

LVM, Sysfs and Me

chris procter

What we have

Some storage arrays:

NetApp Filers (in fibre channel mode)

HP EVA x000's

HP XP 512's

A bunch of Linux servers.

Some pretty awful tools

Sysfs

Window on the kernel like procfs

Reflects the devices attached to the system

Read and write device attributes

But:

Sysfs is not documented or even frozen.

Listing devices

```
$ ls -F /sys/block/
```

```
loop0/  loop4/  ram0/   ram12/  ram2/   ram6/   sda/  
loop1/  loop5/  ram1/   ram13/  ram3/   ram7/   sdb/  
loop2/  loop6/  ram10/  ram14/  ram4/   ram8/   sdc/  
loop3/  loop7/  ram11/  ram15/  ram5/   ram9/   sr0/
```

Adding new disks

```
$ ls /sys/class/scsi_host/  
host0 host1
```

```
$ echo "- - -" > /sys/class/scsi_host/host0/scan
```

Present new disk from SAN

Rescan SCSI bus

Figure out what new disk appeared as!

Listing Devices v2.0

```
$ ls -d /sys/bus/scsi/devices/*/block*
```

```
/sys/bus/scsi/devices/0:0:3:0/block:sda  
/sys/bus/scsi/devices/0:0:3:1/block:sdb  
/sys/bus/scsi/devices/0:0:3:2/block:sdc  
/sys/bus/scsi/devices/0:0:3:3/block:sdd  
/sys/bus/scsi/devices/1:0:0:0/block:sr0
```

Shows all scsi block devices

What we do.

Take snapshots of live databases

Present them into development servers

Rinse

Repeat

Problem!

LVM makes extensive use of ID numbers

Duplicate ID numbers are a problem!

Changing UUIDs

```
vgchange -u myvg
```

```
pvchange -u /dev/sda
```

```
vgcfgbackup -f ${TMPFILE} ${CURRVG}
```

Edit the file to change the LV ids

```
vgcfgrestore -f ${TMPFILE} ${CURRVG}
```

Problem (still!)

LVM makes extensive use of ID numbers

Duplicate ID numbers are a problem!

Presenting snapshots

Take snapshot of server A

Present it to server Z

Change UUIDs

Remove from server Z

Present to server B

Removing Disks

Can just unpresent them.

But you get:

```
/dev/sda: read failed after 0 of 4096 at 0:  
Input/output error
```

With every LVM operation.

Removing the old disks

```
$ echo 1 > /sys/block/sda/device/delete
```

Can be a little hairy if you're not careful!

Moving Data Around.

Move Data from old SAN to new SAN

No rebooting

No downtime

Therefore no: cp, tar, cpio, dd etc etc etc

What Uses What?

```
cat /sys/block/sda/device/vendor
```

```
cat /sys/block/sda/device/model
```

Gives us the make and type of the SAN controller

e.g. HP OPENXP

HP HSV200

NETAPP LUN

What Uses What?

```
for DSK in `ls /sys/block/`  
do  
    if [ -e "/sys/block/${DSK}/device/vendor" ]  
    then  
        VENDOR=`cat /sys/block/${DSK}/device/vendor`  
        MODEL=`cat /sys/block/${DSK}/device/model`  
        echo "${DSK} ${VENDOR} ${MODEL}"  
    fi  
done
```

How much space do we need?

```
$ cat /sys/block/sda/size
```

Its in blocks so multiply by 512!

Moving the data

```
$ vgextend myvg /dev/sdb
```

```
$ pvmove /dev/sda /dev/sdb
```

```
$ vgreduce myvg /dev/sda
```

Cant move root filesystem.

But we can restart an interrupted move!

Adding more disk space

Databases grow to fill the available space +10%

Easy to grow a volume on a SAN

Resize devices

```
$ echo "1" > /sys/block/sda/device/rescan
```

BUT:

LVM doesn't like it :-(

So add a new disk and use `vgextend / lvextend`

Questions.

Can you persuade LVM to detect a resized disk?

Can you move an lv between volume groups?

What else can you (mis-)use sysfs for?

Send me your ideas
(so I can claim them as my own!)

chris-procter@talk21.com

SCSI id

0:0:3:1

Host : bus : target : lun

So

```
echo "0 0 3" > /sys/class/scsi_host/host0/scan
```

Listing Devices v3.0

```
$ ls -d1 /sys/bus/scsi/drivers/sd/*/block*
```

```
/sys/bus/scsi/devices/0:0:3:0/block:sda
```

```
/sys/bus/scsi/devices/0:0:3:1/block:sdb
```

```
/sys/bus/scsi/devices/0:0:3:2/block:sdc
```

```
/sys/bus/scsi/devices/0:0:3:3/block:sdd
```

Shows all scsi block devices