EOS file transfer & reliability testing

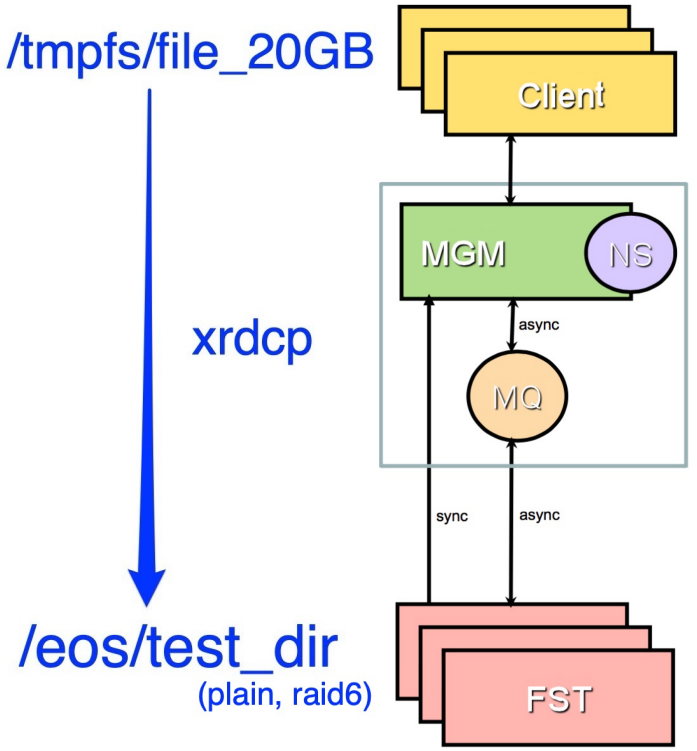
Wenlong Yuan

EOS testing server

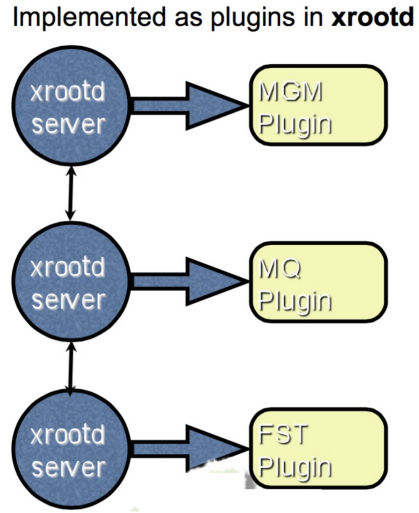
- Using neeps.ph.ed.ac.uk as EOS testing server
- Installed EOS on Docker containers
- Storage: 2TB x 36 disks
- 32G RAM, 32 CPU cores
- Testing file transfer performance and redundancy of EOS

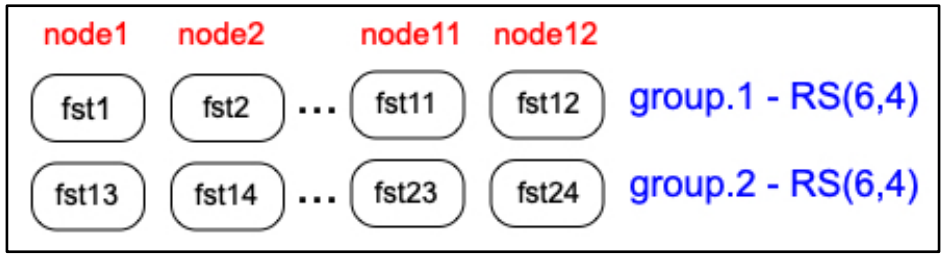
FSTs transfer performance testing

- Building 12 nodes, mounted one 2TB disk per node, also tested two 2TB disks per node with two subgroups
- Filesystem (fs): **xfs,** ext4, ZFS, dir. …
- Using **xrdcp** to transfer a 20GB file from RAM to EOS dir. to test transfer speed
- Using **xrdcp** to transfer 20GB (~50k small files) to test redundancy
- Comparing two layouts
  - Plain: no RAIN
  - Raid6: default **RAIN** (Erasure Coding), **RS(6,4)**
- Erasure Coding affects each subgroup



