TA48

Advanced VMFS Configuration and Troubleshooting

Mostafa Khalil, VCP

Staff Engineer, Product Support Engineering Team

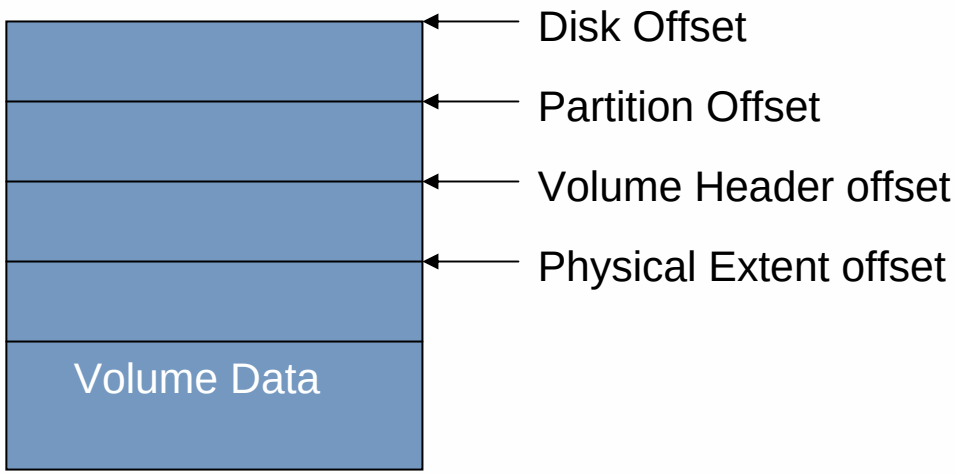
VMware

Agenda

- **VMFS2 layout (public version)**
- **VMFS3 layout (public version)**
- **VMFS3 and Snapshot/Replica LUNs**
- **Backing up VMFS3 metadata**
- **Distributed Lock handling**
- **SCSI Reservations**
- **Restoring VMFS2 and 3 partition table**
- **Aligning VMFS3 partition**

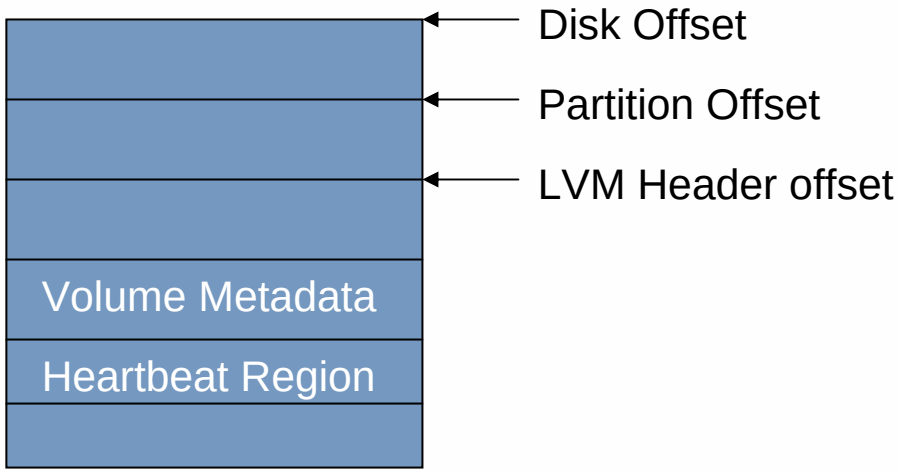
VMFS2 Layout

- **Simplified Layout (not to scale)**



VMFS3 Layout

- **Simplified Layout (not to scale)**



LVM (Logical Volume Manager)

- Volume ID (Signature)
- Number of Extents
- Number of devices
- Volume size
- Snapshot?
- Exists on all devices comprising the volume
- Volume Label is on the Volume Metadata (not here)

Handling Snapshots

- **Why are VMFS Volumes seen as snapshots when they are not?**
  - ESX server A is presented with a LUN on ID 0.
  - Same LUN is presented to ESX server B on ID 1.
  - VMFS-3 volume created on LUN ID 0 from server A.
  - Volume on server B will **not** be mounted when SAN is rescanned.
  - Server B will state that the volume is a snapshot.
- **LUNs must be presented as the same IDs on all hosts. Why?**
  - When a VMFS-3 volume is created, the **SCSI Disk ID** data from the LUN/storage array is stored in the volume's LVM header.
  - When another ESX server finds a LUN with a VMFS-3 filesystem, the **SCSI Disk ID** information returned from the LUN/storage array is compared with the LVM header metadata.
  - The VMkernel treats a volume as a **snapshot** if there is a mismatch in this information.

