OVIRT 4K

TEACHING AN OLD DOG NEW TRICKS

Vojtech Juranek
Senior Software Engineer
vjuranek@redhat.com

Nir Soffer
Principal Software Engineer
nsoffer@redhat.com

Red Hat

AGENDA

Why 4K?

Challenges

Detecting block size

Using block size in vdsm

Managing hosts

Troubleshooting

Demo

WHY 4K?

WHY 4K?

RHHI

HyperConverge + Hosted engine + Gluster + VDO/4K

(Creating simplicity is complex)

WHY 4K?

VDO

Did you ever feel like you have too much storage?

Using sector size of 4k instead of 512 bytes emulation may improve performance.

WHY 4K?

Support disks with sector size of 4k

Users owning 4k disks are not
happy when they cannot use them.

CHALLENGES

4K CHALLENGES

Storage format assumes
512 bytes block size

4K

CHALLENGES

~~Storage format assumes 512 bytes block size~~

Storage format V5

4K CHALLENGES

Sanlock cannot detect block size with file storage

CHALLENGES

~~Sanlock cannot detect block size with file storage~~

Sanlock 4K API

4K

CHALLENGES

```
sanlock.write_lockspace(
    "my-lockspace",
    "/path/to/lockspace",
    align=1048576,
    sector=4096)
```

CHALLENGES

VDSM uses hard-coded block size everywhere

4K CHALLENGES

~~VDSM uses hard-coded block size everywhere~~

Moving to bytes

CHALLENGES

```
def setCapacity(self, capacity):
    """
    Sets volume capacity in bytes.

    Arguments:
        capacity (int) - new capacity value in bytes.
    """
    self.setMetaParam(sc.CAPACITY, capacity)
```

4K CHALLENGES

There is no API for detecting block size on file storage

4K CHALLENGES

~~There is no API for detecting block size on file storage~~

Detect block size by accessing storage

4K CHALLENGES

(more on this later)

