/dev/> fusex ls -l
client : eosxd                    slc7.cern.ch 4.2.11   online   Tue, 30 Jan 2018 11:32:17 GMT 0.92 0.85
......   ino          : 255
......   ino-to-del   : 0
......   ino-backlog  : 0
......   ino-ever     : 4846
......   ino-ever-del : 4437
......   threads      : 29
......   vsize        : 0.544 GB
......   rsize        : 0.154 GB
```

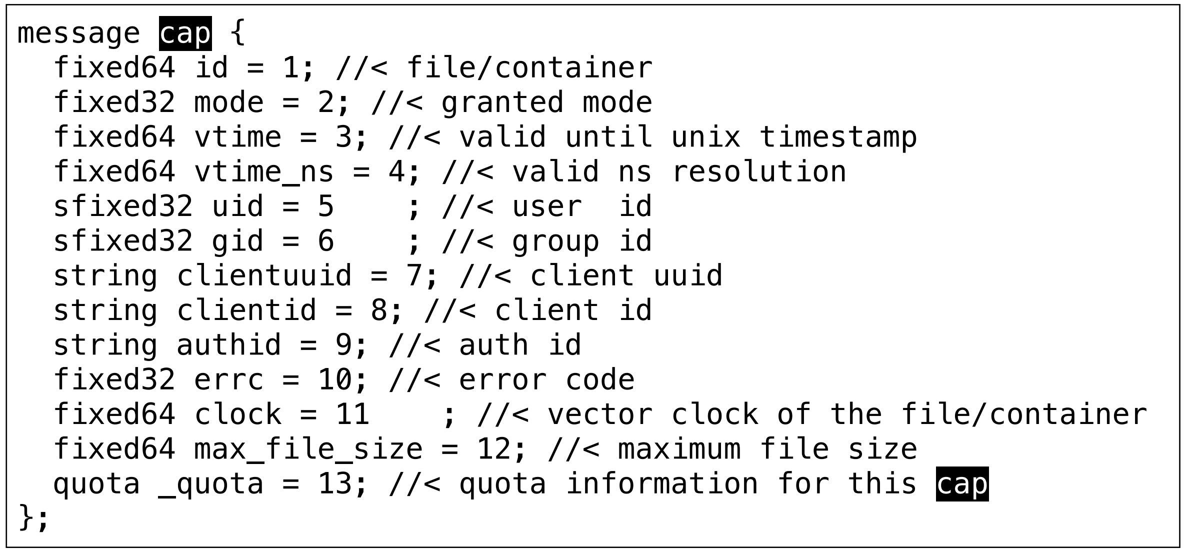
every 60s clients attach monitoring information to a heartbeat

**HINT**: if your local clock drift + network latency > 2s, **eos**xd will fail requests and return EL2NSYNC

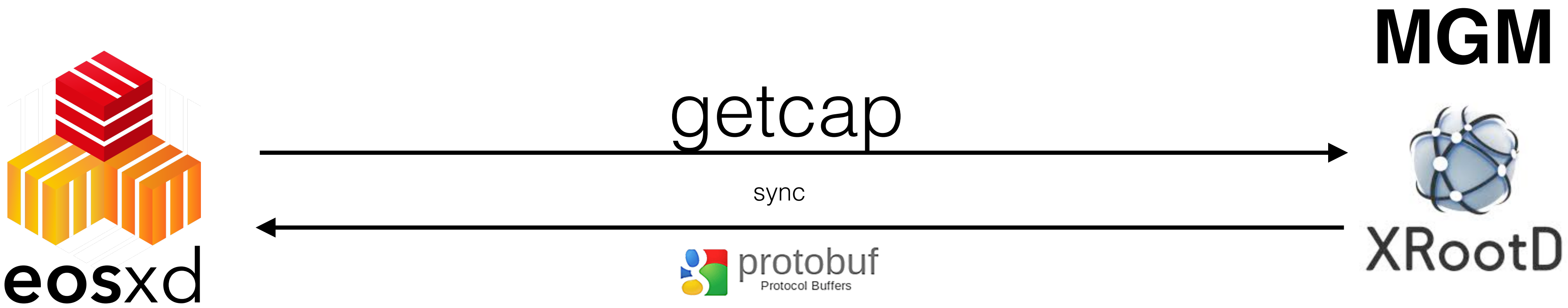
Cache consistency

# **C**ache **A**uthority **P**rovider Tokens

Every FUSE clients retrieves first a **CAP** token for a given directory inode.
This object gets stored on the MGM and **eosxd** client and has a default lifetime of 300s. The token is used to identify whom to call if a directory inode changes meta data or listing information.
A CAP token has embedded several additional policies e.g. client **identity**, client **permissions**, maximum **file size policy** and **quota** information.



```
message cap {
  fixed64 id = 1; //< file/container
  fixed32 mode = 2; //< granted mode
  fixed64 vtime = 3; //< valid until unix timestamp
  fixed64 vtime_ns = 4; //< valid ns resolution
  sfixed32 uid = 5    ; //< user  id
  sfixed32 gid = 6    ; //< group id
  string clientuuid = 7; //< client uuid
  string clientid = 8; //< client id
  string authid = 9; //< auth id
  fixed32 errc = 10; //< error code
  fixed64 clock = 11    ; //< vector clock of the file/container
  fixed64 max_file_size = 12; //< maximum file size
  quota _quota = 13; //< quota information for this cap
};
```

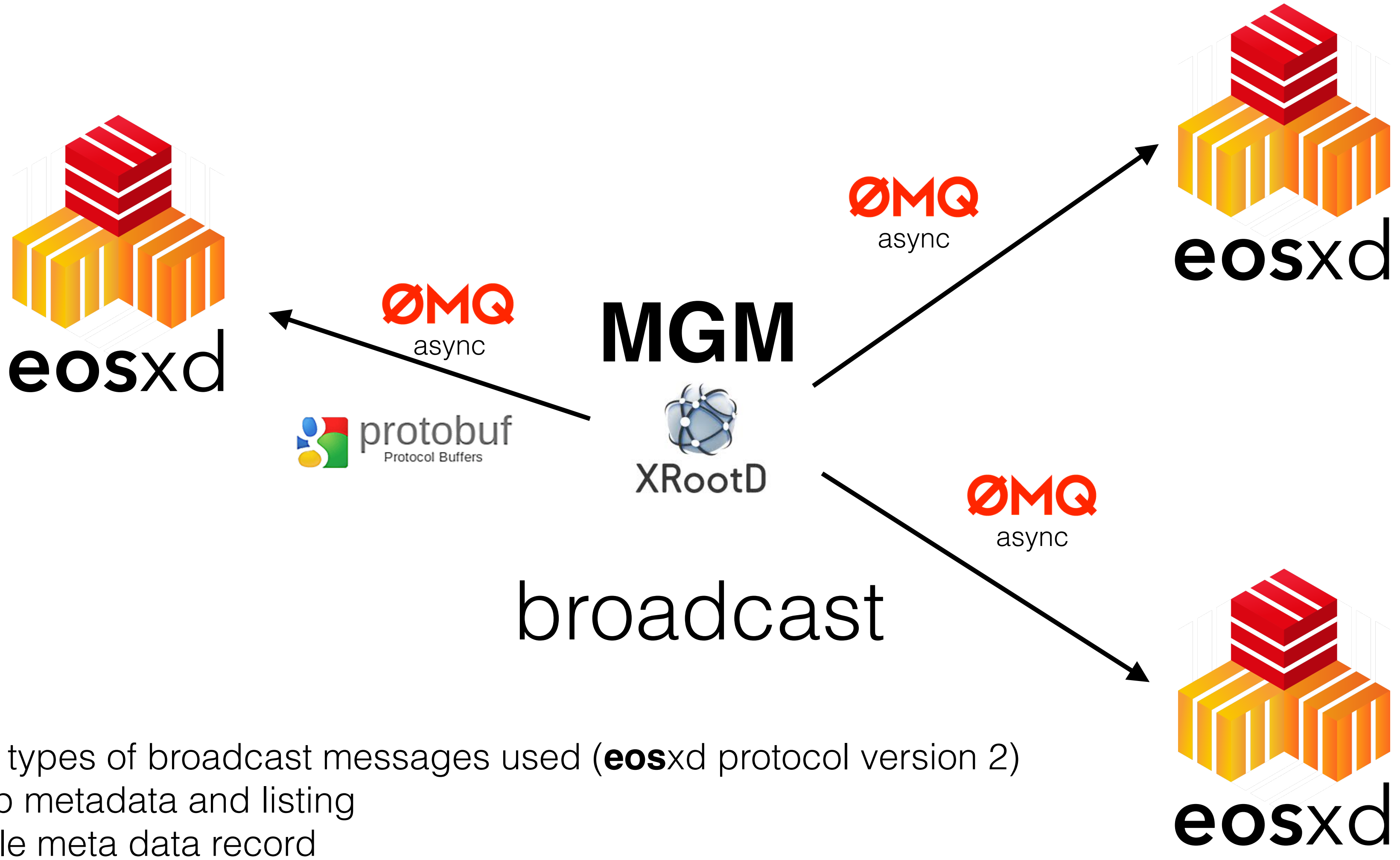


Cache consistency

# **C**ache **A**uthority **P**rovider Callbacks

Whenever meta data information changes on the MGM two callbacks can be invoked

- **internal broadcast** (triggered by one eosxd client, broadcasts to all concerned clients - not himself)
- **external broadcast** (triggered by external clients like xrdcp, cernbox - broadcasts to all concerned clients)



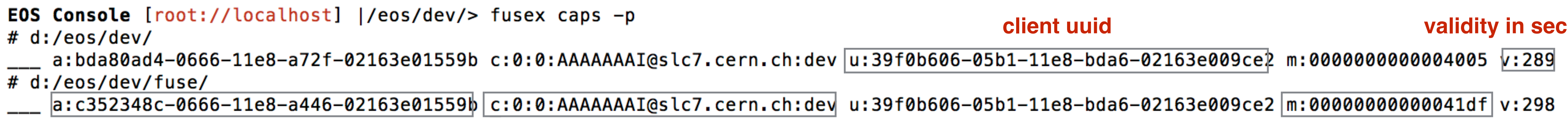
currently two types of broadcast messages used (**eos**xd protocol version 2)

- lease: drop metadata and listing