How does this mismatched information get reported?

```
LVM: 5739: Device vmhba2:2:2:1 is a snapshot:
LVM: 5745:   disk ID: <type 3, len 15, lun 2, devType 0, scsi 3, h(id) 17714234126755338790>
LVM: 5747:   m/d disk ID: <type 3, len 15, lun 2, devType 0, scsi 3, h(id) 9219142619163180480>
LVM: 5739: Device vmhba2:2:2:1 is a snapshot:
LVM: 5745:   disk ID: <type 3, len 15, lun 2, devType 0, scsi 3, h(id) 17714234126755338790>
LVM: 5747:   m/d disk ID: <type 3, len 15, lun 2, devType 0, scsi 3, h(id) 9219142619163180480>
ALERT: LVM: 4903: vmhba2:2:2:1 may be snapshot: disabling access. See resignaturing section in SAN
config guide.
.
.
.
LVM: 5739: Device vmhba2:2:9:1 is a snapshot:
LVM: 5745:   disk ID: <type 3, len 15, lun 9, devType 0, scsi 3, h(id) 17714234126755338790>
LVM: 5747:   m/d disk ID: <type 3, len 15, lun 9, devType 0, scsi 3, h(id) 9219142619163180480>
LVM: 5739: Device vmhba2:2:9:1 is a snapshot:
LVM: 5745:   disk ID: <type 3, len 15, lun 9, devType 0, scsi 3, h(id) 17714234126755338790>
LVM: 5747:   m/d disk ID: <type 3, len 15, lun 9, devType 0, scsi 3, h(id) 9219142619163180480>
ALERT: LVM: 4903: vmhba2:2:9:1 may be snapshot: disabling access. See resignaturing section in SAN
config guide.
```

- This logging appears in the **/var/log/vmkernel** log file.
- The line containing **m/d** is the metadata.
- In this case it is the h(id) data in the LVM header which is mismatched.

Handling Snapshots

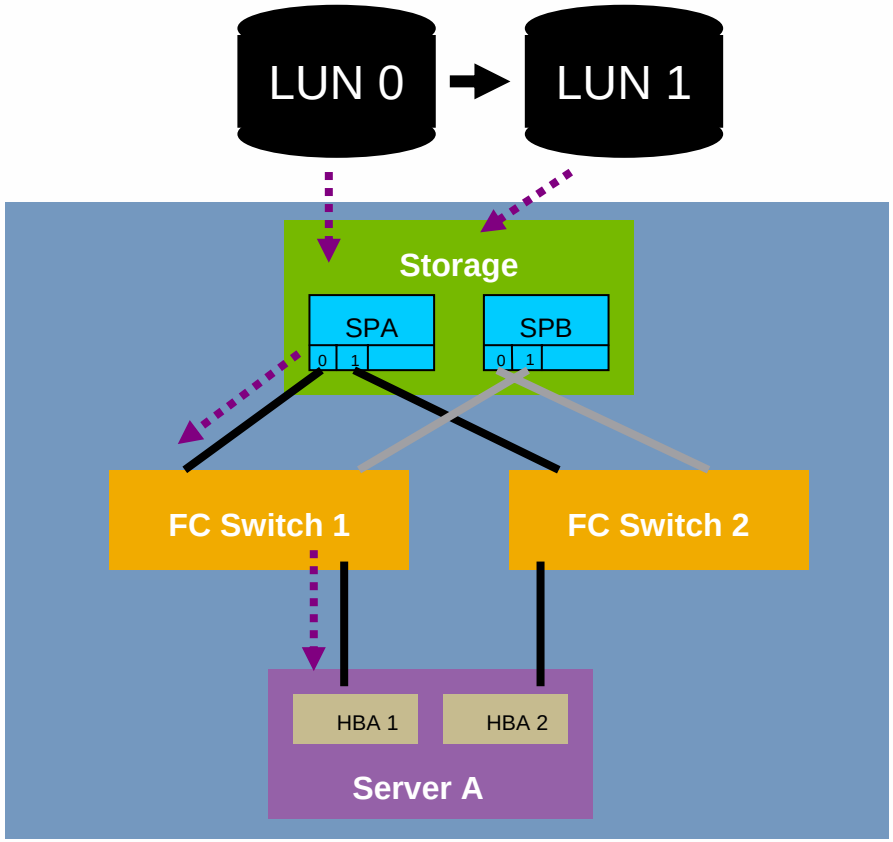
- **First of all, is it really a snapshot?**
  - If it is a mismatch of LUN IDs across different ESX hosts, fix the LUN ID through array management software to ensure that the same LUN ID is presented to all hosts for a share volume.
  - Other reasons a volume might appear as a snapshot could be changes in the way the LUN is presented to the ESX (e.g. HDS Host Mode, EMC SPC-2 director flag).
- **If it is a snapshot, you can:**
  - Turn on **LVM.EnableResignature**
- **Or**
  - Turn off **LVM.DisallowSnapshotLun**

