RED HAT :: NASHVILLE :: 2006

SUMMIT

Storage, Uninterrupted

Tom Coughlan & Rod Nayfield

Causes of data unavailability

- Planned interruptions
  - to add or reconfigure storage
    - increase filesystem size
  - backup
- Unplanned interruptions
  - hardware failure
    - HBA, cable, switch, storage controller, disk drive
  - repair time

Solutions for data availability

- Hardware redundancy, with automatic failover and recovery
- Hardware hotplugging, for:
  - on-line repair
  - on-line hardware addition and reconfiguration
- Filesystem expansion to incorporate new capacity
- Application pause, data snapshot, then resume
  - backup while the application continues

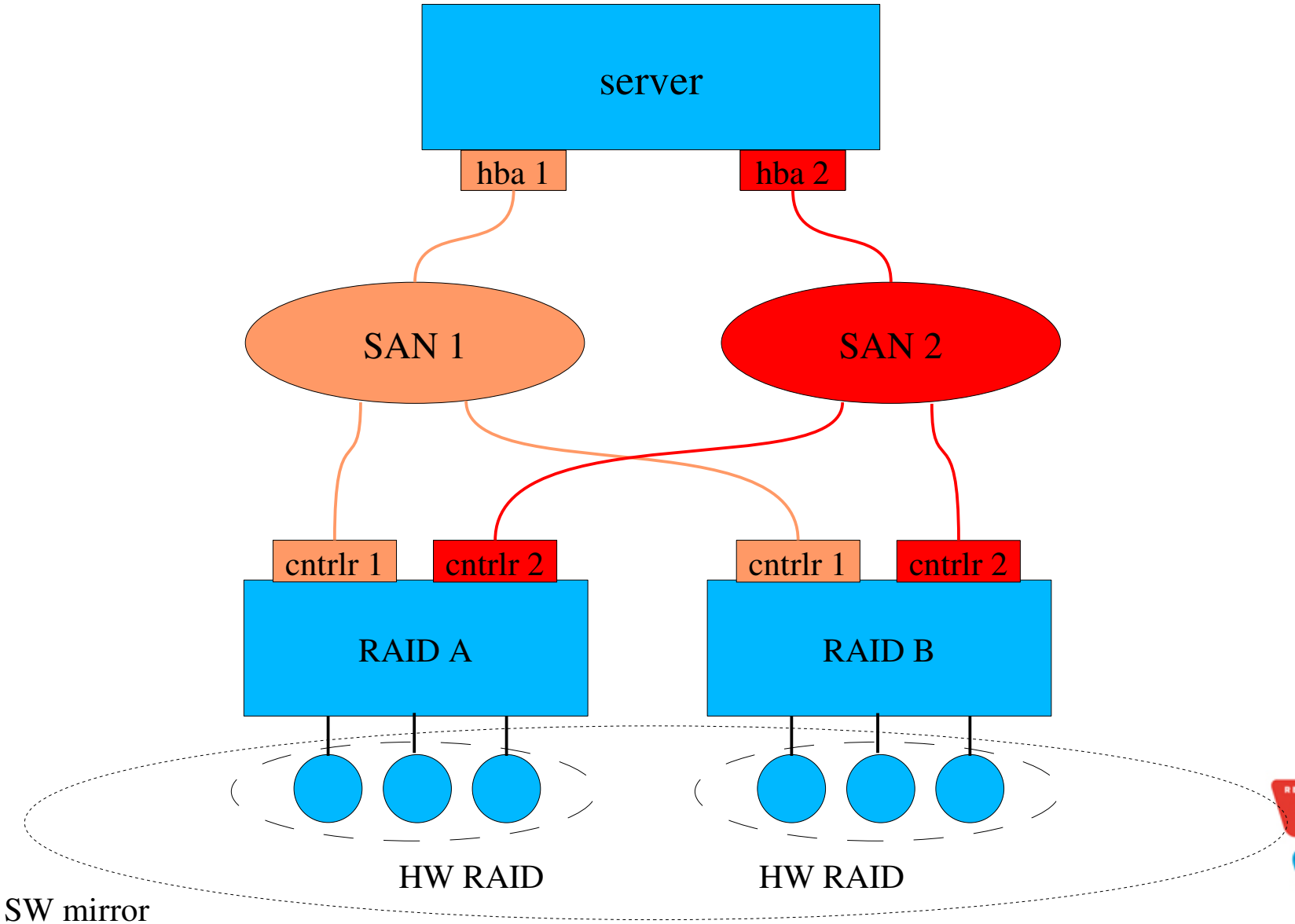
Sounds expensive?

- All you need is RHEL!
  - multipath
  - mirroring
  - hot plug
  - expandable logical volumes
  - expandable filesystems
  - snapshot
- Entirely hardware-neutral solution.
  - mix and match hardware vendors

Device Mapper

- A general-purpose method for creating logical devices by mapping:
  - specified sectors on underlying devices
  - according to the rules implemented in a “target”, for example:
    - multipath, mirror, linear, striped, snapshot
- dm devices can be stacked, for example:
  - snapshot of a mirror whose components are multipath devices
- this is the basis for multipath and LVM