CHALLENGES

Poor tests in vdsm storage

CHALLENGES

~~Poor tests in vdsm storage~~

Testing real storage domains and volumes

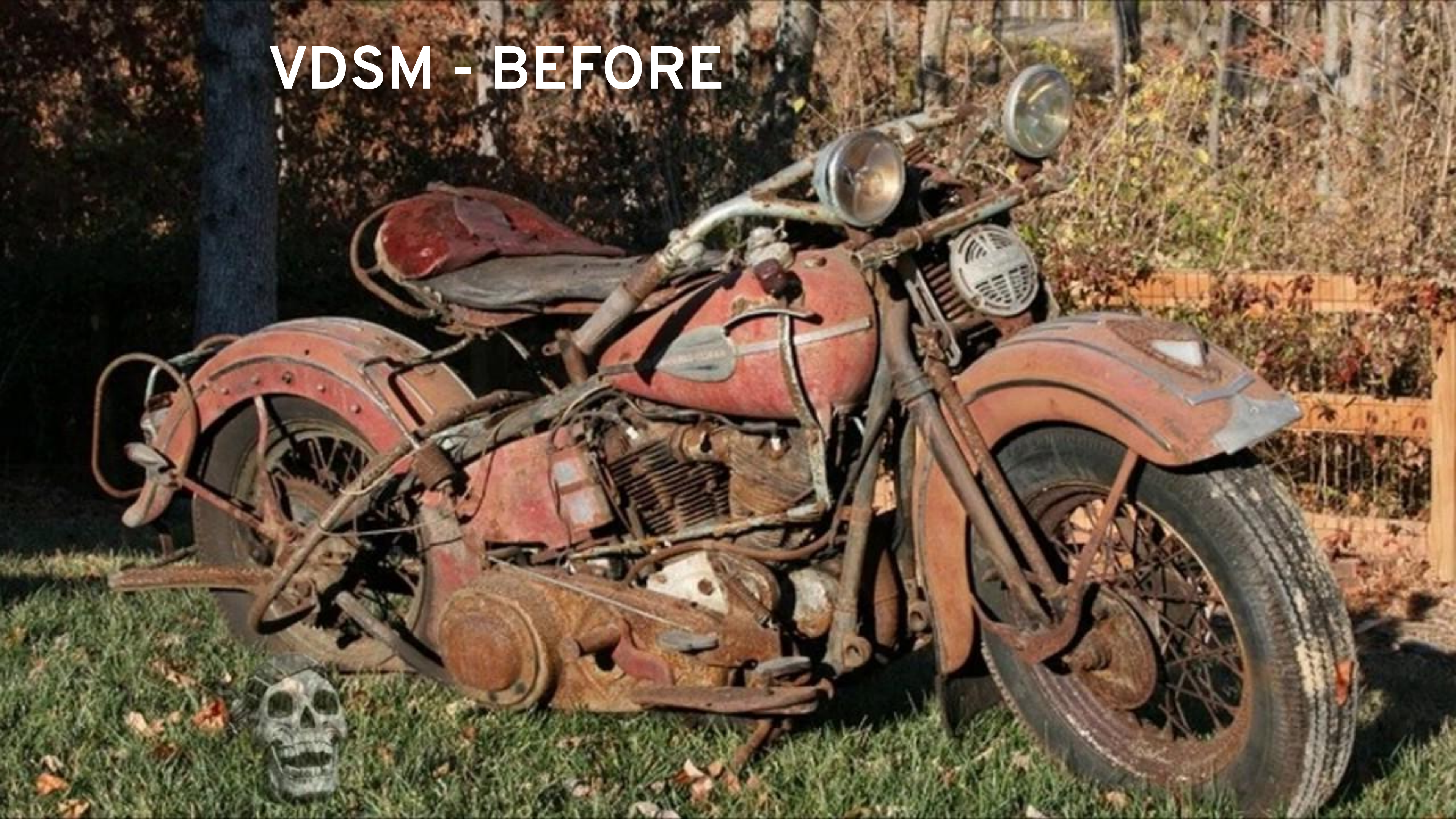
CHALLENGES

```
dom = tmp_repo.create_localfs_domain(
    name="Fano",
    version=5,
    block_size=user_mount_v5.block_size,
    max_hosts=user_mount_v5.max_hosts,
    remote_path=user_mount_v5.path)
```