LVM.EnableResignature

- Mounting Original and Snapshot VMFS Volumes to same ESX
  - Mount both the original and snapshot VMFS on the same ESX by setting **LVM.EnableResignature** to **1**.
  - This updates the volume with:
    - new **SCSI Disk ID** information.
    - a new VMFS-3 **UUID.**
    - a new **label**.
      - Label format will be **snap-<generation number>-<label>**, or **snap-<generation number>-<uuid>** if there is no label, e.g.
        - Before resignature: **/vmfs/volumes/lun2**
        - After resignature: **/vmfs/volumes/snap-00000008-lun2**
  - Any virtual machines on this VMFS have to be manually registered.

LVM.EnableResignature

-- snapshot --



LVM.EnableResignature will have to be used to make the volume located on cloned LUN, LUN 1, visible to the same ESX server after a rescan.

2 volumes with the same UUID must not be presented to the same ESX server. Issues with data integrity may occur.

Enable Resignaturing & Rescan

```
[root@esx LVM]# pwd
/proc/vmware/config/LVM

[root@esx LVM]# cat EnableResignature
EnableResignature (Enable Volume Resignaturing) [0-1: default = 0]: 0

[root@esx LVM]# echo 1 > EnableResignature

[root@cork LVM]# esxcfg-rescan vmhba2
Rescanning vmhba2...done.
On scsi6, removing: 0:0 0:1 0:2 0:3 2:1 2:10 2:11 2:12 2:13 2:14 2:15 2:16
2:17 2:18 2:19 2:2 2:20 2:21 2:22 2:23 2:24 2:240 2:241 2:25 2:26 2:27 2:28
2:29 2:3 2:30 2:31 2:32 2:33 2:34 2:35 2:36 2:37 2:38 2:39 2:4 2:40 2:41 2:42
2:48 2:49 2:5 2:6 2:7 2:8 2:9.
On scsi6, adding: 0:0 0:1 0:2 0:3 2:1 2:10 2:11 2:12 2:13 2:14 2:15 2:16 2:17
2:18 2:19 2:2 2:20 2:21 2:22 2:23 2:24 2:240 2:241 2:25 2:26 2:27 2:28 2:29
2:3 2:30 2:31 2:32 2:33 2:34 2:35 2:36 2:37 2:38 2:39 2:4 2:40 2:41 2:42 2:48
2:49 2:5 2:6 2:7 2:8 2:9.

[root@cork LVM]# echo 0 > EnableResignature
```

Remote/Target side messages during Resignaturing

```
LVM: 5739: Device vmhba2:2:2:1 is a snapshot:
LVM: 5745:   disk ID: <type 3, len 15, lun 2, devType 0, scsi 3, h(id) 17714234126755338790>
LVM: 5747:   m/d disk ID: <type 3, len 15, lun 2, devType 0, scsi 3, h(id) 9219142619163180480>
LVM: 5739: Device vmhba2:2:2:1 is a snapshot:
LVM: 5745:   disk ID: <type 3, len 15, lun 2, devType 0, scsi 3, h(id) 17714234126755338790>
LVM: 5747:   m/d disk ID: <type 3, len 15, lun 2, devType 0, scsi 3, h(id) 9219142619163180480>
LVM: 6031: Device vmhba2:2:2:1 unsnapped
.
.
.
.
LVM: 5739: Device vmhba2:2:9:1 is a snapshot:
LVM: 5745:   disk ID: <type 3, len 15, lun 9, devType 0, scsi 3, h(id) 17714234126755338790>
LVM: 5747:   m/d disk ID: <type 3, len 15, lun 9, devType 0, scsi 3, h(id) 9219142619163180480>
LVM: 5739: Device vmhba2:2:9:1 is a snapshot:
LVM: 5745:   disk ID: <type 3, len 15, lun 9, devType 0, scsi 3, h(id) 17714234126755338790>
LVM: 5747:   m/d disk ID: <type 3, len 15, lun 9, devType 0, scsi 3, h(id) 9219142619163180480>
LVM: 6031: Device vmhba2:2:9:1 unsnapped

LVM: 3211: Snapshot LV <snap-1b47edf4-45b60b75-157d9671-088a-000423c5> complete on device vmhba2:2:9:1
Vol3: 586: Begin resignaturing volume label: lun2, uuid: 45dd7f39-e69dd091-c5f4-000423c5a2ec
Vol3: 625: End resignaturing volume label: snap-00000008-lun2, uuid: 45dd7563-b3f744de-12e0-000423c5a56c
```