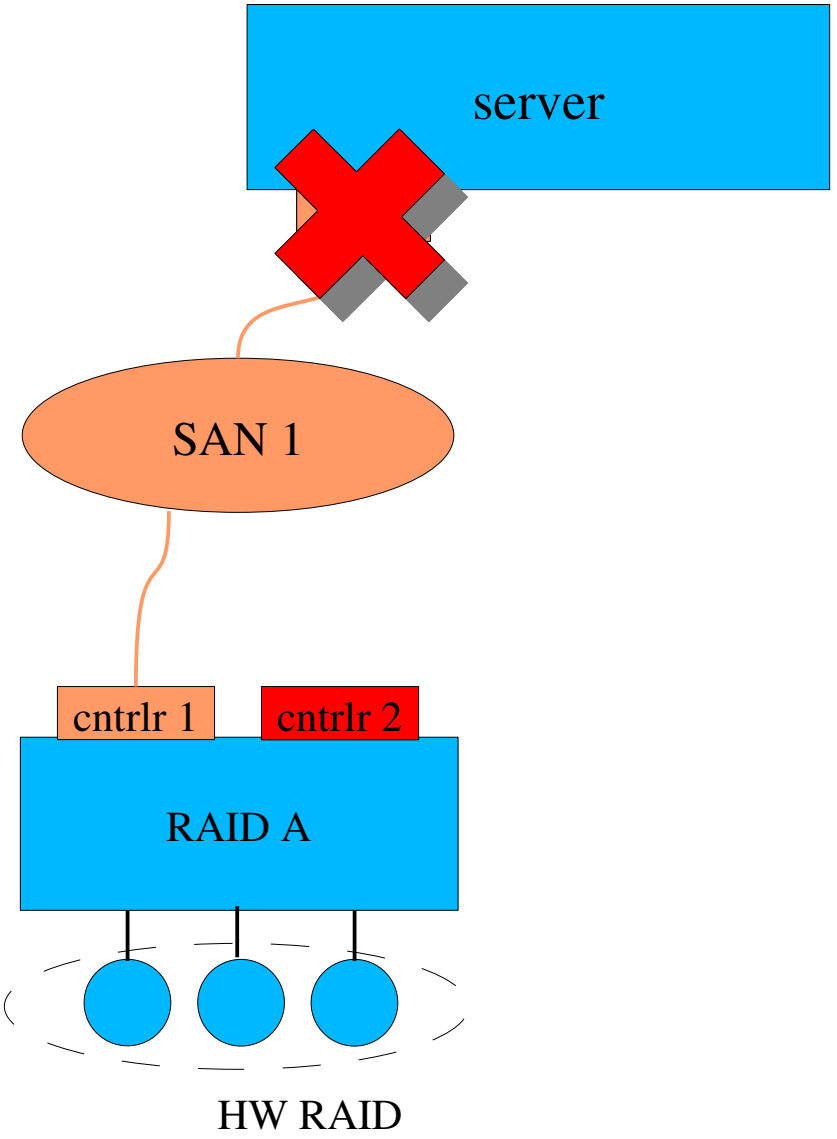
An example configuration



Path failure

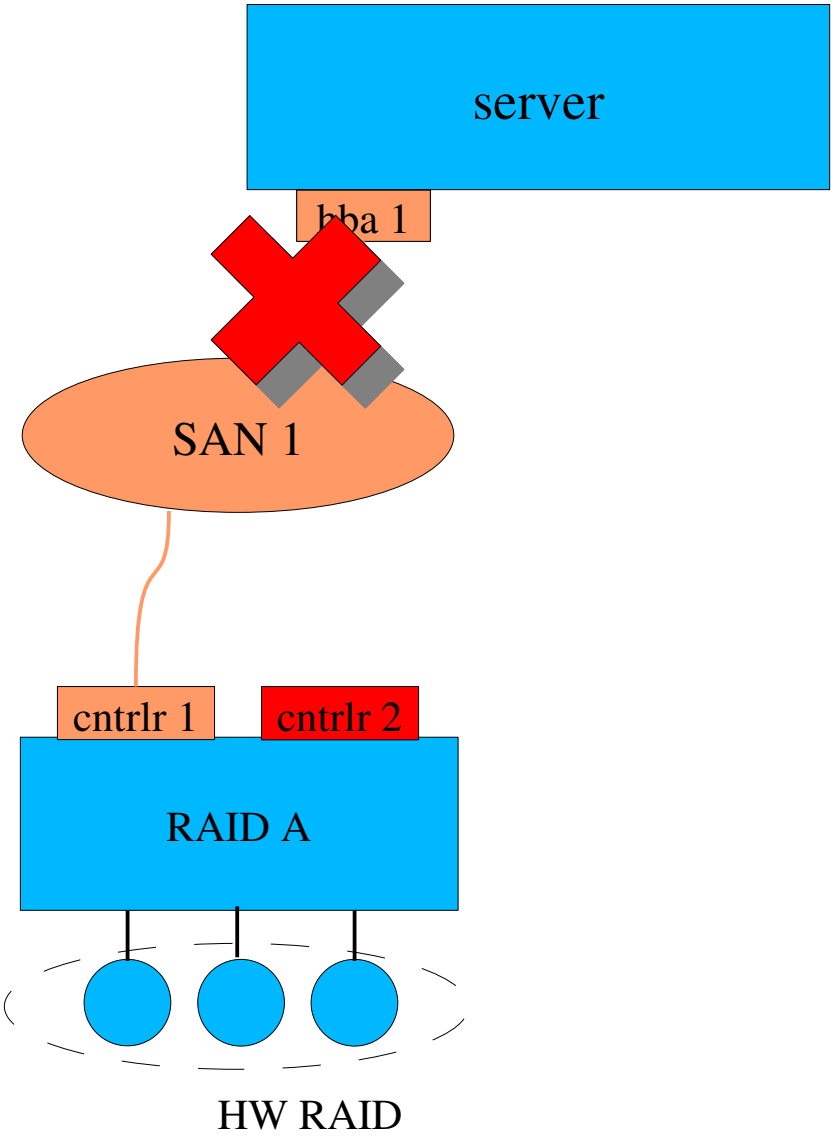
- HBA failure
- FC cable failure
- SAN Switch failure
- Array controller port failure

