Setting up a simple Lustre Filesystem

2008-05-07

Stephan Wiesand
DESY - DV -

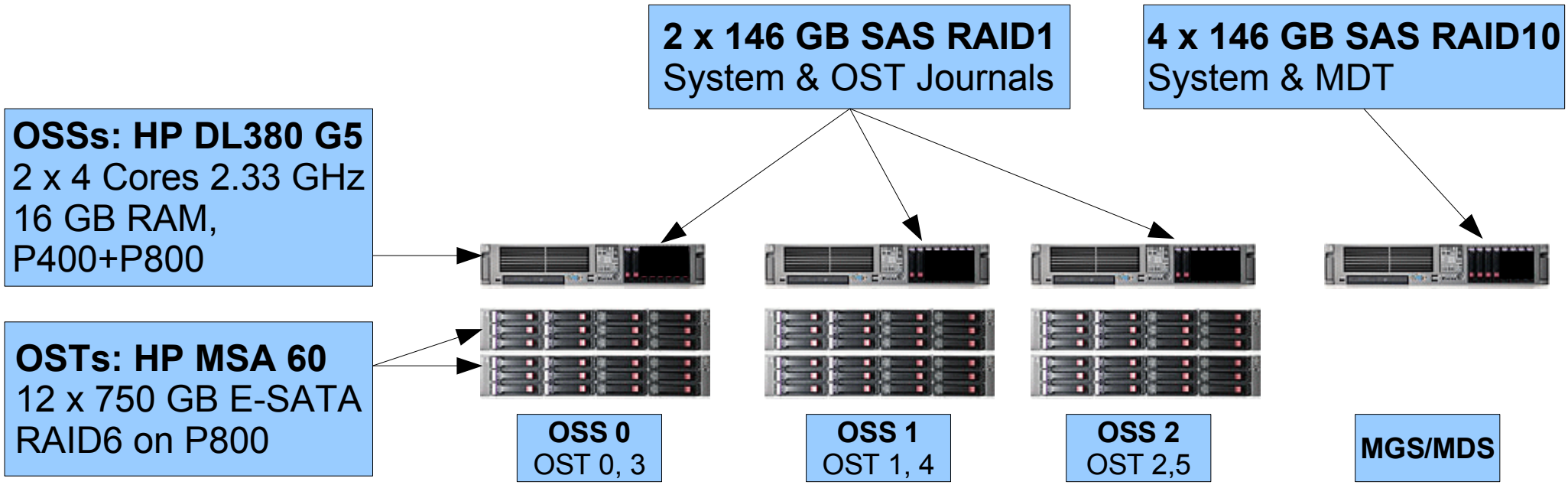
Motivation

- we've been watching Lustre for (>= 5) years
- it always looked promising, but not quite ready for us
- recently, things changed:
  - patchless client (EL4.5+, EL5.x)
  - improved documentation
  - much simplified installation & management (since 1.6)
    - mountconf
  - truly open source
- this may be the right time to start using it in HEP
- we recently made our first serious attempt
- it was rather simple, and results are encouraging

Lustre Terminology

- MGS: "ManaGement Service"
  - one per Lustre site
  - can be co-located on an MDT
- MDS/MDT: "MetaData Server/Target"
  - one per Lustre filesystem
- OSS: "Object Storage Server"
  - a box with storage attached, direct or not
  - one or more per Lustre filesystem
- OST "Object Storage Target"
  - one or more per OSS
  - where files/stripes go