LVM.DisallowSnapshotLUN

- Allow Snapshot VMFS-3 Volumes to be seen by a different ESX
  - You can present a snapshot VMFS-3 volumes to a different ESX Server host using 2 options:

    EnableResignature or **DisallowSnapshotLUN**.
  - To allow snapshot LUNs, set:

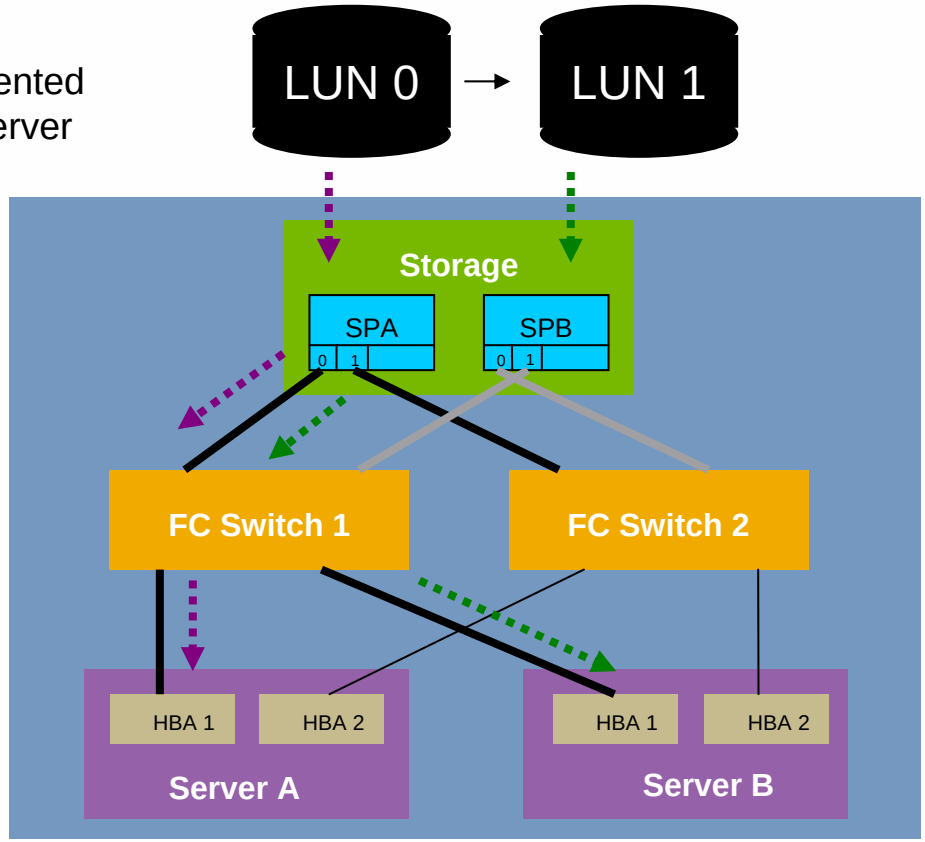
    EnableResignature to **0** (Disable)
  - &

    DisallowSnapshotLUN to **0** (Disable)
  - Do not use **DisallowSnapshotLUN** to present snapshots back to same ESX server as the original LUN.
  - *Unpredictable* results can occur since you will have two VMFS-3 volumes with the same UUID, e.g. data corruption!
  - **LVM.EnableResignature** overrides **LVM.DisallowSnapshotLUN**

