Migration: from EOSUSER to EOSHOME

Luca Mascetti
CERN IT Storage
EOS Architecture

EOS Production Releases

Aquamarine V 0.3.X
Citrine V 4.X

XRooD V3
IPV4 namespace in-memory data on attached disks

XRooD V4
IPV6 plugins for meta data & data persistency

EOS Architecture Diagram

Client
SAMBA, SRM, gridFTP, WebDAV
S3, FUSE, xrdcp, ROOT, https

Apps
MGM, MQ

Client

Namespace

MD Server

Data Server
EOSUSER a.k.a. CERNBox

- Native clients
- Fuse clients
- Windows clients
- Sync and mobile clients
- Samba gateway
- OwnCloud
- Nginx HTTPS LB
- WebDAV
- Namespace
- Data

web frontend (only) oc_shares
EOS Namespace Challenge

Namespace restart (boot) time

MGM restart:
1. software upgrade
2. memory growth
3. crash or out of memory

While booting:
- Instance is in read-only (serving data from MGM Slave)
EOS Namespace Challenge

Memory consumption

Number of files

- 700 Mil
- 600 Mil
- 500 Mil
- 400 Mil
- 300 Mil
- 200 Mil
- 100 Mil

1TB

namespace

MGM
ATTENZIONE PERICOLO
POSSIBILITÀ DI ONDE DI RENNA IMPROVVISE ANCHE PER MANOVRE SU OPERE IDRAULICHE.

DANGER
POSSIBILITY OF SUDDEN FLOOD WAVES ALSO BECAUSE OF MANOEUVRES ON HYDRAULIC PLANTS

ATTENTION DANGER
POSSIBILITÉ DE CRUES Soudaines À LA SUITE Aussi DE MANŒUVRES SUR OUVRAGES HYDRAULIQUES

ACHTUNG GEFahr
MÖGLICHKEIT PLOCKLICHER FLUTWELLEN AUCH ZUFOLGE VON BETÄTIGUNG DER STAUDAMMSCHÜTZE
EOS Namespace Challenge
Shifting namespace paradigm: from scale-up to scale-out

New Namespace
• KV store resident on disk
• very short restart time!!
  • not based on #files and #dirs
• namespace caching in MGM memory
Now, how to proceed ...
Solution 1: EOSUSER upgrade
Upgrade current production

Two steps upgrade:
1. upgrade from aquamarine to citrine
2. namespace conversion

From past experiences:
• very very very long downtime => just not acceptable
• instabilities in booting filesystems with millions of files
Solution 1: EOSUSER upgrade

Upgrade current production

Two steps upgrade:

1. upgrade from aquamarine to citrine
2. namespace reconcile

From past experiences:

• very very very long downtime => just not acceptable
• instabilities in booting filesystems with millions of files
Solution 2: EOSUSER2

Deploy a new empty instance with latest namespace technology

New deployment and migration:
1. build a new empty EOS instance
   1. start immediately with QDB namespace
2. migrate gradually users

From past experiences:
• migration needs to be transparent!!!
• no BIG-BANG! approach
• better load control over time
• better operations “feeling”
Solution 2: EOSUSER2

Deploy a new empty instance with latest namespace technology
FYI: Behind the scenes

sync clients

fuse

data resync???

fusemount

symlinks on

user redirection

user mapping

new inode

101 continue?

sharing functionalities
sharing migration
etag: inode+xs

http mapping functionalities
propfind xattr functionalities

user data migration

MGM

NS

eosuser

eosuser2

RocksDB

RocksDB

RocksDB

web frontend

oc_DB

auth

blender
eos

go-auth

user mapping

MGM

NS
Solution 2: EOSUSER2

Deploy a new empty instance with latest namespace technology

Some additional considerations:
- single instance for all users
- MGMs single point of failures
- Scale metadata performance
- difficult users isolation
- future big upgrade => big bang?
- "plane crash" effect
- Shared Desktop across devices
  - CERN $HOME future plans
  - DFS and linux home discussions
Solution 2: EOSUSER2

Deploy a new empty instance with latest namespace technology

Some additional considerations:

- single instance for all users
- MGMs single point of failures
- Scale metadata performance
- difficult users isolation
- future big upgrade => big bang?”
- “plane crash” effect
- Shared Desktop across devices
  - CERN $HOME future plans
  - DFS and linux/home discussions
Solution 3: ...
Divide et Impera
Solution 3: EOSHOME (running out of names...)

Deploy multiple empty instances with latest namespace technology

Architectural review, new deployment and migration:
1. build a multiple empty EOS instance
   1. start immediately with QDB namespace
2. add a level of indirection
3. support system expansion and reduction
4. migrate gradually users

From past experiences:
- migrations need to be transparent!!!
- no BIG-BANG! approach
  - gradual (future) software roll-out
- better load control over time
- better operations “feeling”
- better user isolation
- less load/stress per instance
Solution 3: EOSHOME (running out of names...)

Deploy multiple empty instances with latest namespace technology

Architectural review, new deployment and migration:
1. build a multiple empty EOS instance
   1. start immediately with ODB namespace
   2. add a level of indirection
   3. support system expansion and reduction
   4. migrate gradually users

From past experiences:
• migrations need to be transparent!!!
• no BIG BANG! Approach
  • gradual (future) software roll-out
• better load control over time
• better operations “feeling”
• better user isolation
• less load/stress per instance
**Solution 3: EOSHOME**

Deploy *multiple empty instances* with latest namespace technology

Build the redirection to allow max spread (at any path level)

```
[root@eoshome ~]# eos route ls
/eos/user/1/lmascett/ => *eoshome-l.cern.ch:1094:8000
```
Solution 3: EOSHOME

Deploy multiple empty instances with latest namespace technology

eoshome instance XY
Solution 3: EOSHOME

Deploy multiple empty instances with latest namespace technology
Solution 3: EosHOME

Deploy multiple empty instances with latest namespace technology

Migration scenario similar to Solution 2
- same requirements on the CERNBox side
- same requirements on the migration tools
Let’s go ...
15k Users Data Migration

Data is mirrored in an hidden folder on destination

Migration pseudo-script:

- check user exists
- check user migration status
- block user quota and sync client
- migrate data (MGM query)
  - create folders + ACLs
  - copy files (XS+size match)
  - set correct ETAG for each file
- rsync -av --delete + rsync -av
- sync client verification (propfind comparison)
  - XS+SIZE+ETAG matching
  - No duplicated file & dir IDs
- MGM query comparison
- CERNBox shares migration
- account swap + re-enable
15k Users Data Migration

We started loving very special encodings ...
15k Users Data Migration

... and as well infinite loops ...
EOSUSER HTTP Activity

Massive migration started

Test migration started
Summary and Outlook

Thanks to all the people involved in this activity! Thanks for the hard work!

General improvement of EOS\CERNBox architecture

• removing SPOFs
• improving metadata performance
• reducing drastically downtimes
  • less user impacted
  • almost zero restart time
• flexibility to scale up and out at the same time
• removing big-bang upgrades
  • simplify small scale testing and software rollout
Thanks for the attention!

Questions?

www.cern.ch