

EOS Deployment and Operations

Sharing information, experiences and suggestions from EOS deployments

Architecture and Design

- Storage design decisions
 - RAID vs HBA
 - SATA vs SAS
 - Planning for data protection
 - Losing a drive
 - Losing a JBOD
 - Planning for performance
 - SAS throughput
 - Network throughput
- Filesystem decisions
 - Type: Ext3/4, XFS, ZFS
 - RAID, strip size
 - E2fsprogs size limit
 - Default blocksize
 - Default fs reservation amounts
 - IO tuning
 - Perf and acceptance testing
- Site filesystem choice, configuration, tuning and testing
 - **ORNL:**
 - 1PB, ZFS, 36TB volumes, SAS drives, 1024 blocksize, read-ahead disabled, No RAIN
 - **NERSC:**
 - 826TB, HWRAID, SAS drives, 1024 blocksize, No RAIN
 - **SUBATECH:**
 - 388TB, ext4, xfs, HWRAID6, No RAIN
- Clarify EOS Design and Expectations
 - Regarding data integrity, “Where will the logic lay?”
 - RAID
 - ZFS, etc.
 - EOS itself (with RAIN and direct drive access?)
 - All are options, but not necessarily together...
- Clarify Performance acceptance testing requirements
 - Simulate number of readers to matched expected CE environment

- Xrdstrss
- Fio, iozone, etc.

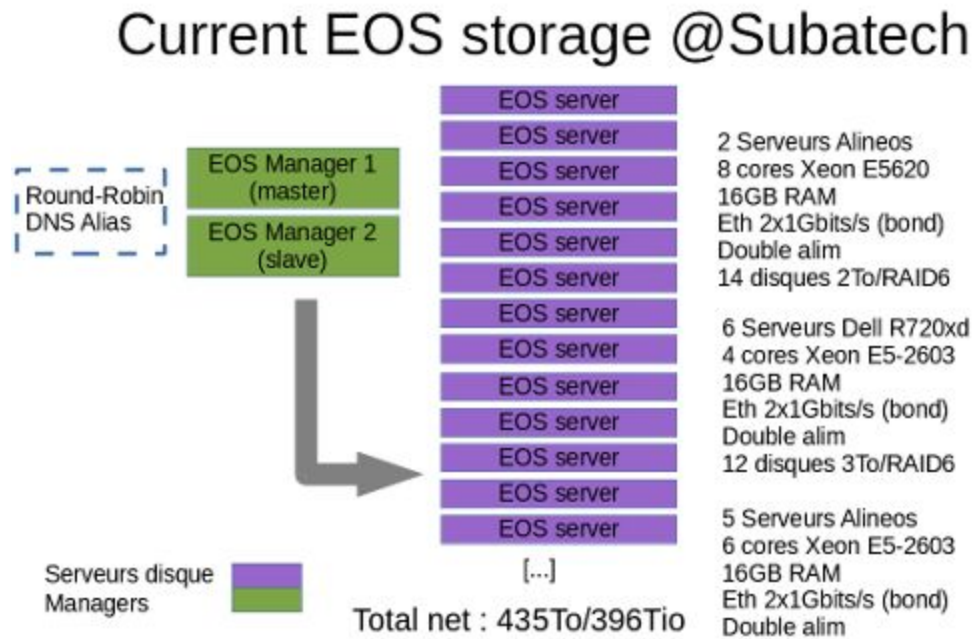
EOS Deployments

ORNL Installation

- Install script
 - SSH keys, dns names, etc. leading to silent failure
 - Required Andrea's assistance to setup EOS
 - Sites perhaps offer improvements to installer via EOS github
- Documentation
 - EOS & xrootd require/assume considerable existing knowledge
 - Documentation lacks coherency
 - EOS Manual (2011) - <http://wiki.ipb.ac.rs/images/f/f7/Eos-manual.pdf>
- EOS needs to set user.foo xattr on files and dirs, ensure file systems are mounted with xattr enabled.
- EOS monitoring traffic not reaching vobox
 - At ORNL root cause was determined to be:
 - EOS reporting udp packets have a ~2400 byte length. Thus the receiving host and intervening network path must support a larger than default 1500 mtu. Monitoring traffic sent to port 9930 using a smaller payload and thus was arriving.
 - Enabling mtu=9000 on vobox.ornl.gov allowed the traffic to be received.
- EOS/CentOS/Iptables interface name issue
 - Long story short: Some part of the stack wasn't liking the interface name to start with a digit.
In any case, renaming 10gbe0 -> xgbe0 and shutting iptables down worked.
 - Interface names need whitelisted
- "ORNL Storage Number Real vs. MonaLisa" thread
 - 192.188.182.* to Wuhan
 - "The ip-to-site mapping should have worked even without this information but another bug prevented it from working properly. This is now fixed and the traffic is correctly assigned to ORNL since a few minutes ago." - Andreas
- **LLNL => ORNL xfer Issues**