LVM.EnableResignature OR LVM.DisallowSnapshotLUN

-- snapshot --

Snapshot LUN presented to a different ESX server

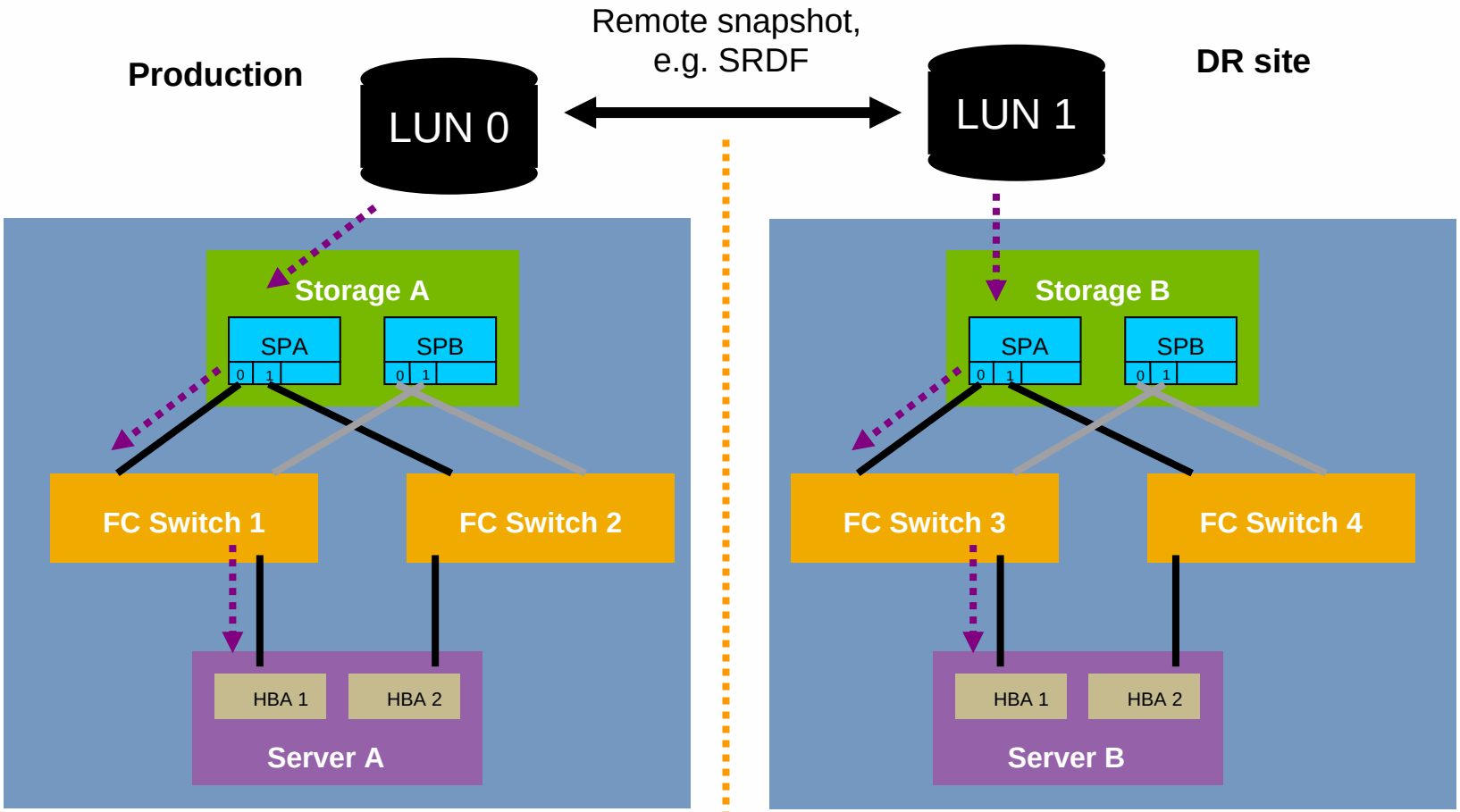


We can present the snapshot LUN, LUN 1, using **DisallowSnapshotLUN = 0** on Server B as long as Server B cannot see LUN 0.

Volume name is unchanged
No need to re-register VMs

If Server B can also see LUN 0, then we must use **resignaturing** since we cannot present two Volumes with the same UUID (Signature) to the same ESX server.