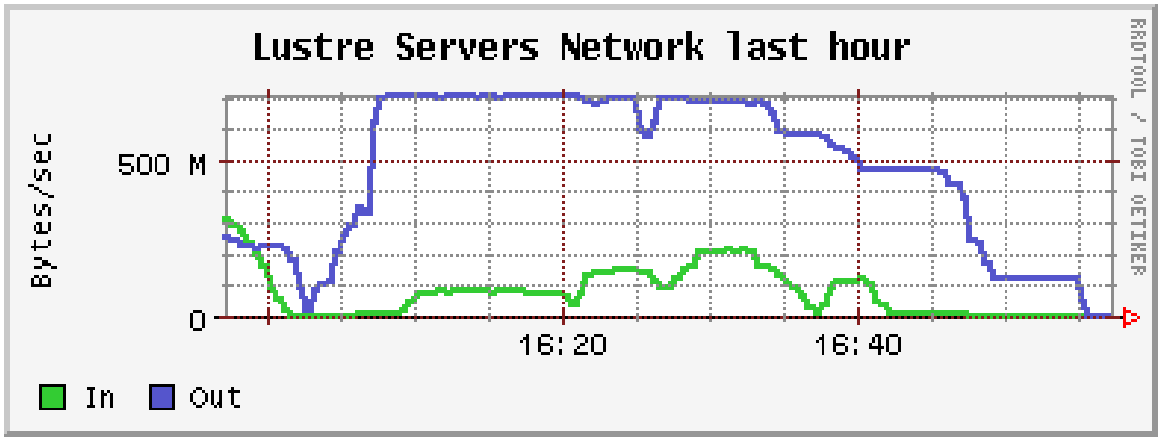
Hardware & Assignment



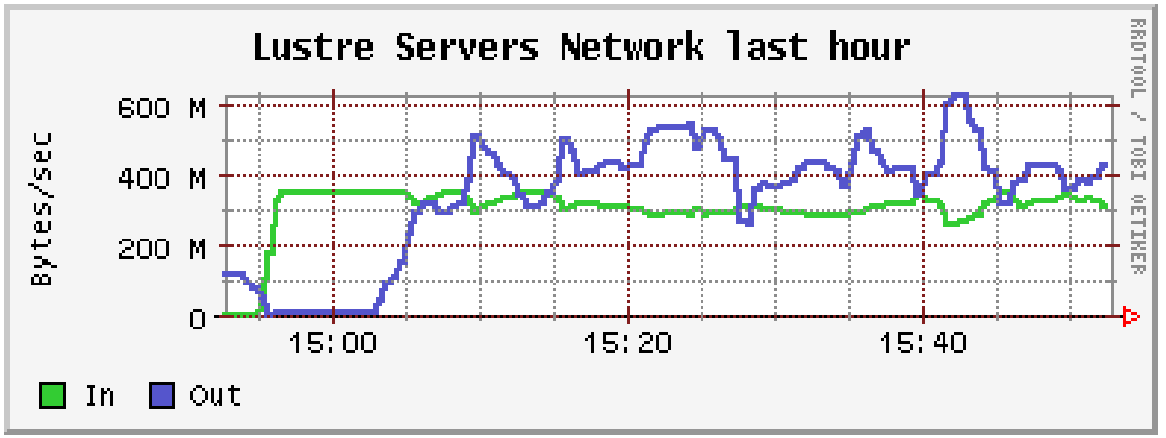
- Scientific Linux 5.1 (x86_64), Lustre 1.6.4.3
- 40 TB usable capacity, Inodes balanced for 1 MB average file size
- 2 bonded GbE links per server
  - 1 Gb/s clients -> server (3 x 118 MB/s)
  - 2 Gb/s server -> clients (6 x 118 MB/s)

Performance

- mostly read:
  saturates 6 GbE links



- mostly write:
  saturates 3 GbE links
- read & write:
  ~ 400 + 300 MB/s



- up to 40 clients
- large files (20 GB), different on each client (no cache eff.)
- large request size (1 MB)

- NO tuning yet
  - except proper stripe (128k) alignment & stride parameters for mkfs

Lustre on the Servers

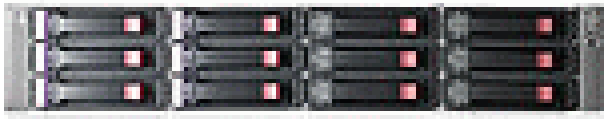
- each target (MDT, OST) is actually a mounted filesystem
  - modified ext3
- mounting the filesystem starts the service
  - kernel threads
- umounting the filesystem stops it
- tune2fs.lustre to modify parameters
- for server use, fs is mounted with type lustre
  - looks empty
- for backups, can be mounted with type ldiskfs
  - MDT will look like the entire Lustre fs
  - all files empty, metadata stored in extended attributes

Servers: OS & Lustre Software

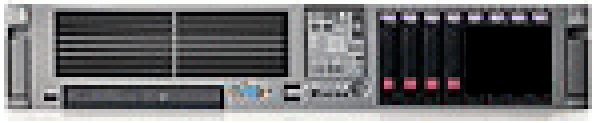
- plain SL5.1
- + Lustre kernel & modules & userland package
  - binary packages as distributed by SUN
    - generally correspond to current RHEL kernels
      - no updates, except with next Lustre release
    - kernel & source is packaged as on EL3 :-(
      - openafs client can be built & run on the servers
- + Lustre version of e2fsprogs
  - required for lustre/ldiskfs
  - binaries as distributed by SUN
    - need to be repackaged for EL5 :-(

Basic Considerations

- maximum OST size
  - 8 TB
    - ldiskfs is a modified ext3
    - RH recently raised support limit to 16 TB
      - but 8 TB still in force for Lustre OSTs
  - MSA60 w/ 12 x 750 GB perfect fit (7.5 TB net)



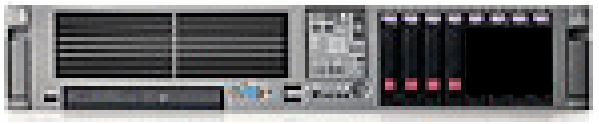
- inode balance
  - 1/file on the MDT, 1/file (or stripe) on OSTs
  - manual recommends 4 kB/inode on MDT (default)
  - ~ 40 TB capacity, 4x146GB RAID-10 on MDS
    - => 1 MB/inode good match



- align all partitions to stripe boundary, tell mkfs about layout

MGS/MDS Setup

- MGS does not need much resources
- MDS does, and is potential bottleneck



  - put MDT on RAID-1 storage (not RAID-5/6)
  - created one RAID-10 array w/ 2 logical drives (16 kB stripes)
    - OS + MDT (MGS co-located)
- MDT holds all metadata for all files
  - complete loss is critical => needs backup
  - backup may be slow (millions of inodes)
  - => decided to place it on a logical volume
    - make backups from LVM snapshot
      - important to backup extended attributes!