- md: new file meta data record

## Cache consistency

# MGM Console Interface

Who currently uses which directory?



```
EOS Console [root://localhost] |/eos/dev/> fusex caps -p
# d:/eos/dev/
___ a:bda80ad4-0666-11e8-a72f-02163e01559b c:0:0:AAAAAAAI@slc7.cern.ch:dev u:39f0b606-05b1-11e8-bda6-02163e009ce2 m:0000000000004005 v:289
# d:/eos/dev/fuse/
___ a:c352348c-0666-11e8-a446-02163e01559b c:0:0:AAAAAAAI@slc7.cern.ch:dev u:39f0b606-05b1-11e8-bda6-02163e009ce2 m:00000000000041df v:298
```

client uuid

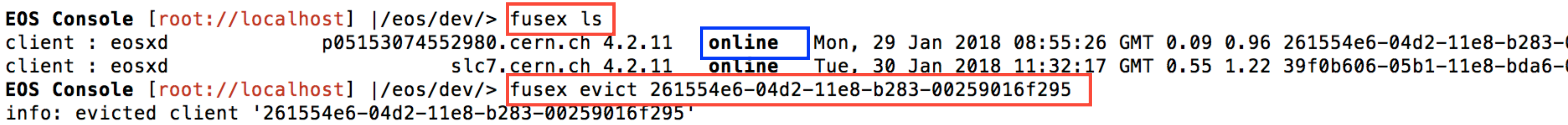
validity in seconds

authentication uuid

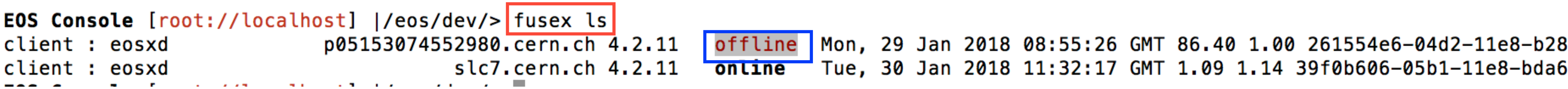
client connection id

granted permissions

Evict a client (leads to a forced umount on client side)



```
EOS Console [root://localhost] |/eos/dev/> fusex ls
client : eosxd              p05153074552980.cern.ch 4.2.11    online   Mon, 29 Jan 2018 08:55:26 GMT 0.09 0.96 261554e6-04d2-11e8-b283-
client : eosxd                          slc7.cern.ch 4.2.11    online   Tue, 30 Jan 2018 11:32:17 GMT 0.55 1.22 39f0b606-05b1-11e8-bda6-
EOS Console [root://localhost] |/eos/dev/> fusex evict 261554e6-04d2-11e8-b283-00259016f295
info: evicted client '261554e6-04d2-11e8-b283-00259016f295'
```

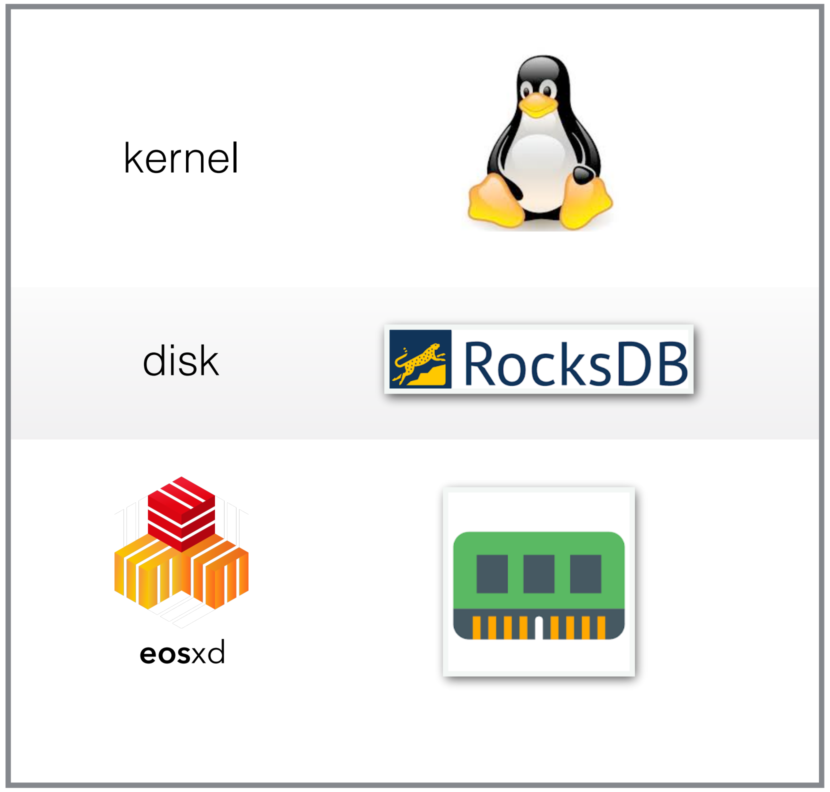


```
EOS Console [root://localhost] |/eos/dev/> fusex ls
client : eosxd              p05153074552980.cern.ch 4.2.11    offline  Mon, 29 Jan 2018 08:55:26 GMT 86.40 1.00 261554e6-04d2-11e8-b28
client : eosxd                          slc7.cern.ch 4.2.11    online   Tue, 30 Jan 2018 11:32:17 GMT 1.09 1.14 39f0b606-05b1-11e8-bda6
```

eosxd

## Client side caching

# Meta Data Caching



**kernel cache**
entry cache (listing)
attribute cache (stat)
[ default lifetime 180s + 5s neg. cache ]

optional

**on-disk** persistent KV store
entry cache (listing)
attribute cache (stat)
v-node table (inode translation local/remote)

optional

**in-memory** meta-data map
[ managed via callbacks and FUSE ]

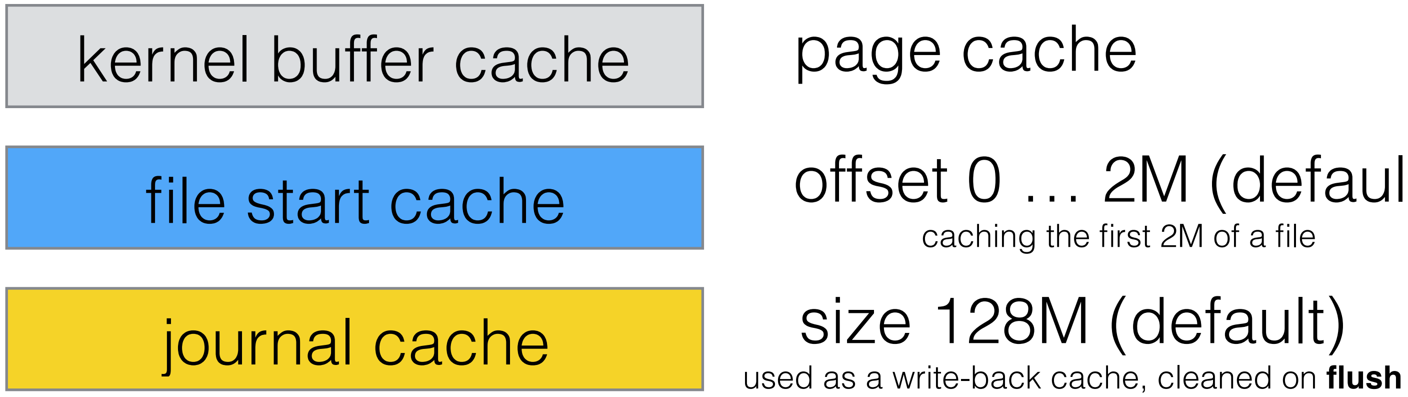
```
message md {
  enum OP { GET = 0; SET = 1; DELETE = 2; GETCAP = 3; LS = 4; GETLK = 5; SETLK = 6; SETLKW = 7; BEGINFLUSH = 8; ENDFLUSH = 9;}
  enum TYPE { MD = 0; MDLS = 1; EXCL = 2;}

  fixed64 id = 1;        //< file/container id
  fixed64 pid = 2;       //< parent id
  fixed64 ctime = 3    ; //< change time