- Under-utilized 10G capabilities
 - A lot of emphasis on LLNL <--> ORNL throughput, iperf tests to prepare, etc.
 - In the end it didn't matter, data trickled in from LLNL
 - Faster from other sites, but capped ~400Mbps
- Xfer timeouts issue
 - "LLNL --> ORNL Transfers" thread
 - no FMD record found
- Enabling 3rd party xfers issue
- Confusion over which methods were current
 - Tried setting ftsofs.thirdparty 1
 - Tried various other settings, which turned out to have been deprecated
- ftsofs.thirdparty.slots 50
- ftsofs.thirdparty.slotrate 500

SUBATECH Installation



Not using the RAIN features but having RAID6 partitions.

Installation

Automatized with Foreman+Puppet. Most of the Puppet code is home-developed and may not be easily shared. It includes base configuration :

- Repositories
- Iptables
- Network (bonding od 1Gbits/s interfaces)
- Kernel IP tuning

... and EOS configuration (mostly RPMs + configuration files), resource BDII for reporting of EOS storage in site BDII, Nagios probes.

Dates :

January 2014 : Testing EOS

April 2014 : First EOS cluster in production

June 2014 : Reinstall Managers (one after the other, no service disruption)

March 2015 : Add servers and update to EOS 0.3.118

LBLN Installation

EOS install at LBL was not typical primarily because PDSF systems at NERSC do not use yum or rpms due to the way systems are net-booted and user environments are managed in separate chroot'd OS. The install required us to extract the yum/rpm from the *eos-deploy* script. Method:

- downloaded rpms and their dependencies using repotrack
- extracted rpms using rpm2cpio
- packaged the software in a single compressed cpio archive
- deploy software as an "addon" using the archive

After installing EOS via the "addon", we ran the modified script & it basically worked except (could have been that we broke the script)

Xrd.cf had a spurious extra line (easy to find via log files)

```
/ext/sysconfig/eos on MGM did not have, but were on FST
export MONALISAHOST=palicevo1.nersc.gov
export APMON_INSTANCE_NAME=ALICE::LBL::EOS
export APMON_STORAGEPATH=vg
```

Wrong ownership and permissions on
/etc/grid-security/xrootd/TkAuthz.Authorization

Base directories inside eos were not created:

```
EOS Console [root://localhost] |/eos/alice/grid/> ls -l  
drwxr-s--- 1 10367 1395          0 Apr 08 13:57 00  
drwxr-s--- 1 10367 1395          0 Apr 08 13:57 01
```

...

Why uid:gid 10367:1395? Docs should give guidance & script should ask for specific user.

Prep included (though not obvious):

- Making system hostname public & routable
- Registered all system in DNS

Did not have to use nginx or have services open on 443 and 8443 which was good, since the default install expected some tie in to LDAP which was not expected nor documented.

Error in LDAP configuration wasn't obvious but knowable to experienced AliEn site admin.

eosapmon:

Initially failed due to perl dependencies that we had to add by hand

1. assumed network interface begins with "eth", ours is ib0.
2. Uses rpm -qa to get package version of xrootd
Both were easy fixes once found (look in /usr/local/lib64/perl5/ApMon/ApMon/)

EOS reports TB as bytes/1000^4, ApMon reports TiB as bytes/1024^4

Valuable to have a description of system configuration, e.g.

For /var/eos & /var/log/eos

Differences there between MGM and FST

General Comment:

There appear to be a limited number of configurations for setting up EOS which the eos-deploy script attempts to do for you. A simple checklist of those configs would be good have on the alien website.

Operations: Lesson Learned

ORNL

Network traffic causing exceptionally high IOWait

- Requires restarting EOS / rebooting FST
- Very apparent during heavy ion run
- ‘what we have seen at CERN is that this are low bandwidth but high IOPS requests e.g. small scattered read requests. This are most likely conditions data files opened by thousand(s) of clients.’ Andreas “Conference Call Tomorrow? (19 Feb) re: EOS / High Load Issue” thread

Read ahead issue

- `ansible fsts -s -m shell -a 'cat /sys/module/zfs/parameters/zfs_prefetch_disable'`

Matching eos behavior to filesystem

- It... kind of looks like xrootd is averaging 10k reads
- Assuming that's even the right order of magnitude, then:
 - Is there a way to tune it to larger IO?
 - Should we consider migrating the backing FS to a smaller block size?

