Linux Storage Stack for the Cloud

Oct 2013

Yeela Kaplan
Software Engineer
Cloud Storage Team
Red Hat

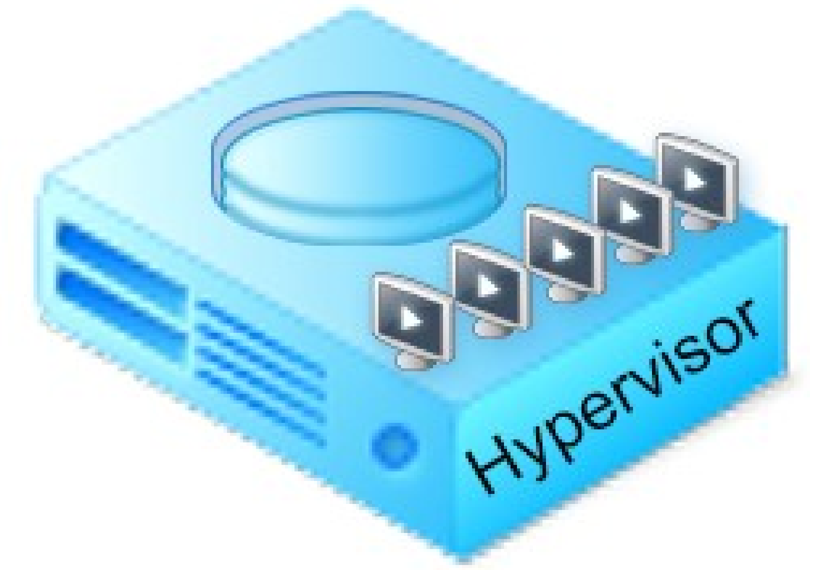
Who am I

Agenda

- Storage virtualization – Why, What and How?
- Challenges & Solutions in the enterprise
- oVirt Design and Implementation
- Q&A

Why storage virtualization?

- Limited physical disk interfaces
- Fixed size
- Can’t join disks
  - Performance
  - Storage array limitations
  - Multiple arrays



Storage virtualization

Create virtual devices with disk behavior

- Partition table
- Storage arrays
- LVM

Storage with benefits

- **Space flexibility**
- **Create devices ‘on the fly’**
- **Snapshots**

Image

A virtual disk for a vm

- One image is worth many volumes
- Volume:
  - YABS – Yet Another Block Sequence
- Volume types:
  - File
  - Block

What is
the problem?