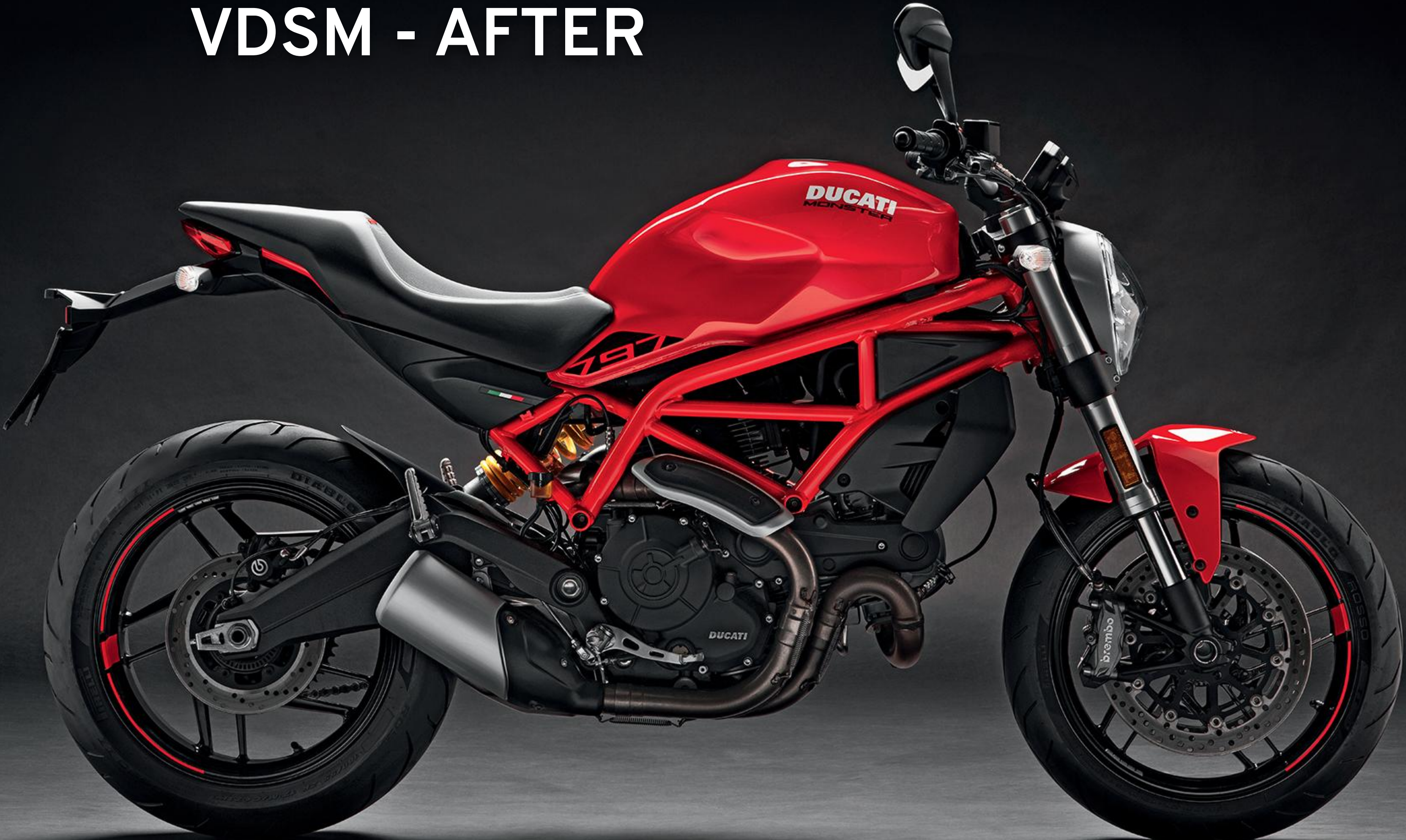
CHALLENGES

```
user_domain.createVolume(
    desc="Better Volume",
    diskType="DATA",
    imgUUID=img_uuid,
    preallocate=sc.SPARSE_VOL,
    capacity=10 * GiB,
    volFormat=sc.COW_FORMAT,
    volUUID=vol_uuid)
```

VDSM - BEFORE



VDSM - AFTER



QEMU fail to probe alignment with Gluster/XFS

4K CHALLENGES

~~QEMU fail to probe alignment with Gluster/XFS~~

Fix QEMU alignment probing

CHALLENGES

VM with 4K boot disk won't boot

```
<blockio logical_block_size="4096"
         physical_block_size="4096" />
```

CHALLENGES

~~VM with 4K boot disk won't boot~~

Emulate logical block size in QEMU

4K CHALLENGES

```
guest   (logical_block_size=512)
---------------------------------
qemu    (logical_block_size=4096)
---------------------------------
storage (logical_block_size=4096)
```

DETECTING BLOCK SIZE

DETECTING BLOCK SIZE

QEMU

1. read 1 byte
2. If ok, cannot detect, fallback to 4096
3. read 512 bytes
4. if ok, alignment is 512
5. read 4096 bytes
6. if ok, alignment is 4096

DETECTING BLOCK SIZE

QEMU - issues

- Cannot detect block size for Gluster/XFS and empty image. "qemu-img create" always allocates the first block to mitigate this.
- Cannot detect block size with NFS (no alignment requirements for direct I/O).

DETECTING BLOCK SIZE

vdsm

1. create temporary file
2. write 1 byte
3. If ok, cannot detect - use 1
4. write 512 bytes
5. if ok, use 512
6. write 4096 bytes
7. if ok, use 4096

DETECTING BLOCK SIZE

vdsm - issues

- No issue with Gluster/XFS and empty file
- Cannot detect block size with NFS

4K

USING BLOCK SIZE IN VDSM

Configuration

Gluster 4k enabled in 4.3.8

```
$ cat /etc/vdsm/vdsm.conf.d/gluster.conf
[gluster]
# Use to disable 4k support
# if needed.
enable_4k_storage = true
```

Reporting supported block size

Hosts report SD block size in Host.getCapabilities()

Reporting supported block size

```
class GlusterStorageDomain:

    supported_block_size = (
        sc.BLOCK_SIZE_AUTO,
        sc.BLOCK_SIZE_512,
        sc.BLOCK_SIZE_4K
    )
```

Auto detect block size

```
BLOCK_SIZE_AUTO = 0
```

When specifying block_size=0 vdsm will detect the block size automatically.