LVM.DisallowSnapshotLUN



Since there is not going to be a LUN with the same UUID at the remote site, one can **allow snapshots**.

How about RDMs?

- **RDMs mapped RAW LUNs can be replicated to the Remote Site**
- **RDMs reference the RAW LUNs via**
  - the LUN number
  - LUN ID
- **VMFS3 Volumes on Remote site will have unusable RDM configuration if either properties change**
- **Remove the old RDMs and recreate them**
  - Better to create a dedicate VMFS3 volume (on each site) to store RDM entries and not replicate it
  - Must correlate RDM entries to correct RAW LUNs
  - Use the same RDM file name as old one to avoid editing the vmx file

Manually backing up VMFS2 volume's Metadata

- **Where is the metadata**
  - Within first 1MB of LUN
- **Locate LUN**

```
vmkfstools -P /vmfs/VMworld
/vmfs/VMworld is a VMFS-2.52 volume spanning 1 physical extents.
Volume label (if any): VMworld
UUID (if any): 4366b49b-7b62c62a-648c-001125298524
Physical Extents:
        vmhba0:0:2:1  ← This is the LUN/Partition number
```

- **Locate SCSI Device**

```
vmkpcidivy –q vmhba_devs
vmhba0:0:1 /dev/sdc
vmhba0:0:2 /dev/sdd  ← This is the device name
```

Manually backing up VMFS2 volume's Metadata

- **Dump Metadata Blocks**

```
dd if=/dev/sdd of=vmworld.bin bs=1024 count=1024
1024+0 records in
1024+0 records out
```

- **Includes Partition Table**

```
fdisk -lu vmworld.bin
You must set heads sectors and cylinders.
You can do this from the extra functions menu.

Disk vmworld.bin: 0 heads, 0 sectors, 0 cylinders ← ignore this line
Units = sectors of 1 * 512 bytes

      Device Boot    Start       End    Blocks   Id  System
vmworld.bin1   *          63 571383854 285691896   fb  Unknown
```

Manually backing up VMFS3 volume's Metadata

- Where is the metadata
  - **Within first 20MB of LUN**
  - **Volume's system files (*.sf) located at volume's root**

```
ls -al /vmfs/volumes/VMworld/
total 574464
drwxrwxrwt    1 root     root         980 Jul 30 18:47 .
drwxrwxrwx    1 root     root         512 Jul 30 19:37 ..
-r--------    1 root     root      229376 Jul 30 18:47 .fbb.sf
-r--------    1 root     root    64946176 Jul 30 18:47 .fdc.sf
-r--------    1 root     root   255655936 Jul 30 18:47 .pbc.sf
-r--------    1 root     root   260366336 Jul 30 18:47 .sbc.sf
-r--------    1 root     root     4194304 Jul 30 18:47 .vh.sf
```

Manually backing up VMFS3 volume's Metadata

- **How to backup system files**

```
cp /vmfs/volumes/VMworld/.*.sf /tmp/backup  (notice the leading dot)
```

- **How to backup Metadata blocks**
  - Locate LUN

```
vmkfstools -P /vmfs/volumes/VMworld
VMFS-3.21 file system spanning 1 partitions.
File system label (if any): VMworld
<snip>
Partitions spanned:
        vmhba0:0:1:1   ← This is the LUN/Partition number
```

  - Locate SCSI device name

```
esxcfg-vmhbadevs -q
vmhba0:0:0 /dev/sdc
vmhba0:0:1 /dev/sdd  ← This is the device name
```

Manually backing up VMFS3 volume's Metadata

- **Dump the blocks (20MB) from the beginning of LUN**

```
dd if=/dev/sdd of=vmworld.bin bs=1024 count=20480
20480+0 records in
20480+0 records out
```

- **Includes partition table**

```
fdisk -lu vmworld.bin
You must set cylinders.
You can do this from the extra functions menu.
<snip>
Units = sectors of 1 * 512 = 512 bytes

      Device Boot    Start       End    Blocks   Id  System
vmworld.bin1           128  20971519  10485696   fb  Unknown
```

Manually backing up VMFS3 volume's Metadata

- **What to send to VMware support for data recovery (just in case)**
  - A recent vm-support dump from the host
  - Current vm-support dump
  - Backup copy of system files
  - dd dump of first 20MB of Disk