HBA Failure

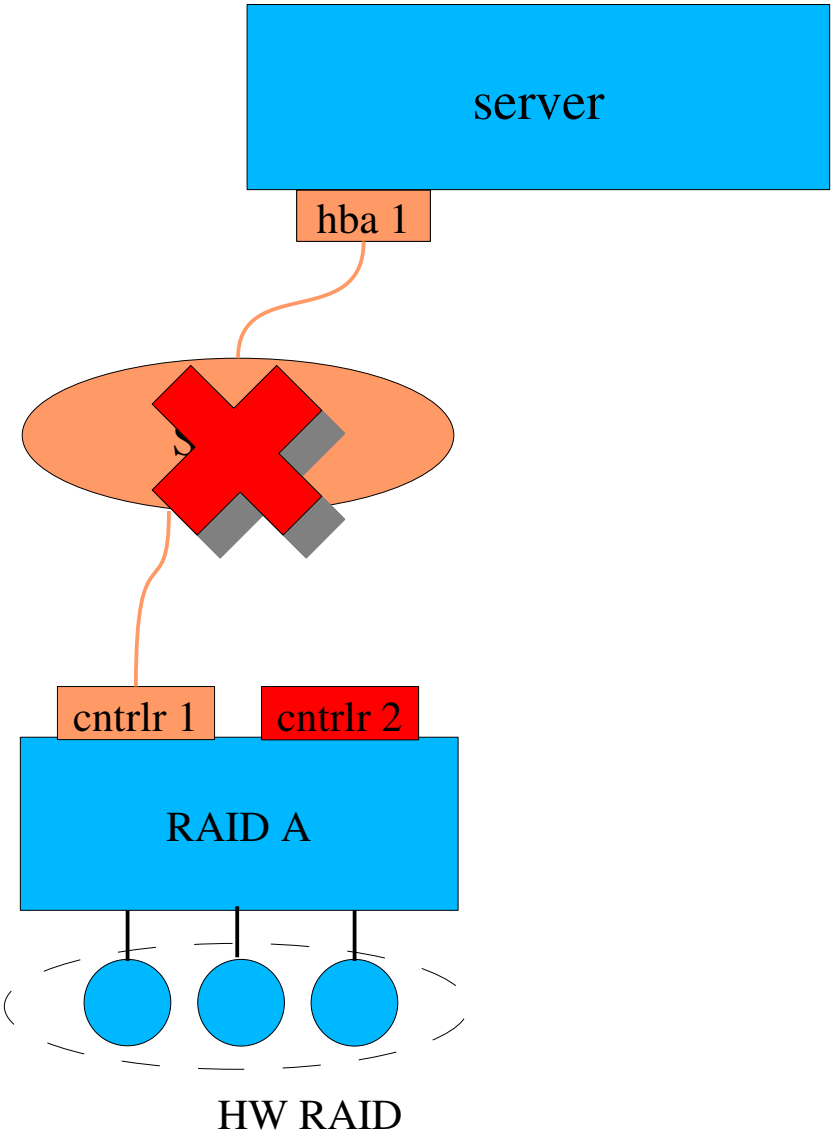

server
SAN 1
cntrlr 1
cntrlr 2
RAID A
HW RAID


RED HAT :: NASHVILLE :: 2006
SUMMIT

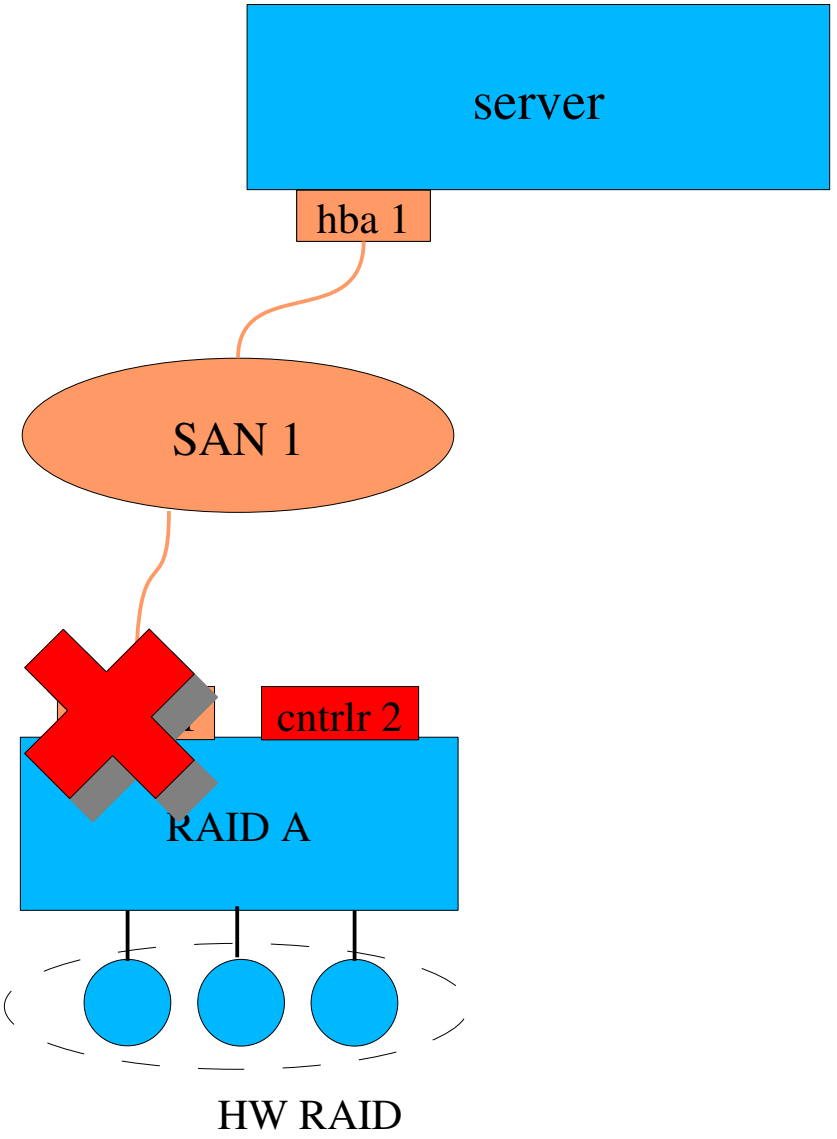
FC Cable Failure



SAN Switch Failure



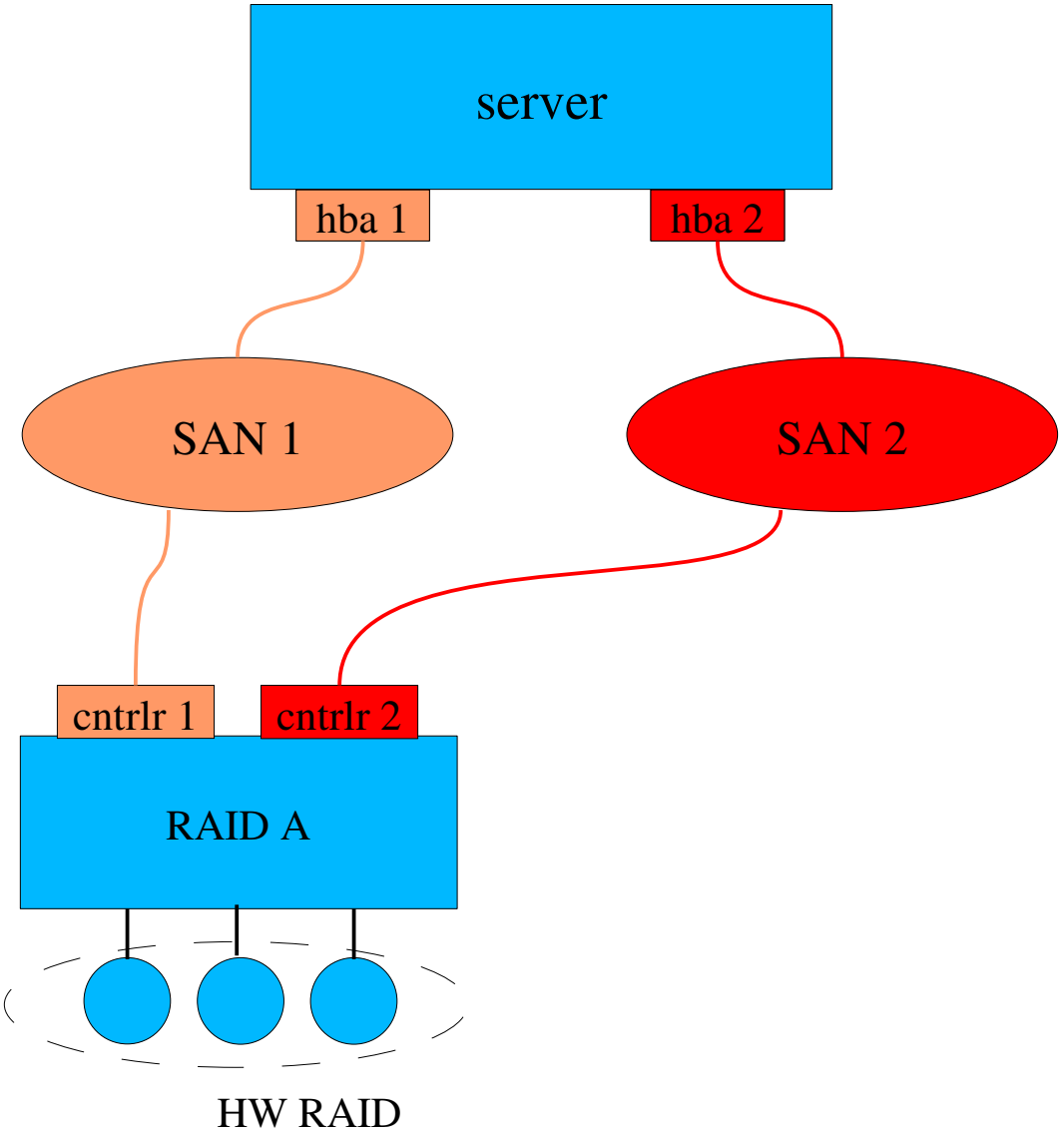
Array Controller Failure



Path failure solved: multipath

- Uses device mapper to combine single path devices into a virtual multipath unit
- Can be round-robin, or failover
- Automatic path recovery and failback
- All tunable via config and callouts

Multipath


server
hba 1
hba 2
SAN 1
