ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CSCS
Swiss National Supercomputing Centre

Parallel file systems for Brutus

Adrian Ulrich

Teo Brasacchio

Overview

- File systems on Brutus
- HA-Lustre setup (Snowbird)
- Experiences with Lustre
- Problems with the 'turnkey' solution
- Client setup
- Monitoring
- Upgrade to 1.8.4
- Questions and answers

File systems on Brutus (serial)

- NFS (12TB)
  - For user homes, applications and batch system
  - Only file system on Brutus with a full backup
  - EMC Symmetrix storage / Solaris (ZFS) NFS HA-servers
  - Attached via 2x 2 GbE (trunked)

- NAS shares (48TB)
  - About 24 shares
  - Operated by ID-Storage team
  - Attached via 2x 1 GbE (trunked)

File systems on Brutus (parallel)

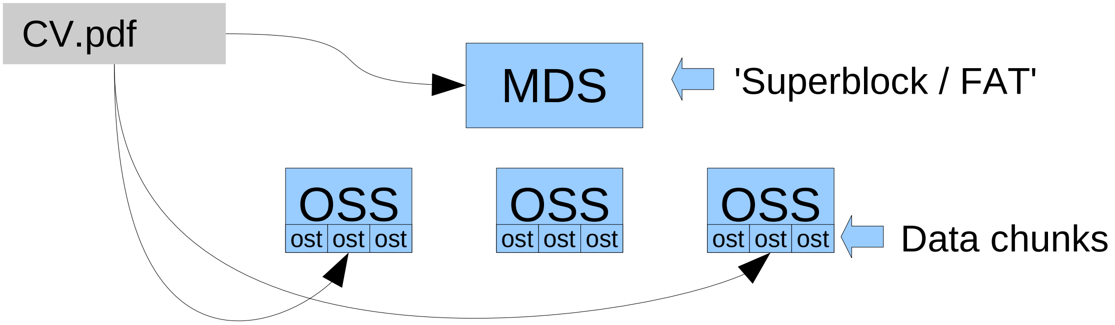
- Panasas (45TB)
  - Work space
  - No backup
  - Raid1 → Raid5
  - Attached via 3x 4x 1 GbE (trunked)

- Lustre – Oracle Snowbird 'turnkey' solution (253TB)
  - Scratch space
  - No backup
  - Raid6 + HA-setup
  - 8x QDR (8x 40Gb)

Why Lustre?

- Runs on Linux
- GPL
- Performance scales with the growth of the file system
- Cheap
- Community (mailing-lists / Bugzilla)
- Commercial 'support' available
- Stable (mostly)

Lustre: quick overview



MDS: Meta Data Server

OSS: Object Store Server
OST: Object Store Target

Stored on MDS

```
{ name=>'CV.pdf',
 stripe_count=>2,
 stripe_size=>512,
 ost=>[0x0001, 0x007],
 .... }
```

OSS01 -> OST 0x0001
Byte 0-511

OSS03 -> OST 0x0007
Byte 512-1023

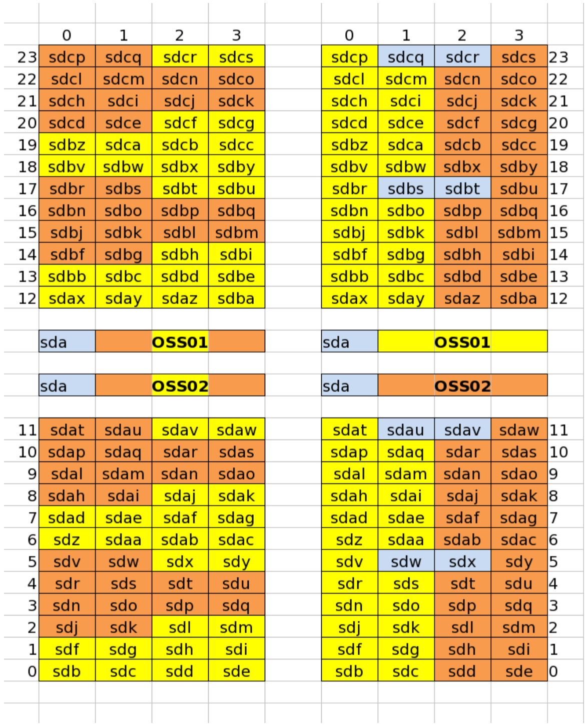
Why HA-Lustre (our setup - 'Snowbird')

