Lustre at GSI
- Evaluation of a cluster file system

Walter Schön, GSI

Topic

- Introduction
  - motivation
  - lustre test cluster
- Performance
  - server
  - controller, RAID level
  - file systems
  - parallel I/O
  - bonding
- Experience
- Outlook

Introduction

Present situation:

data file system: nfs based

Advantages: transparent, posix conform => “like a local disk”

Disadvantages:

- very slow under parallel I/O
- not really scalable
- nightmare with nfs stales under problematic network conditions

Requirements:

- robust
- fully posix conform - existing analysis code should run “out of the box”
- scalable
- open source
- should run on existing hardware
-

=> looking for a scalable cluster file system, having FAIR in mind ....

lustre: www.clusterfs.com

- running on really big clusters
- existing documentation, discussion lists, wikis ...
- good experience with lustre at CEA (HEPIX talk in Hamburg)
- professional support possible e.g. from
  - Cluster File System, Bull,
    Credativ (debian developers)

(minor) technical disadvantage:
production versions still need kernel patches for the server
=> Will the patched kernel work in our environment?

(some) lustre features:

- clients patchless
- server need patch (in future integrated in linux kernel)
- data striping & replication levels
- OSS fail over/fail out mode possible
- Fill balancing (configurable)
- RAID 0 over network, RAID 5 over network in alpha version
- Underlaying FS is an improved version of ext3
- XFS “in principle” possible however this is not the default
- after ZFS on the horizon?