SAN 2
cntrlr 1
cntrlr 2
RAID A
HW RAID


RED HAT :: NASHVILLE :: 2006
SUMMIT

What does multipath work with?

- All active/active arrays
- Active/Passive arrays
  - Requires module per array type
  - EMC Clariion

Multipath configuration

- Get Ready
  - Install device-mapper-multipath rpm
  - Comment out default blacklist
  - Start daemons, load module
- Create multipath device
  - 'multipath' is all you need

Multipath command

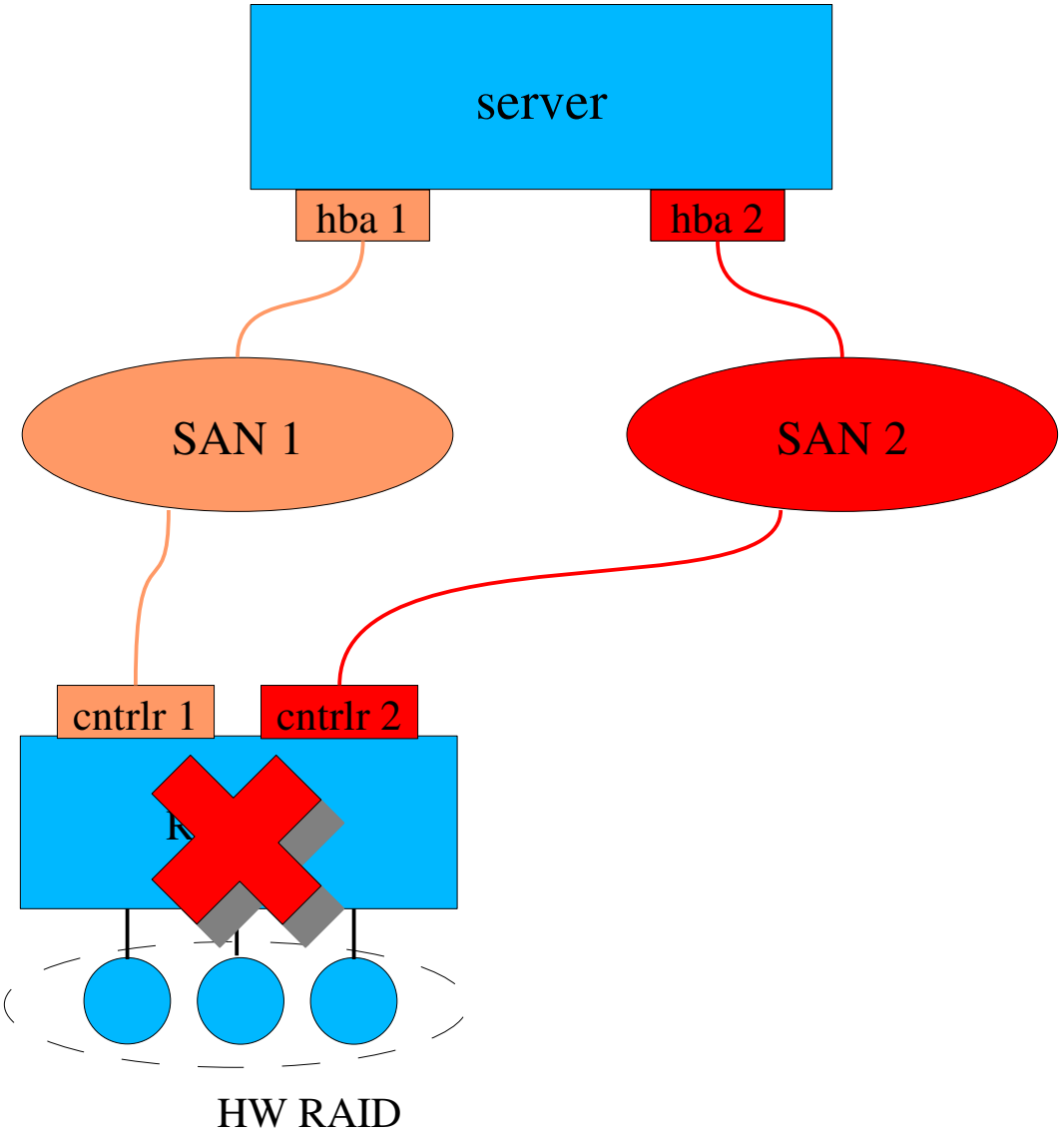
- Multipath -v2
  - Builds the maps with verbosity
- Multipath -ll
  - Shows info