Distributed Lock handling by VMFS3

- Done in-band
- Hosts mount a VMFS3 volume
- Hosts' ids posted to heartbeat region
- Heartbeat records are updated at regular intervals by hosts
- Host X locks a file, the lock is associated with its ID
- If host X dies or loses access to volume the file lock is stale
- Host Z attempts to lock the same file which is locked
- Host Z check the heartbeat record of Host X (~5 times)
- If host X heartbeat record is not updated, Host Z will age the lock
- All other hosts yield to host Z and not attempt to lock the file
- Lock is broken and Host Z acquires the lock
- Journal is replayed by Host Z

SCSI Reservations

- **Non-persistent (not PGR)**
- **Only when metadata gets modified**
  - Creating volumes
  - Creating files
  - Locking and breaking locks
  - Deleting files
  - Growing files
  - VMotion
  - Resignaturing
- **Releasing a SCSI Reservation on a LUN**
  - Done by resetting the LUN using vmkfstools

```
vmkfstools -L lunreset /vmfs/devices/disks/vmhba1\:6\:1\:0
```

Restoring partition table for VMFS2

- **Identify the SCSI device name for the affected LUN**

```
vmkpcidivy -q vmhba_devs
vmhba0:0:0 /dev/sda
vmhba0:0:1 /dev/sdb
vmhba0:0:2 /dev/sdc  ← This LUN
vmhba0:0:3 /dev/sdd
```

- **In this example the affected LUN is vmhba0:0:2 which is mapped to /dev/sdc device**

Restoring partition table for VMFS2

- **Identify the VMFS2 Volume Header offset**

```
hexdump /dev/sdc |less
00001f0 0000 0000 0000 0000 0000 0000 0000 aa55
0000200 0000 0000 0000 0000 0000 0000 0000 0000
*
0017e00 f15e 2fab 0002 0000 b49b 4366 c62a 7b62  ← Volume Header
0017e10 648c 1100 2925 2485 0200 0000 9ff0 220e
0017e20 0000 0000 0000 0020 20e5 0002 0000 0000
0017e30 0100 0000 0800 0000 0200 0000 1000 0000
```

- **Look for the value f15e 2fab which marks the Volume Header Offset**
- **In this example, the offset is at 0017e00**

Restoring partition table for VMFS2

- **Convert the offset value to decimal**
  - 17e00 = 97792
- **Convert bytes to blocks (divide by 512 which is the device block size**
  - 97792 / 512 = 191 blocks
- **The partition offset should be 64KB (128 blocks) prior to this offset**
  - 191 – 128 = 63
- **This partition offset starts at block number 63**

Restoring partition table for VMFS2

- **Create the partition using offset 63**

```
fdisk -u /dev/sdc
        n (to create a new partition)
        p (to create a primary partition)
        1 (to create the 1st partition)
        63 to set the partition offset [enter]
        [enter] to keep the default value for the number of blocks
        t (to change the type of partition)
        fb (to set the partition as VMFS)
        w (to save)
vmkfstools -V (to discover the VMFS)
```

- **The volume should be available at /vmfs mount point**

Restoring partition table for VMFS3

- **Identify the SCSI device name for the affected LUN**

```
esxcfg-vmhbadevs -q
vmhba0:0:0 /dev/sdc
vmhba0:0:1 /dev/sdd
vmhba0:0:2 /dev/sde  ← This LUN
vmhba0:0:3 /dev/sdf
vmhba0:0:4 /dev/sdh
```

- **In this example the affected LUN is vmhba0:0:2 which is mapped to /dev/sde device**

Restoring partition table for VMFS3

- **Identify the VMFS2 LVM Header offset**

```
hexdump /dev/sde |less
00001f0 0000 0000 0000 0000 0000 0000 0000 aa55
0000200 0000 0000 0000 0000 0000 0000 0000 0000
*
0110000 d00d c001 0003 0000 0010 0000 1602 0006  ← LVM Header
0110010 0400 4249 204d 2020 2020 3731 3232 362d
0110020 3030 2020 2020 2020 2020 3530 3032 0a60
0110030 800b 1700 844e 0000 8712 0744 df1c 3731
```

- **Look for the value d00d c001 which marks the LVM Header Offset**
- **In this example, the offset is at 0110000**