  fixed64 ctime_ns = 4 ; //< ns of creation time
  fixed64 mtime = 5    ; //< modification time | deletion time
  fixed64 mtime_ns = 6 ; //< ns of modification time
  fixed64 atime = 7    ; //< access time
  fixed64 atime_ns = 8 ; //< ns of access time
  fixed64 btime = 9    ; //< birth time
  fixed64 btime_ns = 10; //< ns of birth time
  fixed64 ttime = 11   ; //< tree modification time
  fixed64 ttime_ns = 12; //< ns of tree modification time
  fixed64 pmtime = 13   ; //< tree modification time
  fixed64 pmtime_ns = 14; //< ns of tree modification time
```

## Client side caching

# Data Caching

**3 cache layers**



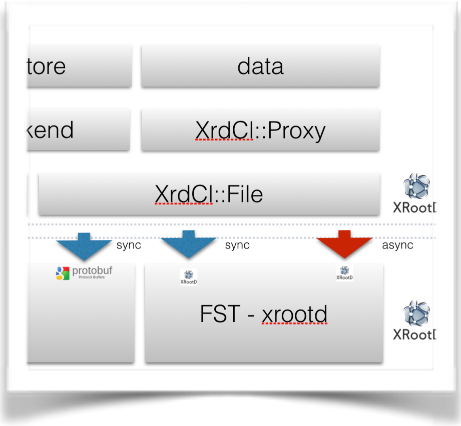
kernel buffer cache — page cache

file start cache — offset 0 … 2M (default)
caching the first 2M of a file

journal cache — size 128M (default)
used as a write-back cache, cleaned on **flush**

- **kernel cache** is invalidated via **CAP** callbacks if files get modified
- **file start cache** is invalidated via **CAP** callbacks
  (file cookie defined by inode,mtime,size)
  - volume based cache cleaning policy
  - by default under /var/eos/fusex/cache/
- **journal cache** is used to persist write operations in flight and to aggregate small sequential writes - journal is truncated with each successful FUSE flush call

# Remote IO
# Data IO



**XrdCl::Proxy** is an extension of the standard XRootD **XrdCl::File** class providing read-ahead and a high-level asynchronous API methods.

**read-ahead strategies**

- none
- **static** read-ahead window
- **dynamic** read-ahead window (window is increased with every new prefetch until max-size)

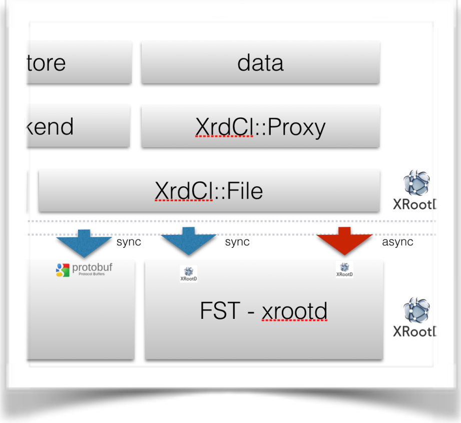
**read-ahead** is disabled if a read falls outside the read-ahead window.

By default **eos**xd uses a dynamic window starting at 1M scaling to 8M. The read-ahea window has an impact on the memory footprint.

The **XrdCl::Proxy** class measures the read-ahead efficiency.

# Remote IO

# Data IO



**XrdCl::Proxy** introduces a latency-free asynchronous API, which is not provided by XrdCl::File e.g. you cannot issue a write before an open has finished a.s.o

=> reason: an *open* can redirect to another machine, but writes should be send only to the final target. You dont' want to send writes to your meta data server!