A multipath map

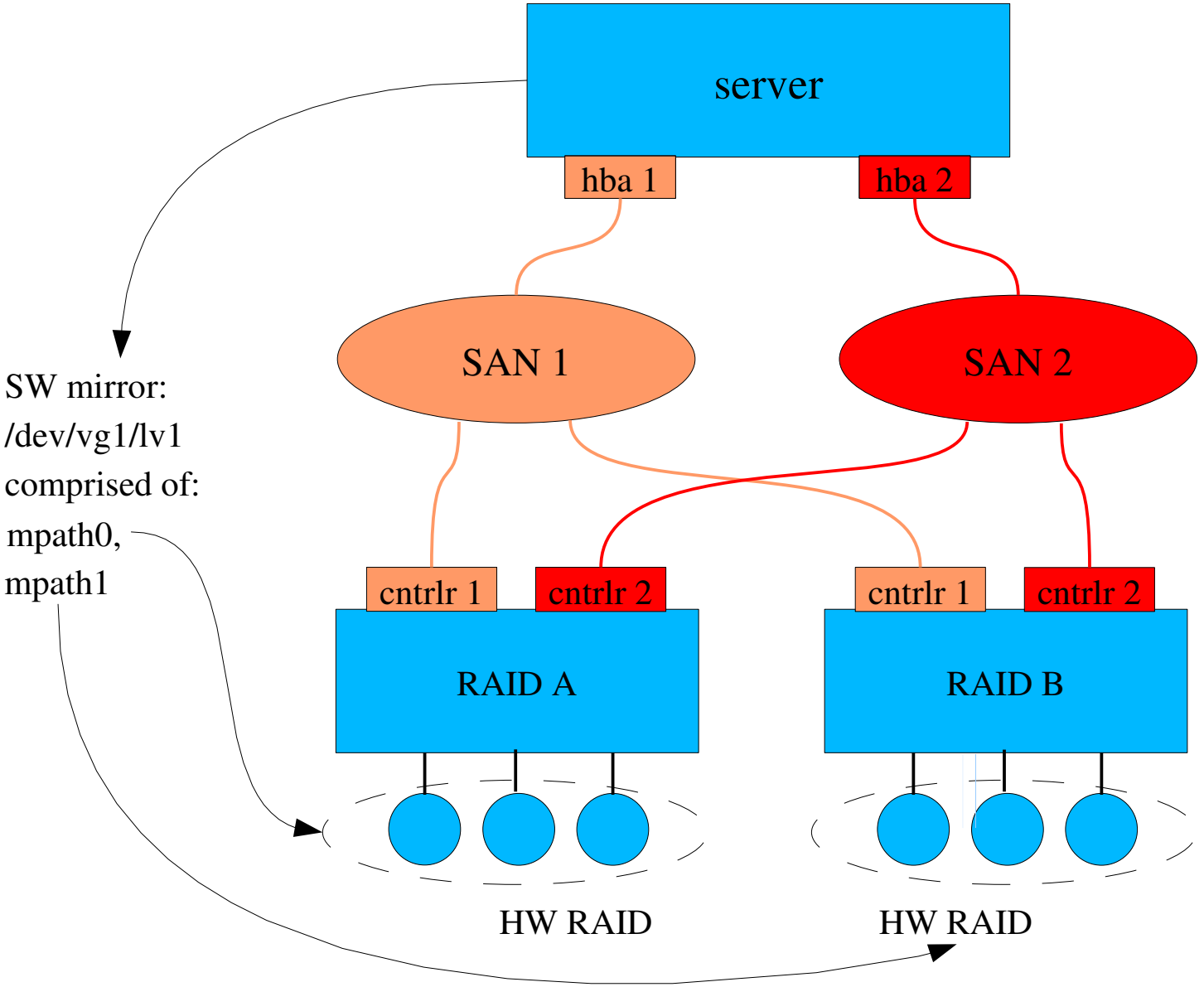
```
[root@clu1 ~]# multipath -ll mpath3
mpath3 (3600d0230003228bc000339414edb8101)
[size=58 GB][features="0"][hwhandler="0"]
\_ round-robin 0 [prio=1][active]
 \_ 2:0:0:6 sdd 8:48 [active][ready]
\_ round-robin 0 [prio=1][enabled]
 \_ 3:0:0:6 sdg 8:96 [active][ready]
```

- You can see WWID, size, bus/lun info, and single path (sd) device names.
- Example is failover, not multibus

What about array failure?

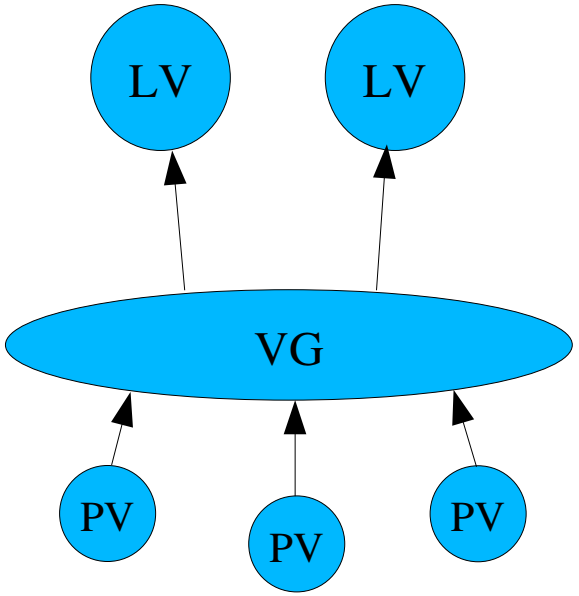


Device mapper mirroring



Logical Volume Manager

- Uses device mapper
- Combine Physical Volumes (PVs) into a storage pool, called a Volume Group (VG).
- Carve Logical Volumes (LVs) as needed from the VG.