lustre look & feel

Starting with lustre: creating lustre fs

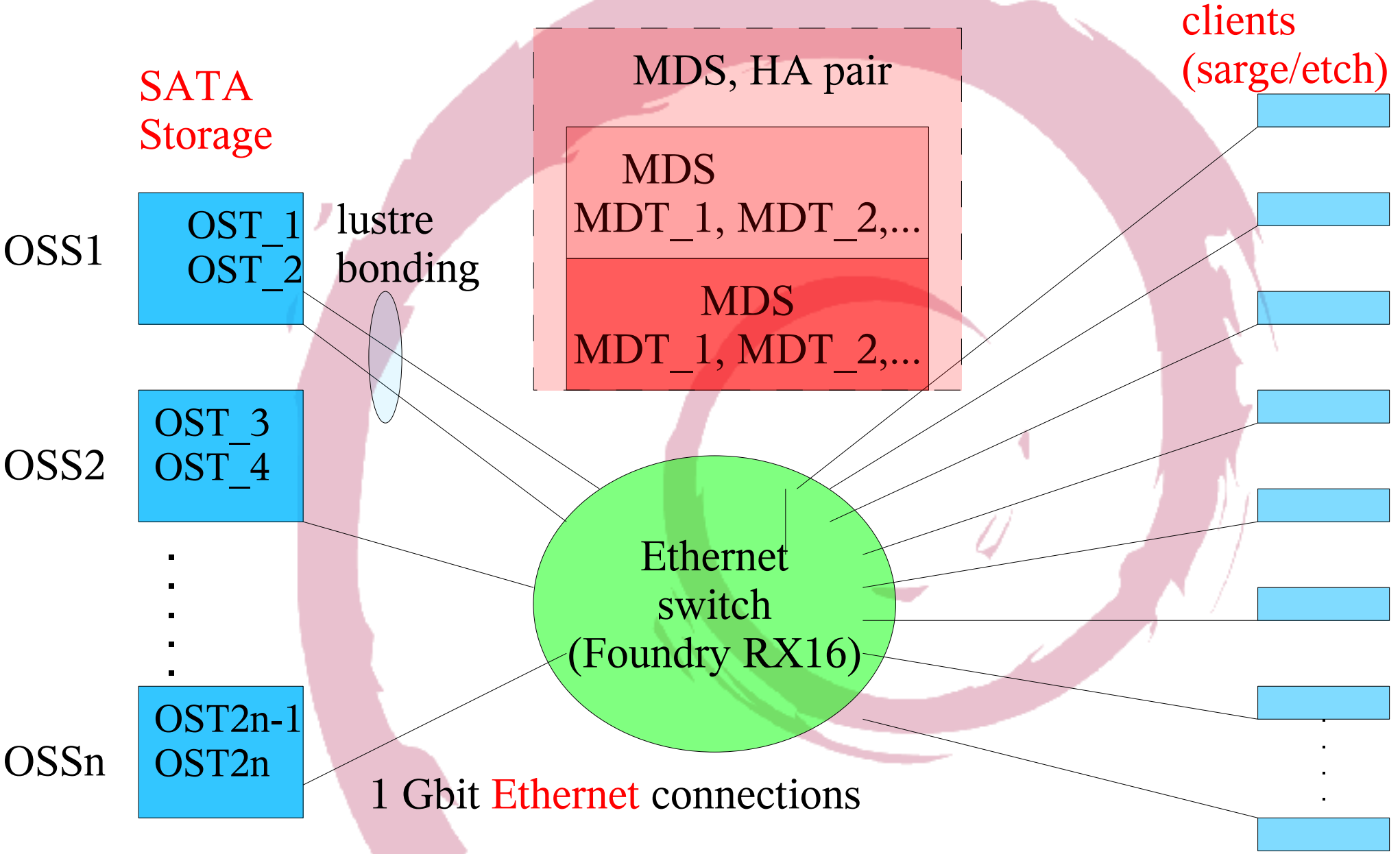
mkfs.lustre
mount -t lustre

creating MDT:
mkfs.lustre –fsname /dev/MGS-Partition
mount -t lustre /dev/MGS-Partition /MGS-MOUNTPOINT
creating OST: similar
mount client:
mount -t lustre MGS@tcp0:/DATEISYSTEM /MOUNTPOINT

However: messages are strange ..... :-)

lustre Testcluster: Architecture

running lustre 1.6.x (recently 1.6.3), debian, 2.6.22 Kernel



lustre Testcluster:

hardware based on SATA storage and Ethernet connections

OSS in “Fail out mode”

Number of MDS: ---------- 1
Number of MDT's : ---------- 3
Number of OSSs : ---------- 12
Number of OSTs : ---------- 24
Number of RAID controllers: 24
Number of data disks : ------ 168
Size of file systems: -------- 67 TB
Number of clients : -------- 26
Number of client CPU's --- 104

default striping level: 1
default replication level:1

cost (server + disks) : 42.000 Euro

3 HE server

- redundant power supplies
- LOM modul
- redundant fans
- excellent cooling of disks, memory, CPU
- 16 slot SATA, hot swap
- 14 slots for data
- 2 slots for RAID 1 system
- 2 SATA RAID controller
- 4/8 GB RAM
- Dual CPU Dual core
- 500 GB disks WD RAID ed.
  24x7 cert.,
  100% duty cycle cert.

5,6 TB per 3 HE RAID 5
73 TB per RACK


1 Rack:
13 servers
73 TB

Performance – where is the bottleneck?

The RAID controller: 3W9650, 8 channel RAID 5/6
WD 500 GB, RAID edition, 100% duty cycle, 7x24
Check: Memory to disk performance: as function of

- number of disks in RAID array (6 or 8)
- filesystem (ext3, XFS .....)
- kernel parameters (read ahead cache, nr_requests, max_sectors_kb....)

Measuring tool: IOZONE
using really huge transmitted files (size >> RAM) to avoid
cashing effects..... and biased results!

RAID level, filesystems, Kernel parameters.....

! memory to disk performance !

#disks	filesystem	RAID level	kernel param	write [MB/s]	read [MB/s]
6	ext3	6	default	66	81
8	ext3	6	default	91	97
6	XFS	6	default	140	95
8	XFS	6	default	190	100
6	XFS	5	default	192	122
8	XFS	5	default	227	122
6	EXT3	6	opt	66	180
6	EXT3	5	opt	72	180
6	XFS	6	opt	145	180
8	XFS	6	opt	205	380
8	XFS	5	opt	260	490

Summary of the RAID controller/disk/file system test: (valid only for the tested combinations)

- 8 disks are more than 33% faster than 6 disks .....
- RAID5 is about 30% faster than RAID 6
- XFS is much faster than ext3
- especially the read performance can be optimized by tuning kernel parameters
- The new generation of SATA controller is really fast .....