Restoring partition table for VMFS3

- **Convert the offset value to decimal**
  - 0110000 = 1114112
- **Convert bytes to blocks (divide by 512 which is the device block size**
  - 1114112 / 512 = 2176 blocks
- **The partition offset should be 1MB (2048 blocks) prior to this offset**
  - 2176 – 2048 = 128
- **This partition offset starts at block number 128**

Restoring partition table for VMFS3

- **Create the partition using offset 128**

```
fdisk -u /dev/sdc
        n (to create a new partition)
        p (to create a primary partition)
        1 (to create the 1st partition)
        128 to set the partition offset (first sector) [enter]
        [enter] to keep the default value for the last sector
        t (to change the type of partition)
        fb (to set the partition as VMFS)
        w (to save)
vmkfstools -V (to discover the VMFS)
```

- **The volume should be available at /vmfs mount point**

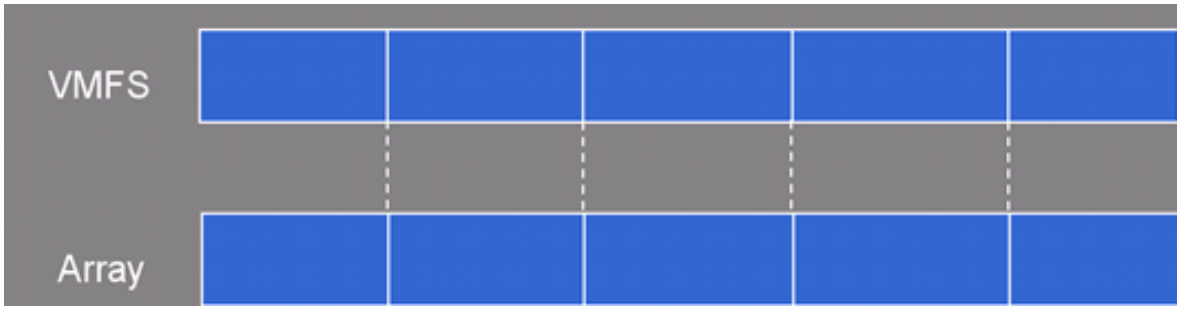
VMFS3 partition alignment

- **Why**
  - VMFS partitions that align to 64KB track boundaries result in reduced latency and increased throughput (based on tests on EMC CLARiiON CX series)
  - Partition alignment on both physical machines and VMware VMFS partitions prevents performance I/O degradation due to track misalignment.
  - Creating VMFS partitions using VirtualCenter 2.0 or later results in a partition table aligned on the 64KB boundary as storage and operating system vendors recommend.

VMFS Partition Alignment

VMFS volume is not aligned

VMFS Partition Alignment



VMFS Volume is aligned

VMFS Partition Alignment

- **Manual Alignment using fdisk**
- **To be done on new volumes only. Existing misaligned LUNs cannot be modified**
- **The following assumes the new LUN is vmhba0:0:6 and the device name is /dev/sdf**

```
fdisk -u /dev/sdf
    n (to create a new partition)
    p (to create a primary partition)
    1 (to create the 1st partition)
    128 to set the partition offset (first sector) [enter]
    [enter] to keep the default value for the last sector
    t (to change the type of partition)
    fb (to set the partition as VMFS)
    w (to save)
```

VMFS Partition Alignment

- **Create the VMFS3 volume on the newly created partition**

```
vmkfstools -C vmfs3 -b 1m -S VMworld /vmfs/devices/disks/vmhba0\:0\:6\:1
Creating file system on "vmhba0:0:6:1" with blockSize 1048576 and volume label
"VMworld".
Successfully created new volume: 46ae9494-14d85418-90a2-001125298524
```

- **Verify partition information**

```
fdisk -lu /dev/sdf

Disk /dev/sdf: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdf1           128  20971519  10485696   fb  Unknown
```

VMFS Partition Alignment

- **Verify Volume details**

```
vmkfstools -P -h /vmfs/volumes/VMworld
VMFS-3.21 file system spanning 1 partitions.
File system label (if any): VMworld
Mode: public
Capacity 9.8G, 9.1G available, file block size 1.0M
UUID: 46ae9494-14d85418-90a2-001125298524
Partitions spanned:
        vmhba1:6:1:1
```

Questions?

TA48

Advanced VMFS Configuration and Troubleshooting

Mostafa Khalil, VCP

VMware Product Support Engineering

For more information…

http://www.vmware.com

VMWORLD 2007

EMBRACING YOUR VIRTUAL WORLD

BREAKOUT SESSION