Initialize Four PVs

```
# pvcreate /dev/mapper/mpath[0123]
  Physical volume "/dev/mapper/mpath0"
  successfully created
  . . .

# pvs
  PV         VG          Fmt  Attr PSize  PFree
  /dev/dm-2              lvm2 a-   39.06G  8.00M
  /dev/dm-3              lvm2 a-   39.06G  8.00M
  /dev/dm-4              lvm2 a-   19.53G 652.00M
  /dev/dm-5              lvm2 a-   19.53G 648.00M
```

Create Volume Group

```
# vgcreate nv_group /dev/mapper/mpath0
  /dev/mapper/mpath1 /dev/mapper/mpath2
  /dev/mapper/mpath3

 Volume group "nv_group" successfully created

# vgs
  VG          #PV #LV #SN Attr   VSize   VFree
  VolGroup00    1   2   0 wz--n-  16.84G  64.00M
  nv_group      4   0   0 wz--n- 117.17G 117.17G
```

Create SW mirror LV

- An “n” member mirror set consists of “n+1” PVs. The extra member is a log volume.
  - the log keeps track of which regions are clean, not synchronized, or have a write-in-progress
  - if there is a failure, this log makes recovery of the set faster
    - if the log fails, then the mirror set must undergo a full re-sync whenever it is recovered
- To create a 2-member mirror, with a log file:

```
# lvcreate -m1 --size 39.05GB --name nv1 nv_group
  Rounding up size to full physical extent 39.05 GB
  Logical volume "nv1" created
```

Create SW mirror LV (cont.)

- With “lvs -a” we see:
  - the two member volumes, plus the log
  - progress of the sync copy

```
#  lvs -a
  LV                VG         LSize     Log       Copy%
  nv1               nv_group 97.00G     nv1_mlog  5.03
  [nv1_mimage_0]    nv_group 97.00G
  [nv1_mimage_1]    nv_group 97.00G
  [nv1_mlog]        nv_group  4.00M
```

- Yep, there's i/o going on:

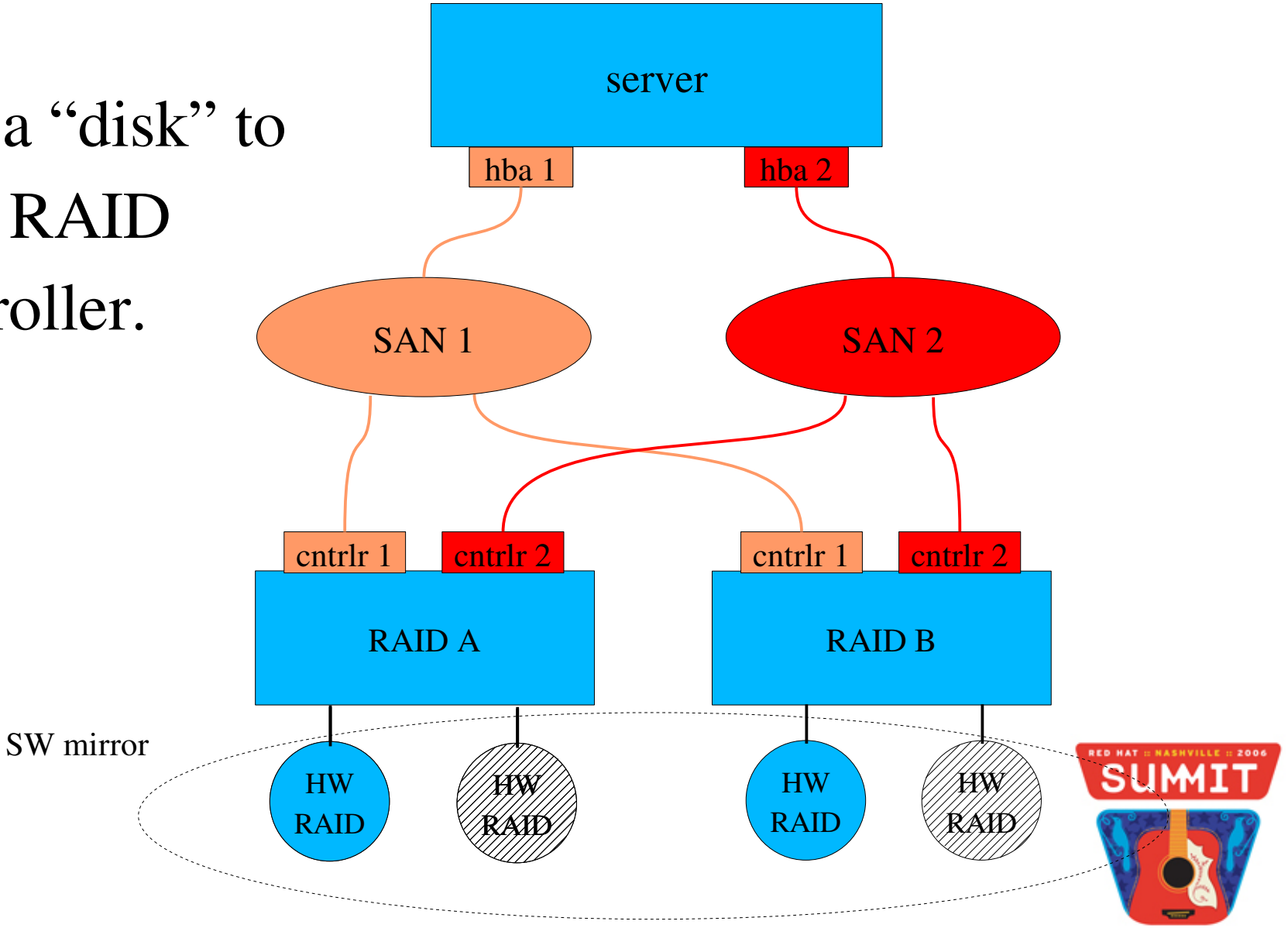
```
# vmstat 1
procs -----------memory---------- -----io----
 r  b   swpd free      buff  cache    bi    bo
 0  0      0 3655900  20336 136184 19968 20482
```

Mount SW mirror

```
# mke2fs -j /dev/nv_group/nv1
mke2fs 1.35 (28-Feb-2004)
# mount /dev/nv_group/nv1 /mnt/testnv1/
# df
Filesystem  1K-blocks Used Available Mounted on
/dev/mapper/nv_group-nv1
            40303992  81984  38174624 /mnt/testnv1
```

Outgrow the SW mirror Vol?