MGM - metadata server
FST - storage server

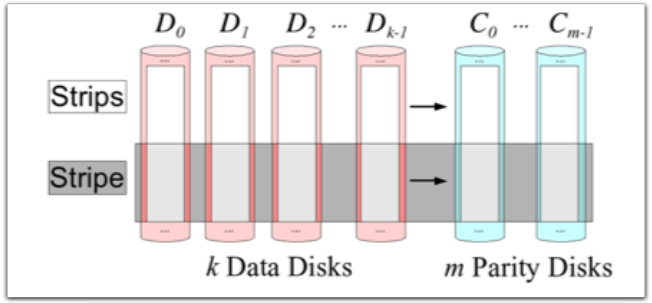


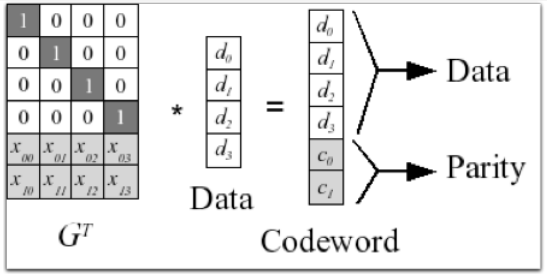


RAIN - reed-solomon encoded files with data and parity blocks

Erasure Coding

- **Erasure Coding** (EC) - data is broken into fragments, expanded and encoded with redundant data pieces, and stored in different locations or storage medias
- Default **Raid6 - RS(6,4)**, each block was encoded into 4 data and 2 parity chunks. The files will be stored in 6 strips, and they should be retrieved up to any 2 of 6 strips are failed
- If one fs/disk is failed(IO error), the **drain system** will trigger file conversion, convert data on failed fs to a new fs






```
  File: '/eos/test-6/f_raid6_1/mlt-files/root/root-6.18.04/README.md'  Flags: 0644
  Size: 5086
Modify: Sat Feb  1 09:46:34 2020 Timestamp: 1580550394.761612000
Change: Sat Feb  1 09:46:34 2020 Timestamp: 1580550394.742103478
Birth : Sat Feb  1 09:46:34 2020 Timestamp: 1580550394.742103478
  CUid: 1001 CGid: 1001  Fxid: 00008945 Fid: 35141    Pid: 6015   Pxid: 0000177f
XStype: adler    XS: 1d a9 e9 25    ETAGs: "9433090359296:1da9e925"
Layout: raid6 Stripes: 6 Blocksize: 1M LayoutId: 20640542
  #Rep: 6
┌────┬──────┬──────────────────────────┬──────────────┬──────────┬──────┐
|no.| fs-id|                      host|     schedgroup|      path|      boot|
└────┴──────┴──────────────────────────┴──────────────┴──────────┴──────┘
 0       7  eos-fst7.eoscluster.cern.ch       default.0       /data1     booted
 1      12 eos-fst12.eoscluster.cern.ch       default.0       /data1     booted
 2       8  eos-fst8.eoscluster.cern.ch       default.0       /data1     booted
 3       4  eos-fst4.eoscluster.cern.ch       default.0       /data1     booted
 4       9  eos-fst9.eoscluster.cern.ch       default.0       /data1     booted
 5      10 eos-fst10.eoscluster.cern.ch       default.0       /data1     booted
```

FSTs file transfer performance

- 20GB sing file Transferring (5 times)
- Raid6 has a significate faster speed
- Raid6 transferring data in parallel

RAIN layouts	Trans 1 (MB/s)	Trans 2 (MB/s)	Trans 3 (MB/s)	Trans 4 (MB/s)	Trans 5 (MB/s)
Plain	125	121	138	132	130
Raid6	363	370	350	384	384

- 20GB small files (~50k) transferring
- Raid6 takes longer time to convert data on stripes
- Based on latest version (took longer time on older versions)
- EC is ideal for large files

RAIN layouts	Transfer time
Plain	20 min
Raid6	1h40min – 2h
Raid6 (xrdcp --parallel 2)	1h10min
Raid6 (xrdcp --parallel 4)	55min

Redundancy Test

- Test on 20G small files (50k), when killing FST nodes/disks, to see how many files could be retrieved
- Direct kill FST nodes, cannot trigger drain system (**no drain, worst situation**)
- Remove FST disks(fs), will trigger drain system (**w/ drain - remove each disk when previous disk draining completed**)

RAIN layouts	1/12 disk fail	2/12 disks fail	3/12 disks fail	4/12 disks fail	5/12 disks fail	6/12 disks fail
Plain	91.6%	83.4%	75.1%	66.7%	58.2%	50.0%
Raid6 (no drain)	100%	100%	91.0%	72.8%	50.0%	28.3%
Raid6 (w/ drain)	100%	100%	100%	100%	100%	100%

Redundancy Test

- Removing 6/12 disks at the same time, drain system failed, only 28% data can be retrieved. Then remounted the 6 lost disks, all data can be found again
- Kill the FST nodes and rebuild the nodes, the system remain the same
- When the drain process failed, all dumped files on the failed fs can be printed

- Some problems during the test
  - xrdcp single large file (20G) may let the nodes offline under some newer versions
  - Drain empty file may cause drain procedure fails

Summary

- A test on Tier 2 level
- EOS docker works well during the test
- Erasure Coding
  - Faster transfer (ideal for large file)
  - Flexible filesystem setup
  - Efficient redundancy space
  - Easy to print dumped files on failed disk