EOS throttling

- `eos access set limit 50 rate:user:aliproduct:OpenRead`
- Not implemented at ORNL at this time

Very long time for EOS restart

- `chmod 700 /var/eos/md/*.sql`
- “I am removing this local database, which is too much a bottleneck for the coming release.”

Logging verbosity tuning

- Set EOS logging attribute
- `eos node config * debug.level=notice`

Capacity reporting discrepancy in Alimon

- Restarting eosapmod results in FST not being seen in AliMon
 - Service runs find, no stale pid, no reason shown
- Stopping and Starting the service fixes it

Consider using one of the FSTs (as a backup, not dual) manager

What's Next

- EOS Capacity reporting
 - Available space incorrect
 - Utilization incorrect
- Resolve high IOWait instability issue
 - Maybe it's ZFS, but...
 - Any improvement thus far has been EOS
 - Andreas indicates he is seeing similar issue, with similar trigger, at CERN
- Resume exploring compression
 - Initial tests showed ~10% gain on a few TB test data
 - Would like the paper on compression on xroot trees
- Explore benefits of caching
 - L2ARC
 - ZIL

Subatech

EOS highwatermark and EXT2 reservation

By default, when creating a EXT2 filesystem, 5% of the space is reserved for root to avoid users filling the system disk and having the system hang. This makes little sense in case of data disks. We were not aware of this reservation until EOS writes to data disk started to fail as the highwatermark was not reached but the usable space exhausted. This resulted in 0 length files created instead of directories in the root of the filesystem.

High load on servers and many processes in IOwait state

We also witnessed this on some of the mostly active servers. Seems a bit of kernel tuning makes things better as Christophe Diarra from IPNO already found. To be confirmed.

Bad version of RPM installed during an update

It happened once that a bad version of the RPM xrootd-alicetokenacc was installed after the manager services had been started. The problem only appeared some time later on one of the managers when we had to restart storage following a power cut.

Confusion of EOS master-manager and slave-manager

It happened once without too much damage but causing confusion that we were acting on a manager thinking that it was the master while it was actually the slave. Some commands cannot succeed if executed on the slave manager. We have since a prompt that says while role has the server we are logged into.

EOS manager down because of exhausted RAM

We need a lot of RAM on the managers because the namespace is in RAM. When doing backups of the namespace, we created a big ZIP in /tmp but this file apparently was cached in RAM and we observed several crashes of the manager because of RAM exhaustion.

Problem booting eos-filestystems if the manager have a “stall” flag

This is the latest incident : in our shutdown procedure, it is asked to “stall” the manager before stopping all services. This creates a stall flag which is recorded in the configuration. When we start the services, we remove the flag when everything is fine. However, it seems that, if a FST needs to synchronize with the manager at boot time, the stall flags prevents the operation from succeeding.

Comments and Suggestions

- Better capture and sharing of operational and deployment issues encountered
 - Summarize common issues, pitfalls, etc
- Blueprint overview of EOS / xrootd
 - Upcoming: “T2 in a box talk”

- Architecture
- Services
- Protocols
- Ports
- Define what storage should be tuned for
 - Block size
 - Inode number
 - CNAF::SE encountered some issue, perhaps something to consider?
 - Reservation space
 - Read ahead
 - Etc
 - Otherwise just shooting in the dark
- Kernel Tunings
 - 10G Tunings
 - <https://fasterdata.es.net/host-tuning/linux/>
 - IO Tunings (needs verifying/testing)
 - http://lcg.in2p3.fr/wiki/index.php?title=TCP-Tuning#Tuning_disk_I.2FO
- Improved monitoring / reporting
 - “Real” job efficiency
 - Storage performance
- Clarify clean shutdown procedure
 - Service order on FSTs and manager(s)
 - Database being removed in future release
- Clarify dual / backup MGM roles and deployment
- Address lifecycle management of storage
 - How to remove a file server
 - How to add new servers
 - How to rebalance used space after adding servers
- Bug report / issue tracking sites
 - JIRA <https://its.cern.ch/jira/browse/EOS> (requires CERN authentication)
 - GITHUB <https://github.com/cern-eos/eos/issues>
- Would a chat tool be useful for T2s?
 - Mattermost / Slack / RocketChat / etc.