- Add a “disk” to each RAID controller.



Steps to add new storage:

1) Probe hba1 and add paths to new logical units

- you may need a low-level HW scan, depending on your configuration:

```
# echo 1 > /sys/class/fc_host/host1/issue_lip
```

- then cause the SCSI mid-layer to probe and add “sd” devices:

```
# echo "- - -" > /sys/class/scsi_host/host1/scan
```

- check /var/log/messages for progress:

Step 1: probe hba1 (cont.)

- Multipath gets set up automatically:

```
# tail -30 /var/log/messages

. . .

Apr  2 01:54 kernel: SCSI device sdj: 81920000
Apr  2 01:54 kernel: SCSI device sdk: 81920000
Apr  2 01:54 multipathd: sdj: path checker registered
Apr  2 01:54 multipathd: sdk: path checker registered
Apr  2 01:54 multipathd: mpath4: event checker started
Apr  2 01:54 multipathd: mpath5: event checker started
Apr  2 01:54:multipathd: mpath4: remaining active paths: 1
Apr  2 01:54 multipathd: mpath5: remaining active paths: 1
```

Step 2: Add paths through hba2:

```
# echo 1 > /sys/class/fc_host/host2/issue_lip

# echo "- - -" > /sys/class/scsi_host/host2/scan

# tail -30 /var/log/messages

. . .

Apr  2 01:54 kernel: SCSI device sdl: 81920000
Apr  2 01:54 kernel: SCSI device sdm: 81920000
Apr  2 01:54 multipathd: sdl: path checker registered
Apr  2 01:54 multipathd: sdm: path checker registered
Apr  2 01:54 multipathd: mpath4: event checker started
Apr  2 01:54 multipathd: mpath5: event checker started
Apr  2 01:54:multipathd: mpath4: remaining active paths: 2
Apr  2 01:54 multipathd: mpath5: remaining active paths: 2
```

Step 3: Prepare PVs, add to VG

- Write LVM metadata to each multipath device:

```
#pvcreate /dev/mapper/mpath4 /dev/mapper/mpath5
```

- Add the PVs to the existing VG:

```
# vgextend nv_group /dev/mapper/mpath4
  /dev/mapper/mpath5
  Volume group "nv_group" successfully extended
# vgs
  VG          #PV #LV #SN Attr   VSize   VFree
  nv_group      6   4   0 wz--n- 195.29G 117.18G
```

Step 4: Extend Mirrored LV

- Unfortunately, mirrors can not be extended while active yet.
- So, first deactivate the mirror:

```
# umount /mnt/testnv1/
# lvchange -a n  /dev/nv_group/nv1
```

- Extend mirrored LV:

```
# lvextend -L 97GB /dev/nv_group/nv1
  Extending 2 mirror images.
  Extending logical volume nv1 to 97.00 GB
```

Step 5: Enlarge filesystem

- Re-activate mirrored LV:

```
# lvchange -a y  /dev/nv_group/nv1
# mount /dev/nv_group/nv1 /mnt/testnv1/
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/nv_group-nv1
                39G   81M  37G  1% /mnt/testnv1
```

- Extend ext3 filesystem:

```
# ext2online /mnt/testnv1/
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/nv_group-nv1
                96G   92M  91G   1% /mnt/testnv1
```

LVM snapshots

- Allow you to instantly create a virtual copy of a LV.
  - As data on the original volume changes, the old data is preserved on the snapshot volume first.
    - Reads of the snapshot come from the preserved data, if present, otherwise from the original volume.
    - Writes to the snapshot are allowed.

```
# lvcreate --size 1G --snapshot --name
  nv1-snap-20060513-1122 /dev/nv_group/nv1
```

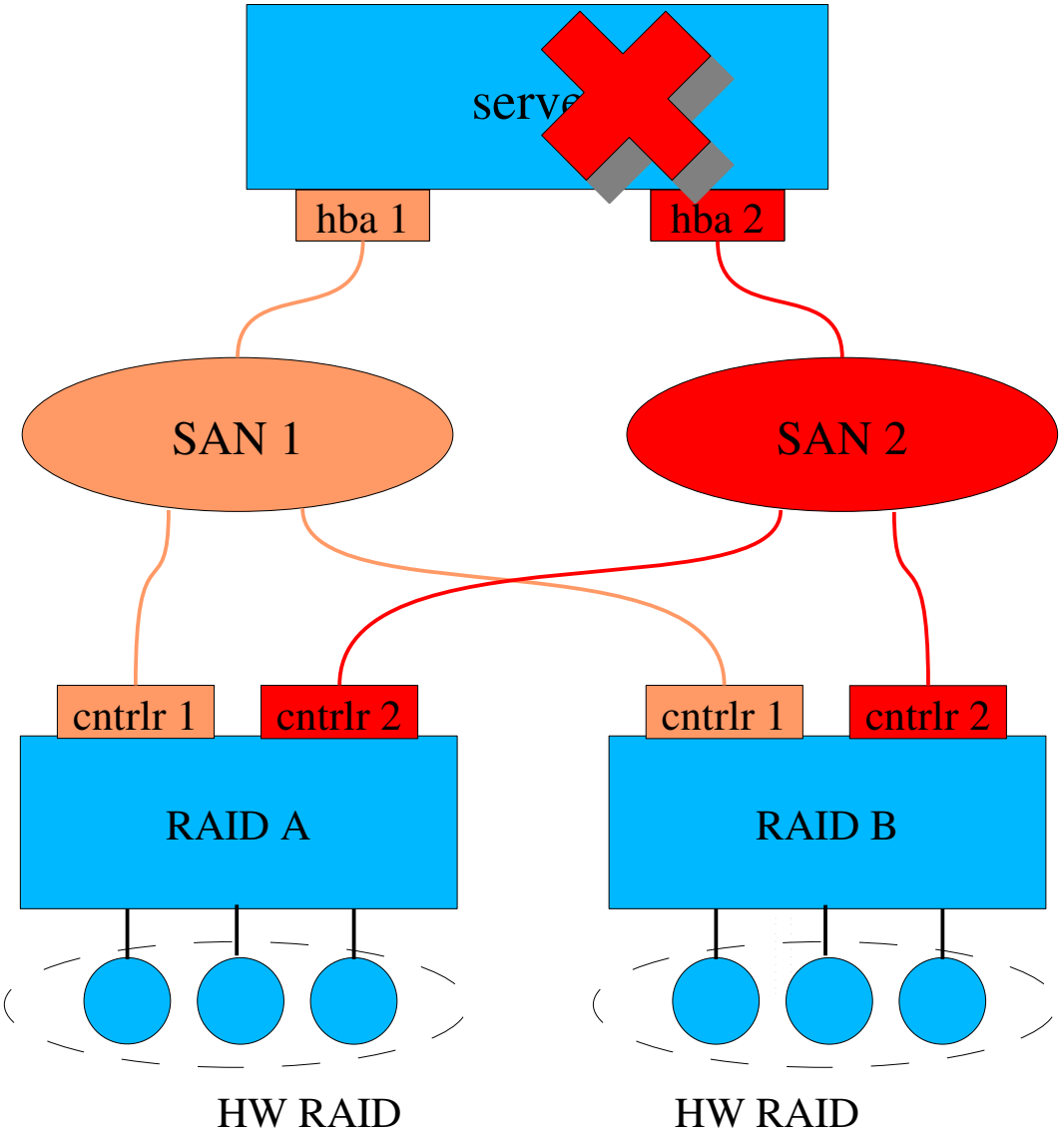
LVM snapshots (cont.)