- Hardware breaks:
  - 10 Server
  - 36 SAS connections
  - 17 JBODs
  - 396 Disks (1TB)
  - 10x IB QDR + 40x GbE
  - (yet) unknown hardware
- Easier to maintain (software upgrades)
- Complexity versus stability

Lustre setup (Snowbird)

- Logical disk layout
  - RAID6 + external journal + external bitmaps (253TB)
  - RAID6 + internal journal/bitmaps and hot-spares (285TB)
- Physical disk layout

Supported layout



	0	1	2	3
23	sdcp	sdcq	sdcr	sdcs
22	sdcl	sdcm	sdcn	sdco
21	sdch	sdci	sdcj	sdck
20	sdcd	sdce	sdcf	sdcg
19	sdbz	sdca	sdcb	sdcc
18	sdbv	sdbw	sdbx	sdby
17	sdbr	sdbs	sdbt	sdbu
16	sdbn	sdbo	sdbp	sdbq
15	sdbj	sdbk	sdbl	sdbm
14	sdbf	sdbg	sdbh	sdbi
13	sdbb	sdbc	sdbd	sdbe
12	sdax	sday	sdaz	sdba
	sda	OSS01		
	sda	OSS02		
11	sdat	sdau	sdav	sdaw
10	sdap	sdaq	sdar	sdas
9	sdal	sdam	sdan	sdao
8	sdah	sdai	sdaj	sdak
7	sdad	sdae	sdaf	sdag
6	sdz	sdaa	sdab	sdac
5	sdv	sdw	sdx	sdy
4	sdr	sds	sdt	sdu
3	sdn	sdo	sdp	sdq
2	sdj	sdk	sdl	sdm
1	sdf	sdg	sdh	sdi
0	sdb	sdc	sdd	sde

0	1	2	3	
sdcp	sdcq	sdcr	sdcs	23
sdcl	sdcm	sdcn	sdco	22
sdch	sdci	sdcj	sdck	21
sdcd	sdce	sdcf	sdcg	20
sdbz	sdca	sdcb	sdcc	19
sdbv	sdbw	sdbx	sdby	18
sdbr	sdbs	sdbt	sdbu	17
sdbn	sdbo	sdbp	sdbq	16
sdbj	sdbk	sdbl	sdbm	15
sdbf	sdbg	sdbh	sdbi	14
sdbb	sdbc	sdbd	sdbe	13
sdax	sday	sdaz	sdba	12
sda	OSS01			
sda	OSS02			
sdat	sdau	sdav	sdaw	11
sdap	sdaq	sdar	sdas	10
sdal	sdam	sdan	sdao	9
sdah	sdai	sdaj	sdak	8
sdad	sdae	sdaf	sdag	7
sdz	sdaa	sdab	sdac	6
sdv	sdw	sdx	sdy	5
sdr	sds	sdt	sdu	4
sdn	sdo	sdp	sdq	3
sdj	sdk	sdl	sdm	2
sdf	sdg	sdh	sdi	1
sdb	sdc	sdd	sde	0

Ideal layout

Problems with the 'turnkey' setup

- Linux blockdev naming not ideal: sda != c0t0d0s2
  - Solution: udev rules

```
KERNEL==”sd*” PROGRAM=”/bin/namefoo %k”, SYMLINK+=”lustre/%c”
```

- Failover trigger: network not monitored

- Benchmarks (IOzone vs. MPI I/O)

- Lustre itself stable – most crashes caused by LSI driver
  - Version 4.18.0 : Crashes & eats your data (CH)
  - Version 4.20.4 : Eats your data (DE)
  - Version 4.18.4 : Stable (?) (US)

Client setup

- Brutus runs CentOS 5.5 (almost 'vanilla')
- Not using the Oracle RPM: We build our own

  Why?
  - Flexibility (bugfixes / custom kernel)
  - Deployment

Client setup - Deployment

- Mounting lustre at boot can be hard:
  - IB might still be 'down'
  - Lustre does not play well with NFS:
    - statd / lockd could 'steal' port 988
    - 988 could be in TIME_WAIT (SunRPC)

- Solution:
  - RPM *%post* changes */etc/sysconfig/nfs* (lockd + statd)
  - Initscript tries to mount lustre for 60 seconds (tcp_fin_timeout)
  - Before: 30% failure / Now: 99.99% success ;-)

Client setup - Deployment

TIME_WAIT - We are not alone:

Solution of LLNL:

```
--- linux+rh+chaos.orig/net/sunrpc/xprtsock.c
+++ linux+rh+chaos/net/sunrpc/xprtsock.c
@@ -960,8 +960,11 @@ static void xs_udp_timer(struct rpc_task
-       unsigned short rand = (unsigned short) net_random() % range;
-       return rand + xprt_min_resvport;
+       unsigned short rand;
+
+       do {
+               rand = (unsigned short) net_random() % range;
+               rand += xprt_min_resvport;
+       } while (rand == 988 || rand == 922); /* hard coded blacklist */
+
+       return rand;
 }
```

Monitoring

- Enable disk-scrubbing!

  ```
  $ echo check > /sys/block/mdX/md/sync_action
  ```

  Lost about 5% of our disks on first scrub!

- netconsole.ko : Post-mortem info for HARD crashes

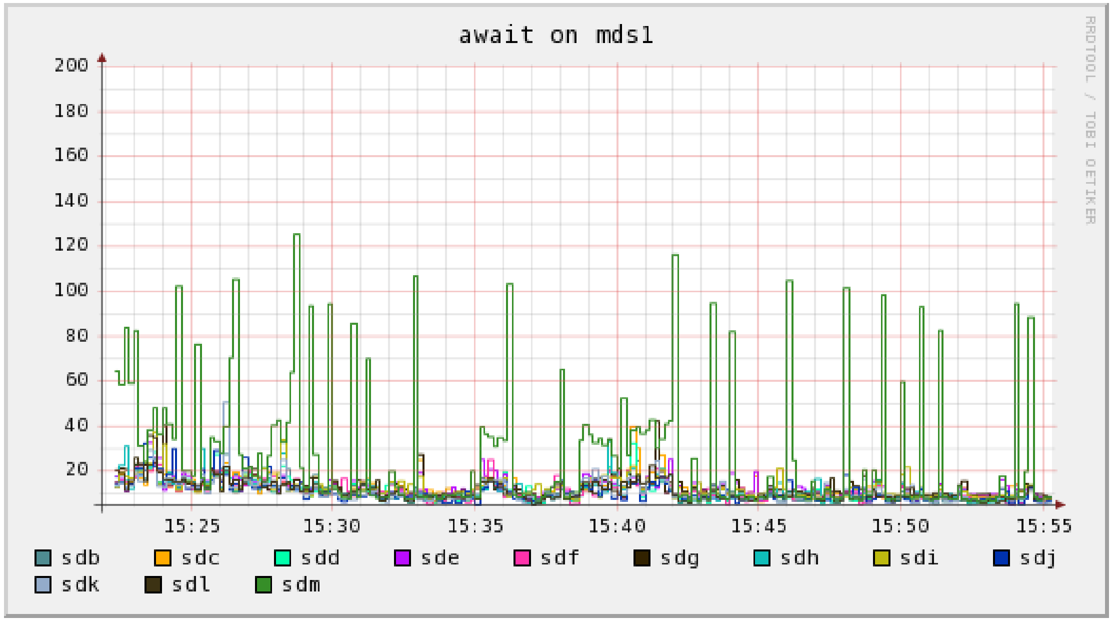
- Install 'blinkenlights' and 'SunSEUS' (fwdl_app) from

  http://dlc.sun.com/linux_hpc/yum/sunhpc/2.0.2/rhel/base/x86_64/SunHPC/

  → Blinkenlights + SunSEUS obsolete CAM

Monitoring - Health and performance

- Faulty drives: Hobbit + `mdadm --monitor`
- Host Status : Hobbit + Ganglia (could also use ha.d)
- Performance: Lustre::Info + Hobbit – Interesting results:



sd[b-l] : Hitachi

sdm : Seagate

Monitoring - 'Realtime' performance

- Lustre::Info

```
$ w3m http://search.cpan.org/~adrian/
$ perl -MCPAN -e 'install Lustre::Info'
```

- Perl interface to /proc/fs/lustre
- Includes 'lustre-info.pl'

Upgrade to 1.8.4

- Successful

Questions ?