Validating storage block size

Requested storage block size is validated against detected storage block size.

```
StorageDomainBlockSizeMismatch: Block size
does not match storage block size:
block_size=512, storage_block_size=4096
```

Handling unknown block size

```
BLOCK_SIZE_NONE = 1
```

Internal vdsm value if vdsm cannot detect the block size. Use requested block size or we fall back to 512, keeping previous behavior.

Sanlock alignment

Alignment is determined by maximum number of hosts parameter.

Sanlock alignment

```
HOSTS_4K_1M = 250
```

Default maximum number of hosts is now 250 to have usual 1MB alignment also for 4k storage.

Storing block size and alignment

File storage domain metadata V5

```
# cat $SD_PATH/dom_md/metadata
ALIGNMENT=1048576
BLOCK_SIZE=4096
...
```

Create storage domain flow

- Detect block size of underlying storage.
- Validate the block size.
- Compute the alignment.
- Create SD metadata.
- Create directory structure.
- Initialize sanlock with block size and alignment.

MANAGING HOSTS

Host activation

Upon host activation call Host.getCapabilities() and store supported_block_size in the DB.

Storage domain creation

Upon storage domain creation check that block size auto detection is supported on all hosts.

Storage domain creation

- Call StorageDomain.create() with blockSize=0.
- Call StorageDomain.getInfo() to find actual block size.
- Store block size into DB.

Storage domain creation

If any of hosts doesn’t support block autodetection, engine will try to create domain with block size of 512 (will fail on 4k storage).

You need to upgrade host to 4.3.8 or add missing Gluster configuration.

TROUBLESHOOTING

4K TROUBLESHOOTING

Do all hosts support automatic block size detection?

4K TROUBLESHOOTING

```
$ vdsm-client Host getCapabilities
...
{
    "GLUSTERFS" : [
         0,
         512,
         4096,
    ]
    ...
```

4K TROUBLESHOOTING

Is storage domain metadata correct?

Red Hat

4K

TROUBLESHOOTING

```
# cat $SD_PATH/dom_md/metadata
...
VERSION=5
BLOCK_SIZE=4096
ALIGNMENT=1048576
```

TROUBLESHOOTING

Did engine ask to detect block size?

4K TROUBLESHOOTING

```
[vdsm.api] START
createStorageDomain(storageType=7, ...
domVersion=u'5', block_size=0, max_hosts=250,
...
```

4K TROUBLESHOOTING

Did vdsm detect the block size?

Red Hat

TROUBLESHOOTING

```
[storage.fileSD] Detected domain 2bca5015-4509
block size 4096
```

TROUBLESHOOTING

Did engine store the host capabilities in the database?

4K

TROUBLESHOOTING

```
# select supported_block_size from vds where
vds_name = 'my-host';

supported_block_size
--------------------------------------
{"FCP": [512], "NFS": [512], "ISCSI": [512],
"LOCALFS": [0, 512, 4096], "POSIXFS": [512],
"GLUSTERFS": [0, 512, 4096]}
```

4K TROUBLESHOOTING

Did engine store the block size in the database?

TROUBLESHOOTING

```
# select storage_name, block_size from
storage_domain_static;

   Storage_name          | block_size
-------------------------+------------
 ovirt-image-repository |     512
 gluster-vol5           |     4096
```

Red Hat

DEMO

Demo

Creating 4k Gluster storage domain

MORE INFO

- 4k RFE with links to 4k patches
- example of vdsm tests using 4k
- userstorage project
- ovirt.org

THANK YOU!

QUESTIONS?