- The snapshot only needs to be large enough to hold the changes that occur while it exists.
  - use “lvs” to monitor how full the snap is

```
# lvs
  LV       LSize  Origin Snap%  Move Log       Copy%
  nv1      97.00G                    nv1_mlog 0.00
  nv1-snap-20060513-1122
           1.00G  nv1     0.02
```

  - use lvextend to add capacity.

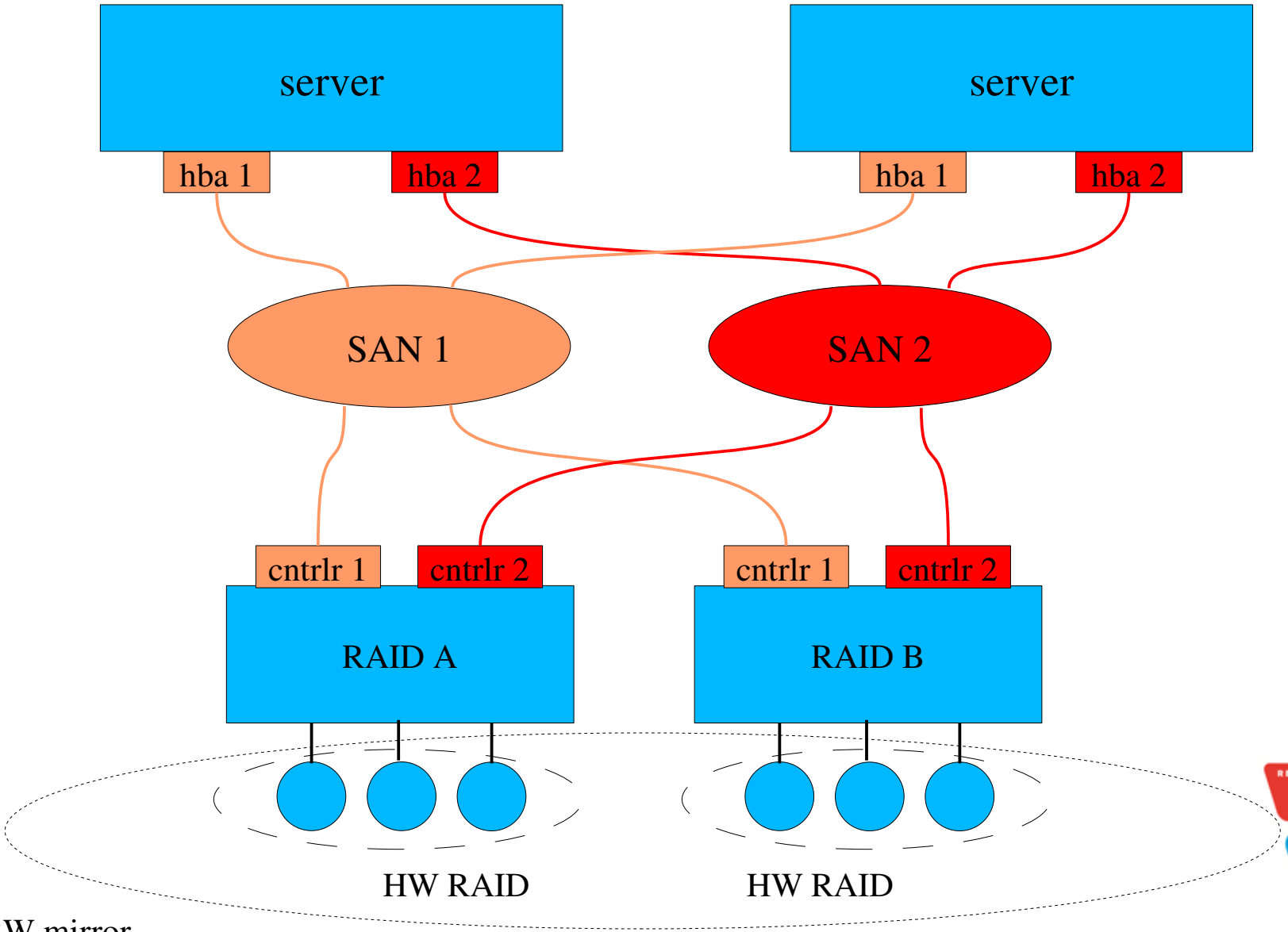
What about server failure?



GFS – the next step

- Server redundancy
  - Multiple systems can have the same LUN mounted read/write at the same time
  - GFS coordinates access across nodes
- See additional talks in the Clustering and Storage Track

GFS example


server
server
hba 1
hba 2
hba 1
hba 2
SAN 1
SAN 2
cntrlr 1
cntrlr 2
cntrlr 1
cntrlr 2
RAID A
RAID B
HW RAID
HW RAID
SW mirror

Summary

- RHEL provides the ability to build HA systems
  - fully redundant
  - repair online
  - extend online
  - backup online
- All components integrated in the o.s.
- No hardware lock-in

More Information

- Available at http://people.redhat.com/nayfield
  - This presentation
  - Enterprise Storage Quickstart w/ multipath
  - Enterprise GFS Quickstart
  - Enterprise Storage Quickstart w/ mirroring