Backup slides

Client Setup - Bugfixes

- Discovered an ugly bug while doing something silly

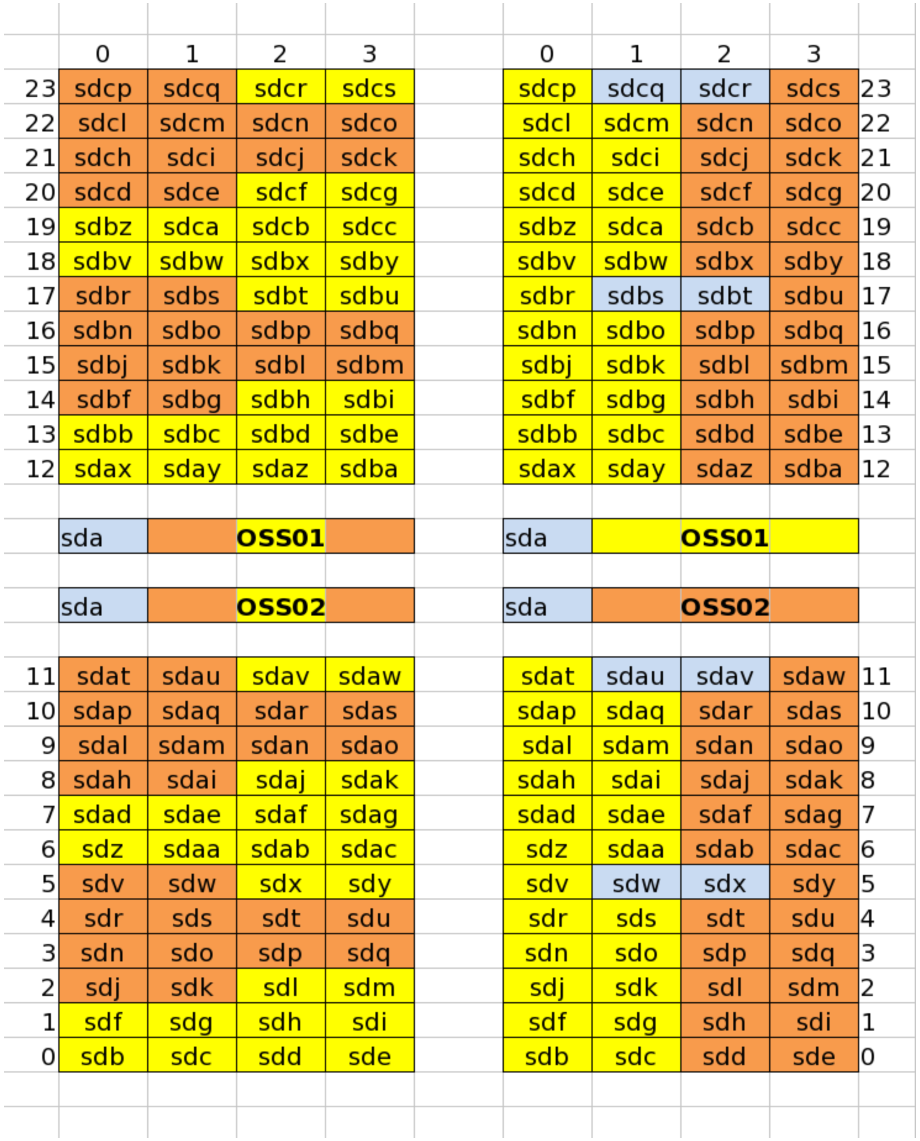
```
$ setfattr -n lustre.lov .  # = kernel panic
```

- Now known as Bugzilla #22187

Option A: Wait until next version is released

Option B:
1. Grab patch from Bugzilla
2. `$ cd 1.8.1.1 && ./lustre.SlackBuild`
3. Upgrade clients
4. Problem solved