- **OpenAsync**
- **ScheduleWriteAsync**
- **ScheduleWriteAsync**
- **ScheduleWriteAsync**
- **CloseAsync**

The corresponding barrier functions are:

- **WaitOpen**
- **WaitWrite**
- **WaitClose**

And state functions:

- **IsOpen**
- **HadFailures**
- **IsClosed**
- …

## half-asynchronous write case

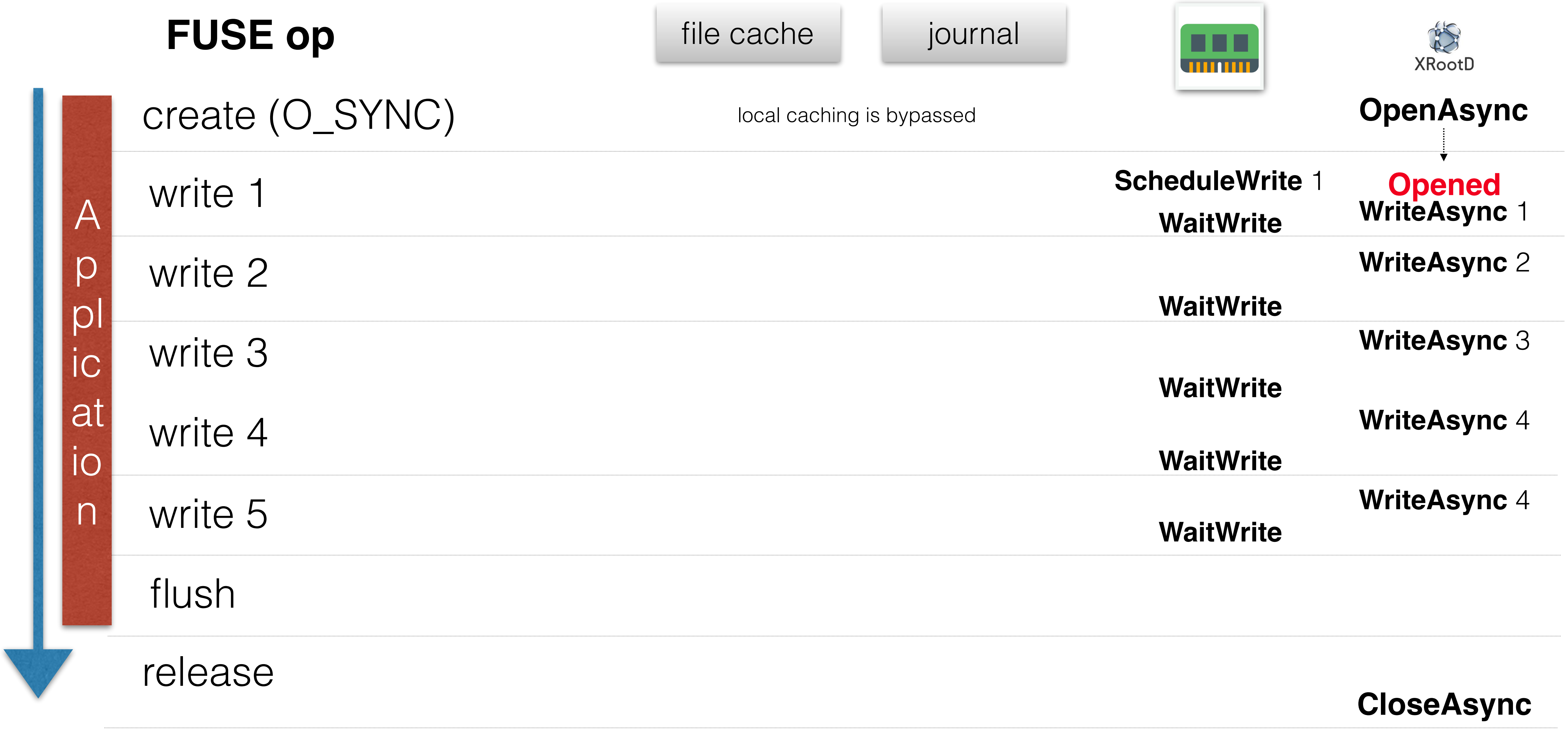
# Data IO

Application

| **FUSE op** | file cache | journal | | XRootD |
|---|---|---|---|---|
| create | create ino.dc | create ino.jc | | **OpenAsync** |
| write 1 | write1  ino.dc | write1 ino.jc | **ScheduleWrite** 1 | **Opened** |
| write 2 | write 2 ino.dc<br>**fc full** | write 2 ino.jc | | **WriteAsync 1**<br>**WriteAsync 2** |
| write 3 | | write 3 ino.jc<br>**jc full** | **WaitWrite** | **WriteAsync** 3 |
| write 4 | | write 4 ino.jc | | **WriteAsync** 4 |
| write 5 | | write 5 ino.jc | | **WriteAsync** 4 |
| flush | | | **HadFailures** | |
| release | | | | **WaitWrite**<br>**CloseAsync** |

# Synchronous write case

# Data IO

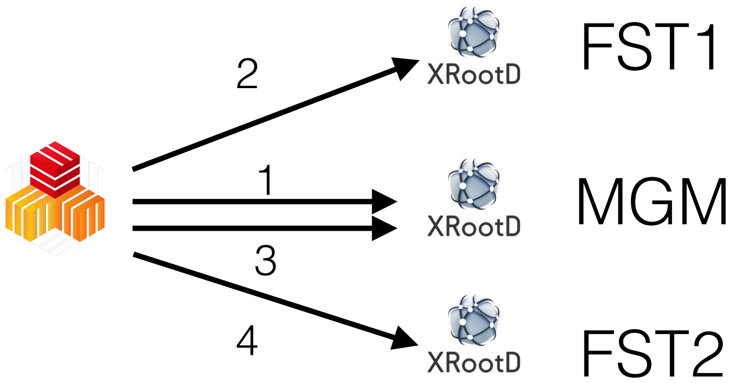


Application

| FUSE op | file cache | journal | | XRootD |
|---|---|---|---|---|
| create (O_SYNC) | local caching is bypassed | | | OpenAsync |
| write 1 | | | ScheduleWrite 1<br>WaitWrite | Opened<br>WriteAsync 1 |
| write 2 | | | WaitWrite | WriteAsync 2 |
| write 3 | | | WaitWrite | WriteAsync 3 |
| write 4 | | | WaitWrite | WriteAsync 4 |
| write 5 | | | WaitWrite | WriteAsync 4 |
| flush | | | | |
| release | | | | CloseAsync |

# Read Recovery
# Data IO

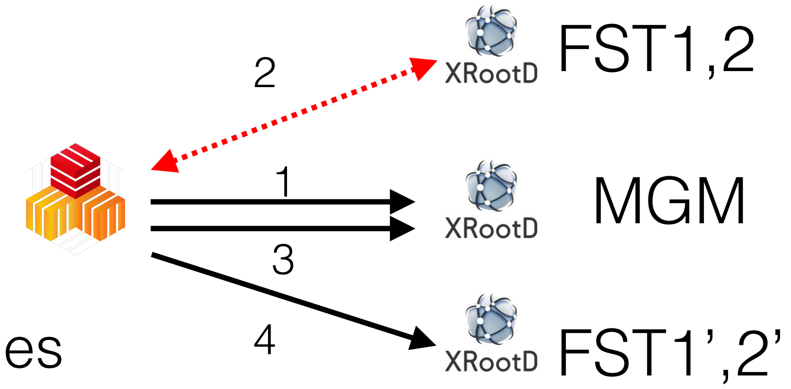
- recovery is implemented inside **eos**xd (not using XrdCl::File recovery)
- falls back to all available replicas/servers
  - uses “?tried=<machine” CGI
- retry period in case all servers are offline can be configured - default 1day



## Write Recovery

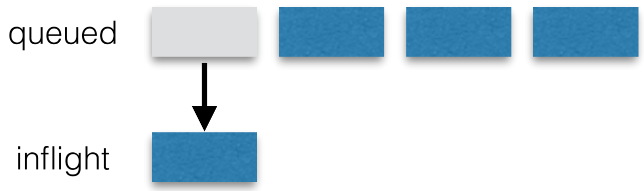
# Data IO

- recovery is steered from the client
- in case of write failures two recovery scenarios exist
  - new file was created and all data is still in local caches
    - recreate inode placement and replay local caches
  - some data exists only in FSTs, update still in local caches
    - stage from available location into local file
    - recreate inode placement, upload staged file and replay local caches
    - requires that the file is still readable
  - client uses the ‘global flush’ facility e.g. no client can open a file which is currently being repaired by a client
  - repair window etc. is configurable



# OOM
# Memory Management

- **XrdCl::Proxy** uses two buffer manager (read ahead+ write) to avoid memory explosion
- **XrdCl:Proxy** limits each manager to max. 1Gb of data in flight
- buffers of a default size of **128k** (max kernel write) are recycled in a queue with max. 128 items [ max. idle persistent size 16 MB ]



- e.g. important of the client writes data faster than the outgoing network pipe can absorb
- Normal reads require additionally temporary buffers
  - served by third buffer manage with 128 x 128k idle persistent size
  - if more read buffers are required, they are allocated as needed
- total idle persisted buffers are **3 x 16 MB = 48 MB**

# Feature Summary

Asynchronous & Synchronous IO (real **fsync** & **O_SYNC**)

Metadata caching  (optional stable inodes)

Data caching/journaling

**Kernel** Metadata & Data caching support

IO error **recovery**

**Symlinks**

Byte-range **locking**

optional **extended attributes** ( birth time eos.btime )

rm -rf level protection

FSYNC **filter** (removes O_SYNC flag for certain file types)

**CPU** core **affinity** + high scheduler **priority** (-10)

autofs support