What does this conclusions mean for the performance tests?

conclusion for the lustre test?

- The controller could be the bottleneck, if the data are focussed on 1 OST with 6 disk RAID if lustre ext3 is as slow as “native” ext3 .... a 1Gbit Ethernet connection is about 115 Mbyte/s .....

How fast is the modified ext3 used by lustre?

lustre performance test

test setup: 1 client connetcted via 1 Gbit (using iozone)
data transfer via lustre

#disks	filesystem	RAID lvl.	kernel par.	write	read	network
6	lustre-ext3	6	default	80	80	1 Gb/s
8	lustre-ext3	6	default	112 MB/s	113MB/s	1Gb/s
6	lustre-ext3	5	default	114 MB/s	114MB/s	1Gb/s
			for comparison the m2d results:			
6	ext3	6	default	66MB/s	81MB/s	-
8	ext3	6	default	91MB/s	97MB/s	-

=> conclusion:

- lustre can saturate easily a 1 Gb connection
- lustre-ext3 ist faster that “native” ext3 but slower as XFS
- the combination 6 disks/RAID6 is a bottleneck

lustre – testing a cluster

setup:

- MDT with 20 OST on 10 OSS with 1 Gbit Ethernet connection
- => cumulated I/O bandwidth in maximum 10x 1 Gbit
- up to 25 clients using 100 I/O jobs parallel
- OST with 6 disks RAID5
- OST with 8 disks RAID6
- testing with IOZONE in cluster mode:
  cluster mode: IOZONE read list of hosts to connect and starts the test
  until the last host is connected to avoid wrong numbers

lustre cluster performance – the results

# OSS	#OST	#clients	#processes	I/O	I/O per OSS
6	7	7	7	544 MB/s	91 MB/s
5	10	20	40	480 MB/s	96 MB/s
10	20	25	100	970 MB/s	97 MB/s

conclusions:

- lustre scales very well
- in our setup limited by the network connection
- lustre bonding effective?

lustre bonding

Test setup: 1 OSS connected with both Ethernet cables

• activating lustre bonding

#OSS	#OST	bonding	#clients	write [MB/s]	network
1	2	on	2	225	2 x 1 GB
1	2	off	2	114	1 x 1 GB

Test: put one cable out of the OSS
=> everything works fine, only the I/O drops to 115 MB/s

conclusion:
• lustre bonding is a “cheap” method to double the I/O performance
• In addition you get a redundant network connection

Reliability and robustness of the lustre test cluster

- Test: cluster in “fail out” mode
- “destruction” of a OSS
  - regular shutdown
  - cut Ethernet connection
  - put 2 disks out of a RAID5 during operation..... :-)

Result: after short “waiting for answer” time (configurable?, the system works o.k. - of course, the files on the missing OST's delivers “not found” messages
After relaunch of the OSS, the missing files are present too....

missing/testing:

- MDS as HA cluster
- a long term many user test for reliability and data integrity
- disaster recovery

Mass storage: lustre connection to tape robot

- first attempt to use gStore (the GSI mass storage) was successful

practical experience with lustre:

- "easy" setup of a cluster
- good documentation – however still with bugs
- alpha version and early production version of the patchless client with bugs – getting better now
- problem with OST >2TB / 32 bit OS. However: solved fast
- still patches needed for the kernel – however no real problem found yet
- messages and error codes cryptic – need experience to speak "lustre"
- mixed operation of different versions of the patchless client and server possible

wishlist:

- more "intuitive" messages from the system
- "management" tool for the 1.6x lustre would be nice

lustre – final conclusions

- excellent scalability – excellent I/O
- installation and configuration was straight forward
- integration in existing hardware and storage without problems
- test user are happy to use lustre mounted file systems feels like “local disk”
- our large experiments are happy to use really huge “disks”
- looking forward for the lustre network RAID 5 feature (end 2007)

=> now testing with “real” users

- if successful, the data file system will be moved to lustre generating a 700 TB file system.