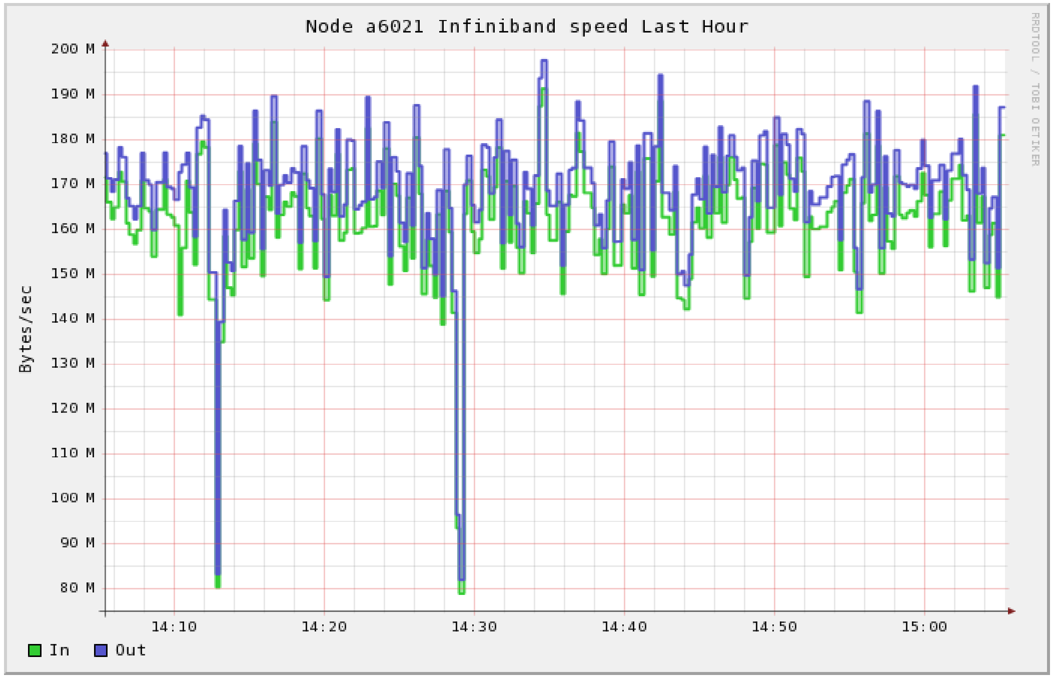
Disk layout



	0	1	2	3		0	1	2	3	
23	sdcp	sdcq	sdcr	sdcs		sdcp	sdcq	sdcr	sdcs	23
22	sdcl	sdcm	sdcn	sdco		sdcl	sdcm	sdcn	sdco	22
21	sdch	sdci	sdcj	sdck		sdch	sdci	sdcj	sdck	21
20	sdcd	sdce	sdcf	sdcg		sdcd	sdce	sdcf	sdcg	20
19	sdbz	sdca	sdcb	sdcc		sdbz	sdca	sdcb	sdcc	19
18	sdbv	sdbw	sdbx	sdby		sdbv	sdbw	sdbx	sdby	18
17	sdbr	sdbs	sdbt	sdbu		sdbr	sdbs	sdbt	sdbu	17
16	sdbn	sdbo	sdbp	sdbq		sdbn	sdbo	sdbp	sdbq	16
15	sdbj	sdbk	sdbl	sdbm		sdbj	sdbk	sdbl	sdbm	15
14	sdbf	sdbg	sdbh	sdbi		sdbf	sdbg	sdbh	sdbi	14
13	sdbb	sdbc	sdbd	sdbe		sdbb	sdbc	sdbd	sdbe	13
12	sdax	sday	sdaz	sdba		sdax	sday	sdaz	sdba	12
	sda		**OSS01**			sda		**OSS01**		
	sda		**OSS02**			sda		**OSS02**		
11	sdat	sdau	sdav	sdaw		sdat	sdau	sdav	sdaw	11
10	sdap	sdaq	sdar	sdas		sdap	sdaq	sdar	sdas	10
9	sdal	sdam	sdan	sdao		sdal	sdam	sdan	sdao	9
8	sdah	sdai	sdaj	sdak		sdah	sdai	sdaj	sdak	8
7	sdad	sdae	sdaf	sdag		sdad	sdae	sdaf	sdag	7
6	sdz	sdaa	sdab	sdac		sdz	sdaa	sdab	sdac	6
5	sdv	sdw	sdx	sdy		sdv	sdw	sdx	sdy	5
4	sdr	sds	sdt	sdu		sdr	sds	sdt	sdu	4
3	sdn	sdo	sdp	sdq		sdn	sdo	sdp	sdq	3
2	sdj	sdk	sdl	sdm		sdj	sdk	sdl	sdm	2
1	sdf	sdg	sdh	sdi		sdf	sdg	sdh	sdi	1
0	sdb	sdc	sdd	sde		sdb	sdc	sdd	sde	0

Monitoring – Traffic

- Ganglia tracks **RAW** Infiniband traffic
- Does NOT use lustre counters / also catches MPI traffic
- Powered by Perl, XS and some (obscure) libibmad calls


Node a6021 Infiniband speed Last Hour
Bytes/sec
200 M
190 M
180 M
170 M
160 M
150 M
140 M
130 M
120 M
110 M
100 M
90 M
80 M
14:10
14:20
14:30
14:40
14:50
15:00
🟩 In
🟦 Out
RRDTOOL / TOBI OETIKER