Enterprise storage needs

Multiple **data centers**

x

hundreds of **hosts**

x

hundreds of **VMs**

x

multiple **disks**

x

potentially dozens of **snapshots**

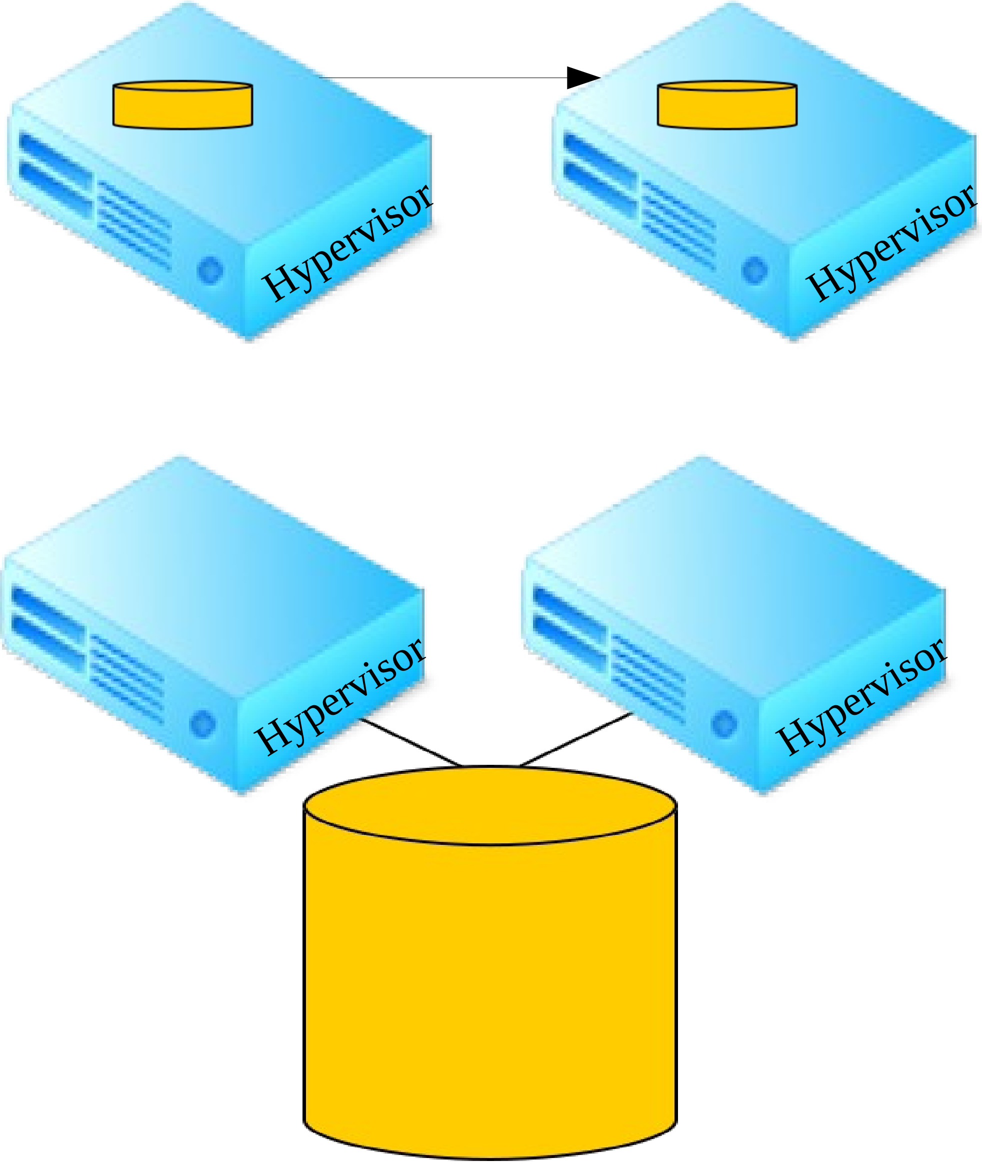
=

VERY BIG HEADACHE

Storage challenges

- **Host independent VMs**
- **Quantity of volumes**
- **Size of storage**

Host independent VMs



Solutions

- **Host independent VMs**
  - Shared storage
- **Quantity**
  - Creation on the fly
  - Templates
  - Centralized DB
- **Size**
  - Over-commitment
  - Thin provisioning
  - Templates (Shared data, same OS)

oVirt Implementation

oVirt snapshot

- Use **qcow2**
- file and block volumes
- provides COW volumes
- Thin volumes

File

File volumes

- **Quantity**
  - create and manage files using the file system
  - “Unlimited”
- **Size**
  - Dynamic sizing
  - Sparse files
- **Shared storage**
  - NAS
  - Synchronizing access

Block

Block volumes

- **Quantity**
  - How do we create a block device?
  - How many block devices are supported?
- **Size**
  - How can we resize a block volume?
  - Is thin provisioning possible?
- **Shared storage**

Using remote storage. But...

- Different storage vendors, models
- No standard interface

Why Block?

- File system performance overhead
- Customer requirements

Using SAN

- Initiator, Target, LUN = **GUID**
- **Transport** for the SCSI commands
  - FC
  - iSCSI
- **Redundancy**
  - Multiple targets for the same LUN
  - How can we tell if it’s the same LUN?