**shared** & **non-shared** mount (by root, by user)

support **kerberos/X509** authentication, trusted unix

memory **buffer tracking** & **recycling**

client side memory & latency **monitoring**

EOS **ACLs**

**non (yet) supported:**

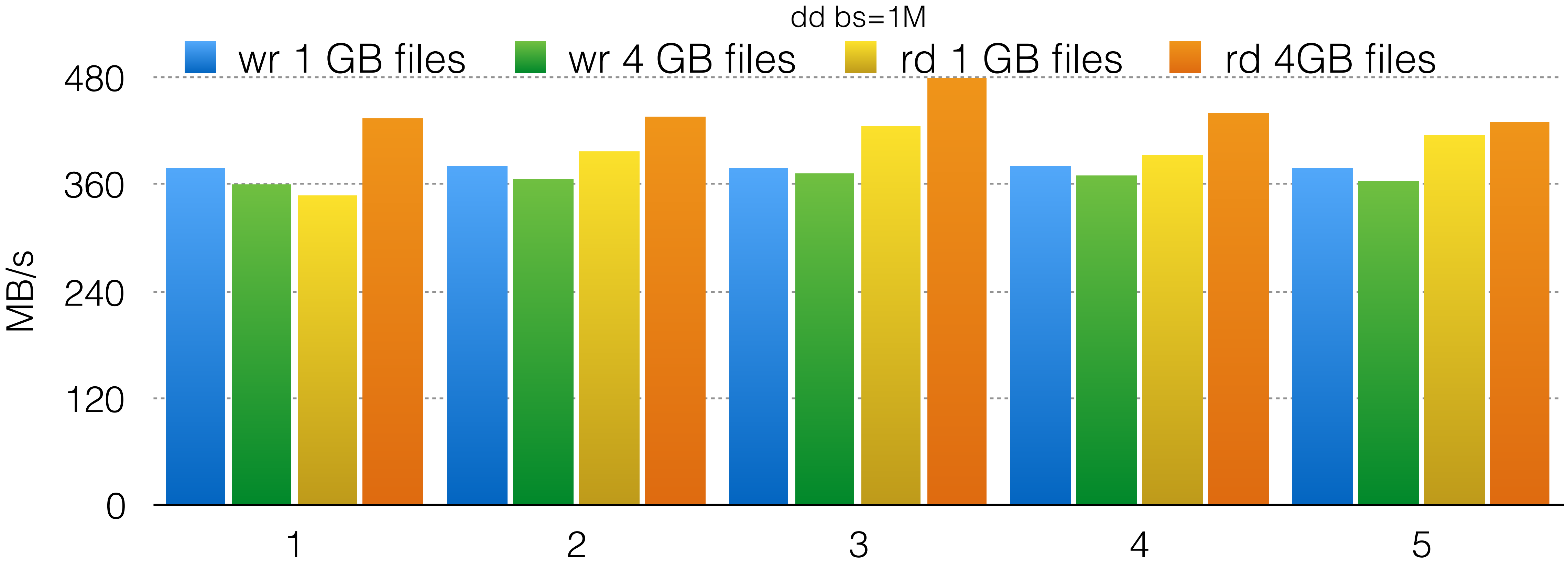
POSIX ACLs (easy with kernel 4.9++)

hardlinks

Measurements

# Performance Metrics

```
1000x mkdir = 870/s
1000x rmdir = 2800/s
1000x touch = 310/s
untar (1000 dirs) = 1.8s
untar (1000 files) = 2.8s
```



# fusex-benchmark

# Performance Metrics

dedicated stress test written to verify race/performance sensitive workloads

```
[root@slc7 ptest]#  /tmp/fusex-benchmark
>>> test 0001
>>> test 0002
>>> test 0003
>>> test 0004
>>> test 0005
>>> test 0006
>>> test 0007
>>> test 0008
>>> test 0009
>>> test 0010
>>> test 0011
>>> test 0012
>>> test 0013

 #0001 : Test::create-delete-loop   2949.578 ms
 #0002 : Test::mkdir-flat-loop      97.297 ms
 #0003 : Test::rmdir-flat-loop      7.999 ms
 #0004 : Test::create-pwrite-loop   302.913 ms
 #0005 : Test::delete-loop          17.351 ms
 #0006 : Test::mkdir-p-loop         387.817 ms
 #0007 : Test::echo-append-loop     7267.009 ms
 #0008 : Test::rename-circular-loop 1606.942 ms
 #0009 : Test::truncate-expand-loop 518.711 ms
 #0010 : Test::journal-cache-timing 627.372 ms
 #0011 : Test::dd-diff-16k-loop     1069.299 ms
 #0012 : Test::dd-diff-16M-loop     3328.005 ms
 #0013 : Test::write-unlinked-loop  6654.631 ms
```

# **eos**xd statics for fusex-benchmark

```
                          [root@xxx tree1]# cat /var/log/eos/fusex/fuse.dev.stats
                          ALL        Execution Time                    1.66 +- 12.30
# ----------------------------------------------------------------------------------------------------------
who      command                          sum             5s      1min      5min       1h  exec(ms)  +-  sigma(ms)
# ----------------------------------------------------------------------------------------------------------
ALL        :sum                        1370516      0.00      0.02    548.47    380.80     -NA- +- -NA-
ALL        access                           62      0.00      0.00      0.20      0.02  0.06882 +- 0.01530
ALL        create                          295      0.00      0.00      0.98      0.08  1.73413 +- 2.48593
ALL        flush                         40300      0.00      0.00     67.87     11.20  0.08496 +- 0.02907
ALL        forget                          458      0.00      0.00      1.53      0.13  0.00076 +- 0.00123
ALL        fsync                            22      0.00      0.00      0.07      0.01 53.08414 +- 38.99056
ALL        getattr                         653      0.00      0.02      1.84      0.18  0.17546 +- 0.07830
ALL        getxattr                          0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        listxattr                         0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        lookup                          481      0.00      0.00      1.59      0.13  0.00562 +- 0.00306
ALL        mkdir                           163      0.00      0.00      0.55      0.05  1.93330 +- 5.61503
ALL        mknod                             0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        open                          20001      0.00      0.00     33.44      5.56  0.13977 +- 0.02817
ALL        opendir                         228      0.00      0.00      0.76      0.06  0.42964 +- 0.68154
ALL        read                            110      0.00      0.00      0.37      0.03  9.50537 +- 36.90263
ALL        readdir                         186      0.00      0.00      0.62      0.05  0.00666 +- 0.00375