MGS/MDS Setup Commands

```
# lvcreate -L 190G -n mgsmds vg00

# mkfs.lustre --fsname=fs1 --mdt --mgs \
  --mkfsoptions="-E stride=4 -E stripe-width=2" \
  /dev/vg00/mgsmds

# tune2fs -i 0 -c 0 /dev/vg00/mgsmds

# mount -t lustre /dev/vg00/mgsmds /mds/fs1
```

- the MGS & MDS are now up and running
- mkfs options rationale:
  - 16 kB stripe size => -E stride = 4
  - 4 disks RAID-10 => 2 data disks => -E stripe-width=2

OSS/OST Setup

- internal disks: RAID-1
  - OS
  - OST journal (on LV)
    - do not put journal on RAID-5/6
    - journal may take up its size in RAM
      - chose 1 GB (we have plenty)
  - 16 kB (default) stripe size
- 1 OST array / 12 disk MSA60
  - 128 kB stripe size, RAID-6
  - 1 stripe aligned OST partition (parted: 128 kiB)
  - set aside a 200 GB partition at end of device
    - for backup & recovery purposes

OSS/OST Setup Commands

```
# lvcreate -L 1G -n j-ost0 vg00
# mke2fs -b 4096 -O journal_dev /dev/vg00/j-ost0

# mkfs.lustre --fsname=fs1 --ost --mgsnode=ringo@tcp0 \
  --mkfsoptions="-E stride=32 -E stripe-width=10
  -J device=/dev/vg00/j-ost0
  -i 1048576" /dev/cciss/c1d0p1

# tune2fs -i 0 -c 0 /dev/cciss/c1d0p1
# mount -t lustre -odata=writeback /dev/cciss/c1d0p1 /ost/0
```

- the first OST is now up and running
- mount OSTs in desired order the first time
  - fs label is assigned upon first mount
- mkfs options rationale:
  - 128 kB stripes => -E stride=32
  - 12 disks RAID-6 => 10 data disks => -E stripe-width=10

Clients

- need the kernel module + lustre userland package
  - not the modified e2fsprogs
  - kernel module can be built against an unmodified SL4/5 kernel
    - since 4.5
  - we build an SL-style kernel-module package
- `mount -t lustre mgshost:/fs1 /lustre/fs1`

Details: SELinux

- disabled on our Lustre servers
  - servers do work with SELinux enabled
  - but an EA "security.selinux="system_u:object_r:unlabeled_t:s0\000" is stored with every file
  - wasteful, may cause problems later
- enabled on almost all clients
  - problems with `mv` into Lustre (also seen with panfs)
    - policy change (handle them like nfs) accepted by maintainer
    - should be in EL5.3 (was too late for 5.2, see BZ #437793)
    - meanwhile, requires modified policy
      - change cannot be applied in a module

Details: Service Threads

- received a warning that servers may create "unlimited" number of threads (mass deletions, ...)
  - although they shouldn't, according to the manual
- automatic thread creation disabled by manually specifying the number
  - chose 256 for both MDS and OSTs
  - set in modprobe.conf:
    - `options ost oss_num_threads=256`
    - `options mds mds_num_threads=256`

Details: Ethernet Bonding

- according to manual, Lustre should be able to load-balance across more than 1 Ethernet link
- in reality it doesn't, and using more than one interface in the same fabric is not recommended
- => running 2 NICs in 802.3ad mode
  - xmit-hash-policy: layer3+4
    - effective even if clients & servers in different layer2 subnets
  - but lustre has to be told which interface to use:
    - `options lnet networks=tcp(bond0)`

Stability Record

- fs is up since ~ 2 months
- no problems with servers
  - survived all testing by us
  - keep surviving actual use by users
    - increasing, but still rather moderate
    - frequent use from O(100) farm jobs
- 2 client panics which are likely to be due to lustre
  - both on WGS, during interactive use
  - probably hit the known stat-ahead bug
    - should be fixed in 1.6.5 (r.s.n)

Summary & Conclusions

- Lustre is worth trying now
  - reasonably simple
    - reading the manual is still a good idea
    - also consult the mailing lists & bug tracker
  - works well with SL on current commodity hardware
    - very good price/performance and price/capacity ratio
  - reasonably stable (especially servers)
- it still has its shortcomings
  - security, OST removal/replacement, free space rebalancing,...
- but it's usable now, and the roadmap (still) promising
- CFS acquisition by SUN probably a good thing