Redundancy and Multipath

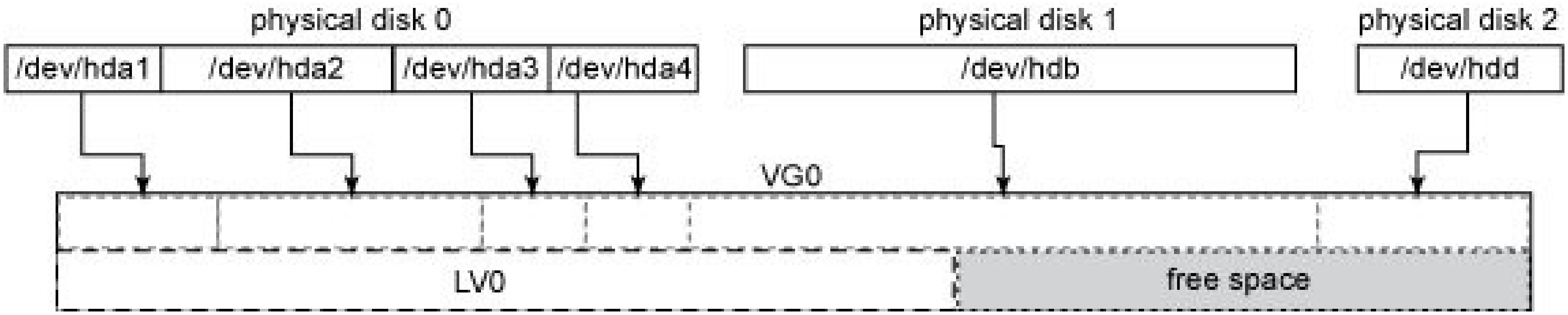
- **Using Multipath**
  - Query the storage to obtain the GUID
  - A new GUID is mapped through **device-mapper**
  - Use rules to choose the preferred path for the device
- Fail fast
- Pause VM
- I/O failure never reaches guest OS
- Auto resume

Why device-mapper?

- mapping block devices onto virtual block devices
- Used by multiple Linux storage stack components
- Multipath, RAID, LVM, crypt, etc...

Creating and managing block images

- LVM provides a unified interface
- Volume is implemented as an LV
- Easy provisioning: lvcreate, lvremove
- Thin provisioning: lvextend



* http://www.markus-gattol.name/ws/lvm.html

Very specialized use of LVM

Thin provisioning

- No use of LVM native thin provisioning
- LV initial size – 1GB
- Extend LV when:
  - VM paused due to ENOSPC
  - High watermark (monitoring qemu) identified

Need a clustered solution

- create, remove, extend are VG MD writes
- Simultaneous writes will cause MD corruption
- cLVM did not scale
- No synchronization mechanisms

LVM configuration

- **Hybrid mode and compartmentalization**
  - Runtime config, separate for vdsm
    - to avoid affecting anything else on the host
    - Allow admin to make changes outside of vdsm
  - LVM short filters
    - Speed up operations (by default LVM scans all devices)
    - Compartmentalize problems
    - Avoid accessing host ‘owned’ devices
- **Activate / deactivate**
  - Keep number of devices lower
  - Avoid refresh

Clustering LVM

- LVM MDA per PV by default
  - Problems
    - In clustered environment with more than 1 PV **will** cause corruption
    - Requires update of multiple areas to commit transaction
  - Solution
    - only 1 active MDA
- oVirt MD as LV and VG tags
- Lock type 4 (patches upstream)

SPM

- **Storage Pool Manager**
- **A role** assigned to one host
- Can be migrated to any host in a data center
- Creation, deletion and manipulation of volumes
- **Single meta data writer**

SPM algorithm

- Cluster membership based on
  - Light-weight leases for storage-centric coordination (Chockler and Malkhi 2004)
- Single recoverable leader
- Primitives: lease and renew
- Uniform
- Simple and efficient

SANLock

- Cluster membership, like SPM, based on
  - Light-weight leases for storage-centric coordination (Chockler and Malkhi 2004)
- Leases based on
  - Disk Paxos (modified for leases)

Summary

- Storage virtualization
- oVirt implementation
- oVirt snapshot
- File implementation
- Block implementation
- Multipath
- Device-mapper
- LVM
- SPM

Q&A

THANK YOU !

http://www.ovirt.org/Home
engine-devel@ovirt.org
vdsm-devel@lists.fedorahosted.org

#ovirt irc.oftc.net

ykaplan@redhat.com