ALL        readlink                          0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        release                       20296      0.00      0.00     34.42      5.64  0.00975 +- 0.00455
ALL        releasedir                      228      0.00      0.00      0.76      0.06  0.00243 +- 0.00143
ALL        removexattr                       0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        rename                            0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        rm                                0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        rmdir                           163      0.00      0.00      0.55      0.05  0.05919 +- 0.01316
ALL        setattr                           3      0.00      0.00      0.00      0.00  3.37133 +- 0.25233
ALL        setattr:chmod                     0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        setattr:chown                     0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        setattr:truncate                  3      0.00      0.00      0.00      0.00  3.18167 +- 0.23384
ALL        setattr:utimes                    0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        setxattr                          0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        statfs                            3      0.00      0.00      0.01      0.00  0.38000 +- 0.45679
ALL        symlink                           0      0.00      0.00      0.00      0.00     -NA- +- -NA-
ALL        unlink                          294      0.00      0.00      0.98      0.08  0.38519 +- 1.11495
ALL        write                       1286567      0.00      0.00    401.94    357.48  0.03366 +- 0.00880
# ----------------------------------------------------------------------------------------------------------
```

1.3 M FuseOPS

on EL7 machine: max **20kHz** write IOPS

fsync slowest operation (fsync's FST disk)

# QA
# Code Certification

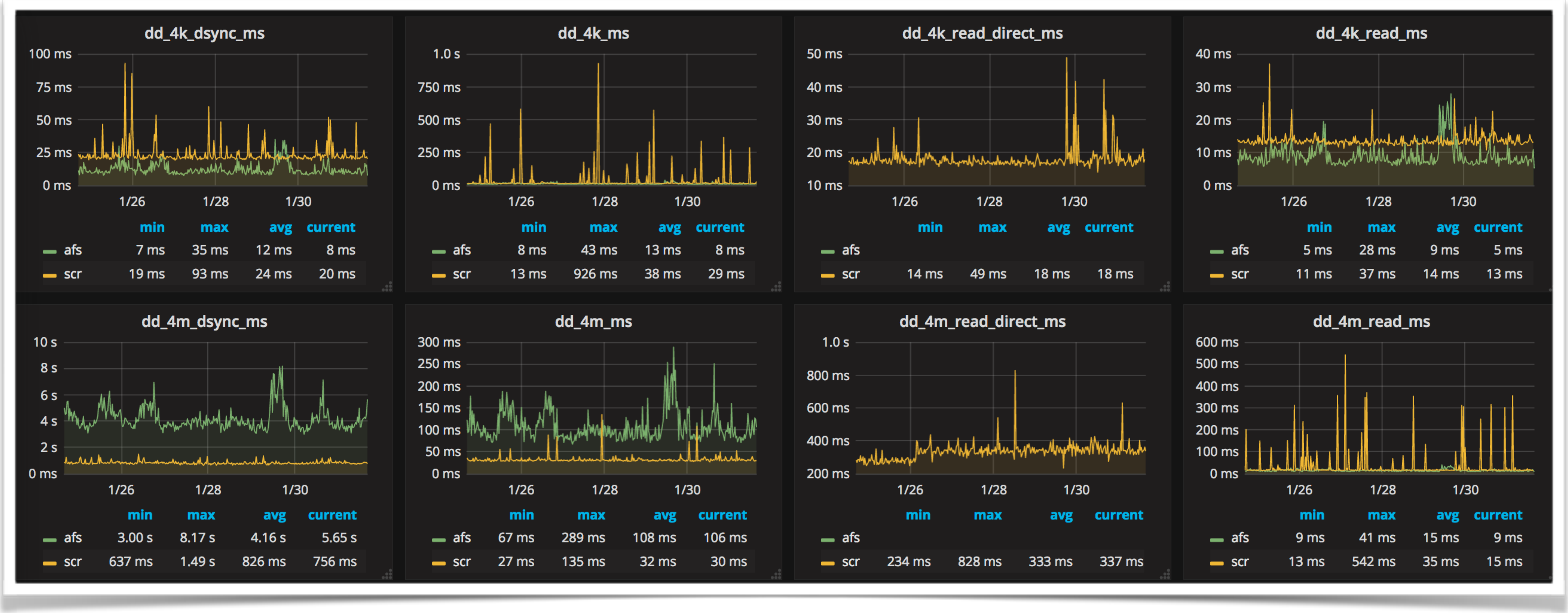
developed dedicated certification script **eos-fusex-certify**

1. build zlib rpm (autotools)
2. git clone
3. rsync trees
4. sqlite tests
5. microtests
6. build eos rpm
7. test write recovery

## Comparison to AFS

# Microtest Performance

production AFS volume compared to EOS instance (1000 disks / 5PB / 50 nodes)



dd_4k_dsync_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 7 ms | 35 ms | 12 ms | 8 ms |
| scr | 19 ms | 93 ms | 24 ms | 20 ms |

dd_4k_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 8 ms | 43 ms | 13 ms | 8 ms |
| scr | 13 ms | 926 ms | 38 ms | 29 ms |

dd_4k_read_direct_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | | | | |
| scr | 14 ms | 49 ms | 18 ms | 18 ms |

dd_4k_read_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 5 ms | 28 ms | 9 ms | 5 ms |
| scr | 11 ms | 37 ms | 14 ms | 13 ms |

dd_4m_dsync_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 3.00 s | 8.17 s | 4.16 s | 5.65 s |
| scr | 637 ms | 1.49 s | 826 ms | 756 ms |

dd_4m_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 67 ms | 289 ms | 108 ms | 106 ms |
| scr | 27 ms | 135 ms | 32 ms | 30 ms |

dd_4m_read_direct_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | | | | |
| scr | 234 ms | 828 ms | 333 ms | 337 ms |

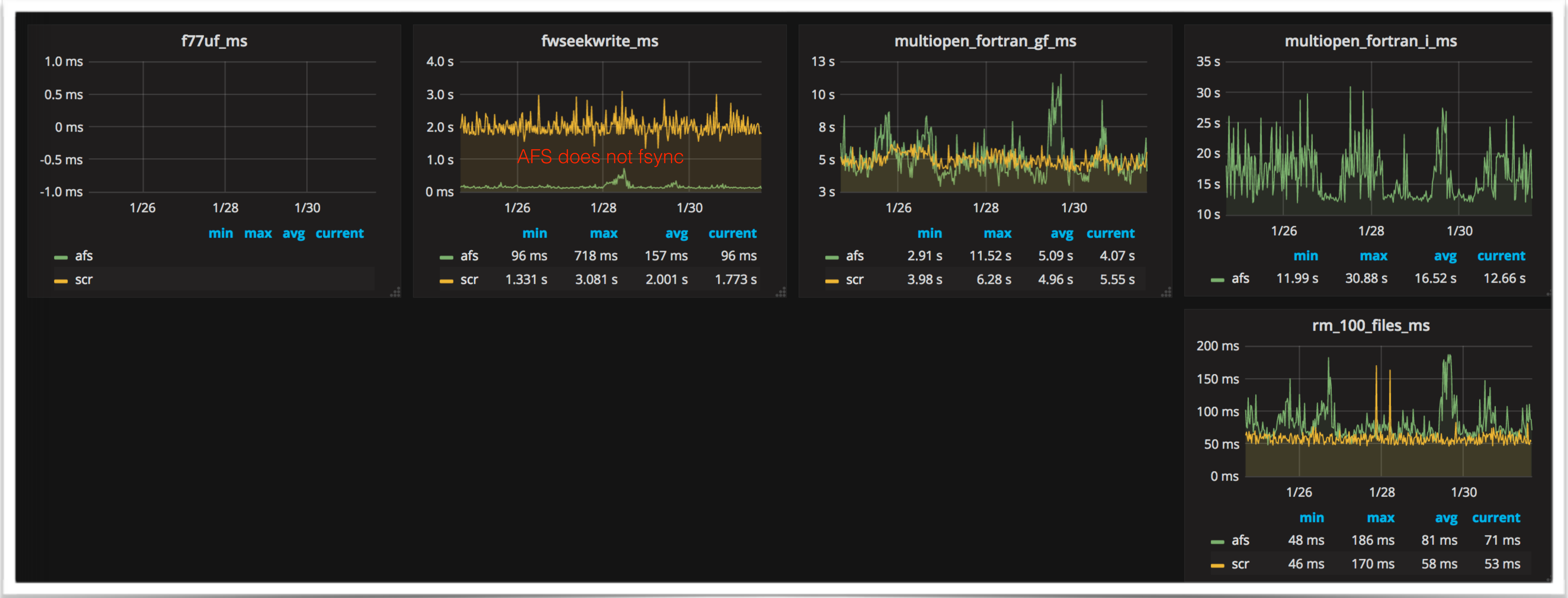
dd_4m_read_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 9 ms | 41 ms | 15 ms | 9 ms |
| scr | 13 ms | 542 ms | 35 ms | 15 ms |

# Comparison to AFS

# Microtest Performance

production AFS volume compared to EOS instance (1000 disks / 5PB / 50 nodes)



AFS does not fsync

f77uf_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | | | | |
| scr | | | | |

fwseekwrite_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 96 ms | 718 ms | 157 ms | 96 ms |
| scr | 1.331 s | 3.081 s | 2.001 s | 1.773 s |

multiopen_fortran_gf_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 2.91 s | 11.52 s | 5.09 s | 4.07 s |
| scr | 3.98 s | 6.28 s | 4.96 s | 5.55 s |

multiopen_fortran_i_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 11.99 s | 30.88 s | 16.52 s | 12.66 s |

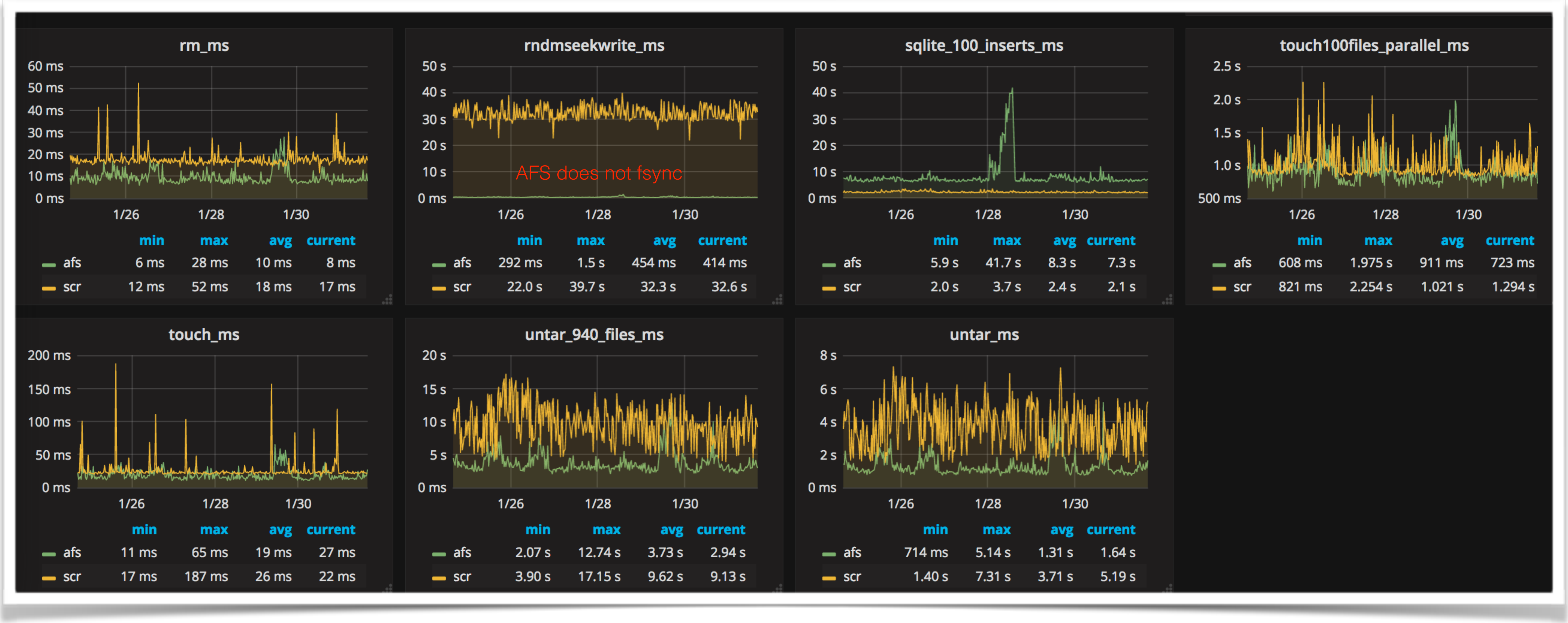
rm_100_files_ms

| | min | max | avg | current |
|---|---|---|---|---|
| afs | 48 ms | 186 ms | 81 ms | 71 ms |
| scr | 46 ms | 170 ms | 58 ms | 53 ms |

# Comparison to AFS

# Microtest Performance

production AFS volume compared to EOS instance (1000 disks / 5PB / 50 nodes)



rm_ms

| | min | max | avg | current |
|---|---|---|---|---|
| 🟩 afs | 6 ms | 28 ms | 10 ms | 8 ms |
| 🟨 scr | 12 ms | 52 ms | 18 ms | 17 ms |

rndmseekwrite_ms

AFS does not fsync

| | min | max | avg | current |
|---|---|---|---|---|
| 🟩 afs | 292 ms | 1.5 s | 454 ms | 414 ms |
| 🟨 scr | 22.0 s | 39.7 s | 32.3 s | 32.6 s |

sqlite_100_inserts_ms

| | min | max | avg | current |
|---|---|---|---|---|
| 🟩 afs | 5.9 s | 41.7 s | 8.3 s | 7.3 s |
| 🟨 scr | 2.0 s | 3.7 s | 2.4 s | 2.1 s |

touch100files_parallel_ms

| | min | max | avg | current |
|---|---|---|---|---|
| 🟩 afs | 608 ms | 1.975 s | 911 ms | 723 ms |
| 🟨 scr | 821 ms | 2.254 s | 1.021 s | 1.294 s |

touch_ms

| | min | max | avg | current |
|---|---|---|---|---|
| 🟩 afs | 11 ms | 65 ms | 19 ms | 27 ms |
| 🟨 scr | 17 ms | 187 ms | 26 ms | 22 ms |

untar_940_files_ms

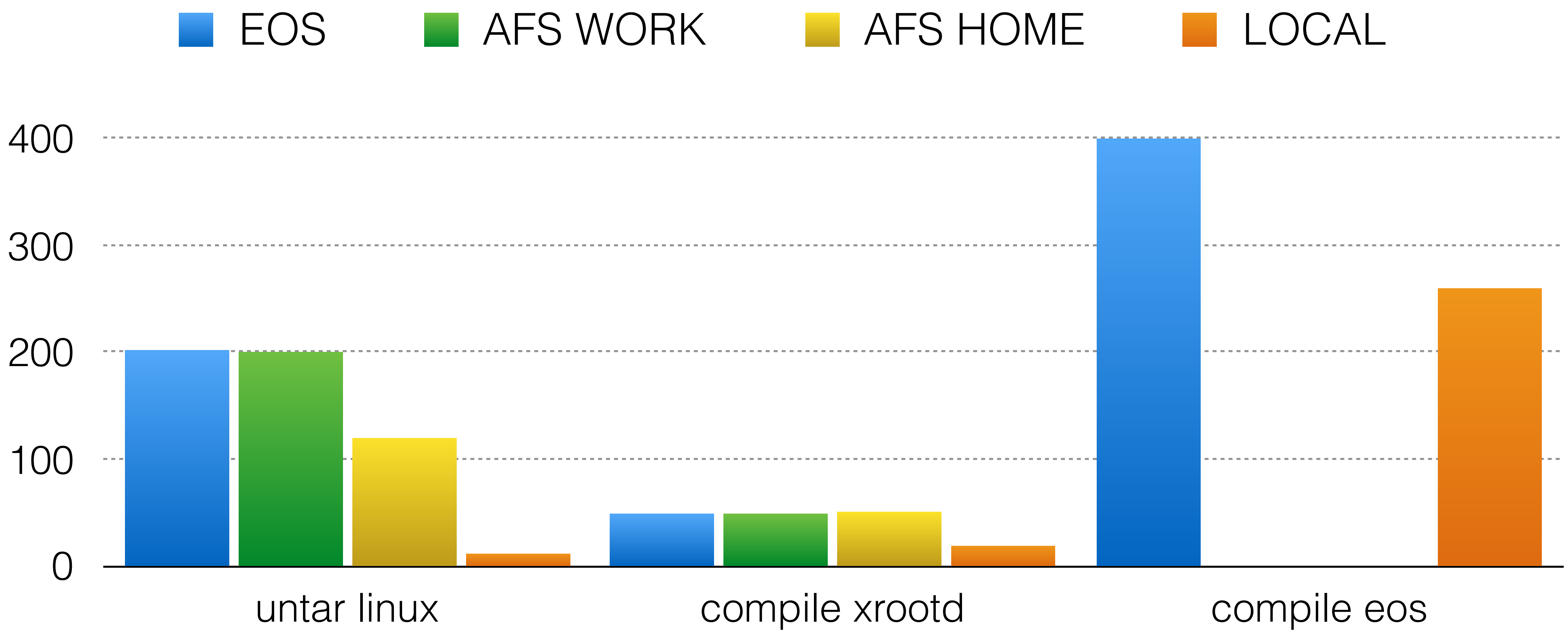
| | min | max | avg | current |
|---|---|---|---|---|
| 🟩 afs | 2.07 s | 12.74 s | 3.73 s | 2.94 s |
| 🟨 scr | 3.90 s | 17.15 s | 9.62 s | 9.13 s |

untar_ms

| | min | max | avg | current |
|---|---|---|---|---|
| 🟩 afs | 714 ms | 5.14 s | 1.31 s | 1.64 s |
| 🟨 scr | 1.40 s | 7.31 s | 3.71 s | 5.19 s |

# Comparison to AFS
# Daily work

- untar linux source (65k files/directories)
- compile xrootd
- compile eos



When comparing, keep in mind: AFS is a kernel implementation, **eos**xd user space

eosxd

# Comparison to AFS
# Known Issues

- race condition/implementation fault: **listing** vs **invalidation callback** vs **kernel cache**
    - leads to invisible or ghost entries in listing (in particular when used via NFS)
    - now fully understood - WIP - fixed in next release
- quota enforcement is (too) lazy e.g. clients can overrun quota within 300s window if files are produced on several machines

# Configuring **eos**xd

# Configuration File

All you need to know is explained here:

https://gitlab.cern.ch/dss/eos/tree/master/fusex

Simplest way to mount with default settings:

```
mount -t fuse eosxd eosuat.cern.ch:/eos/scratch /eos/scratch
```

MGM host MGM path local mountdir

Mounting with configuration files is explained here:

https://gitlab.cern.ch/dss/eos/tree/master/fusex#configuration-default-values-and-avoiding-configuration-files

# Inspecting **eos**xd

# Statistics File

cat /var/log/eos/fusex/fuse[.instance].stats

```
# --------------------------------------------------------------------------------------------
ALL        inodes              := 1         inodes known to the mount
ALL        inodes-todelete     := 0         inodes which still need to be deleted
ALL        inodes-backlog      := 0         inodes which are still to be flushed
ALL        inodes-ever         := 1         inodes ever in use
ALL        inodes-ever-deleted := 0         inodes ever deleted
ALL        inodes-open         := 0         files currently open
ALL        inodes-vmap         := 1         map size local/remote inode translation
ALL        inodes-caps         := 0         map size of CAP token
# --------------------------------------------------------------------------------------------
ALL        threads             := 17        threads in use
ALL        visze               := 336.41 Mb virtual memory size
All        rss                 := 53.10 Mb  physical memory size
All        wr-buf-inflight     := 0 b       size of write buffers in flight
All        wr-buf-queued       := 0 b       size of write buffers recycled
All        ra-buf-inflight     := 0 b       size of read-ahead buffers in flight
All        ra-buf-queued       := 0 b       size of read-ahead buffers recycled
All        rd-buf-inflight     := 0 b       size of read buffers in flight
All        rd-buf-queued       := 0 b       size of read buffers recycled
All        version             := 4.2.11    eosxd version
ALl        fuseversion         := 28        fuse protocol version
All        starttime           := 1517583072 start time unix
All        uptime              := 1         uptime in seconds
All        instance-url        := e.cern.ch MGM host
# --------------------------------------------------------------------------------------------
```

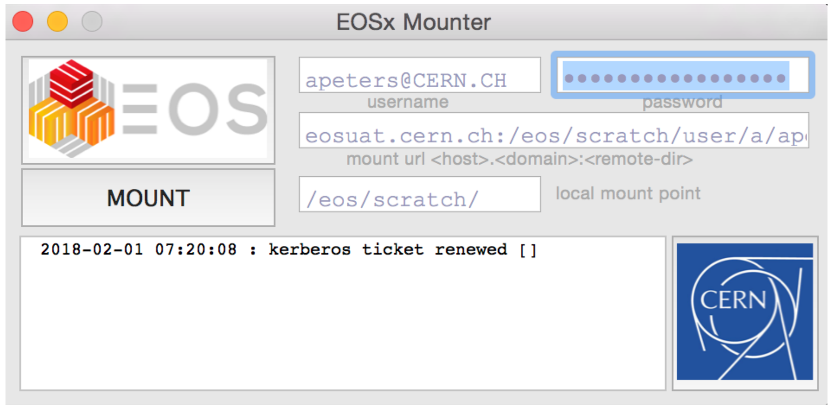
**eos**xd

# Short-term Q1 Roadmap

1. finalise and certify production quality release with protocol version 2
   leases & file update broadcast

2. validate NFS & CIFS gateway configuration

3. validate OS X client



apeters@CERN.CH

•••••••••••••••••

eosuat.cern.ch:/eos/scratch/user/a/ap

/eos/scratch/

```
2018-02-01 07:20:08 : kerberos ticket renewed [ ]
```

4. performance tuning
   - sync->(half-)async MD flush queue (for high latency links)

# Enhancements

# Possible Evolution

- remove ZMQ, use SSI (see CTA talks)
- implement protocol version 3
  - distinguish shared & exclusive lease
    - shared lease: create/mkdir must be **sync** call
    - exclusive lease: create/mkdir can be **async** call
      = 3 x performance boost  (300 files/s => 1000 files/s)
  - use differential broadcasts
    (instead of invalidating cache, broadcast what has changed)
- support multi MGM deployments

# Enhancements

# Possible Evolution

- implement **O**verlay **A**rchive **V**olumes to get rid of the many small files in EOS

  [ think of ZIP archives with COW overlay files ]

- standard behaviour
  - support RICH acls (requires new kernel)
  - support hardlinks
  - evaluate permissions on files
- FUSE3 write-back cache (requires new kernel)

# Summary

- implementing a reliable & performant filesystem client is a complex task
- re-implementation has good performance indicators and much improved posix-ness
  - nevertheless EOS by design is not full posix compliant - similar to AFS
  - FUSE can not compete in some aspects with a kernel FS driver
- implementation is almost feature complete (missing hardlinks)
- we will certainly collect production experience this year
  - target use is with QuarkDB backend - it is a threat to the in-memory namespace

A gentil reminder: if you need a parallel filesystem, use one.
If you need a posix filesystem, use one.
If you can use a local filesystem, use one.
**eos**xd is neither of the latter.

# Acknowledgments

- journal cache & thread pool implementation by **M. Simon**
- strong security and deep dead-lock debugging by **G. Bitzes**
- server-side CITRINE port by **E. Sindrilaru**
- CI integration **J. Makai**
- valuable feedback, packaging, testing and discussion with the **CERN Ops team** and AFS guru **R. Toebbicke**