


Storage Virtualization for KVM – Putting the pieces together

Bharata B Rao – bharata@linux.vnet.ibm.com
Deepak C Shettty – deepakcs@linux.vnet.ibm.com
M Mohan Kumar – mohan@linux.vnet.ibm.com
(IBM Linux Technology Center, Bangalore)

Balamurugan Aramugam - barumuga@redhat.com
Shireesh Anjal – sanjal@redhat.com
(RedHat, Bangalore)


Agenda

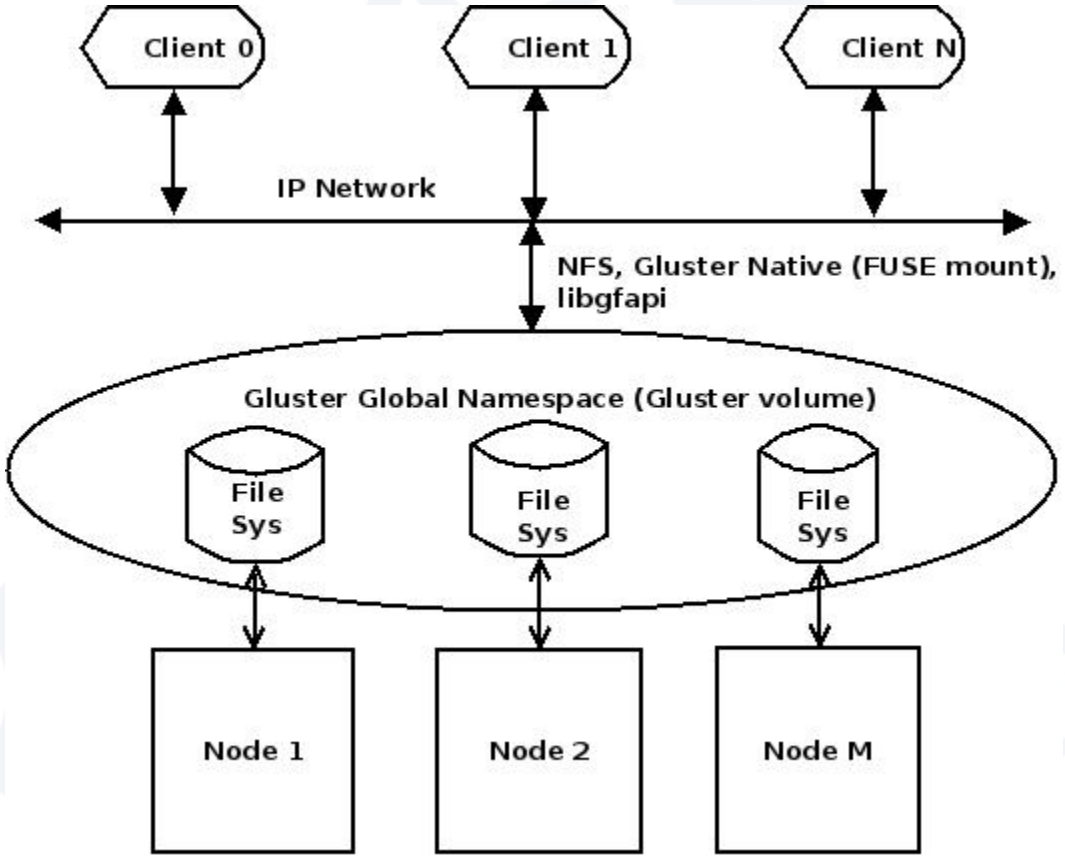
- Problems around storage in virtualization
- GlusterFS as virt-ready file system
  - QEMU-GlusterFS integration
  - GlusterFS – Block device translator
- Virtualization management - oVirt and VDSM
  - VDSM-GlusterFS integration
- Storage integration
  - libstoragemgmt

Problems in storage/FS in KVM virtualization

- Multiple choices for file system and virtualization management
- Lack of virtualization aware file systems
- File systems/storage functionality implemented in other layers of virtualization stack
  - Snapshots, block streaming, image formats in QEMU
- No well defined interface points in the virtualization stack for storage integration
- No standard interface/APIs available for services like backup and restore
- Need for a single FS/storage solution that works for local, SAN and NAS storage
  - Mixing different types of storage into a single filesystem namespace

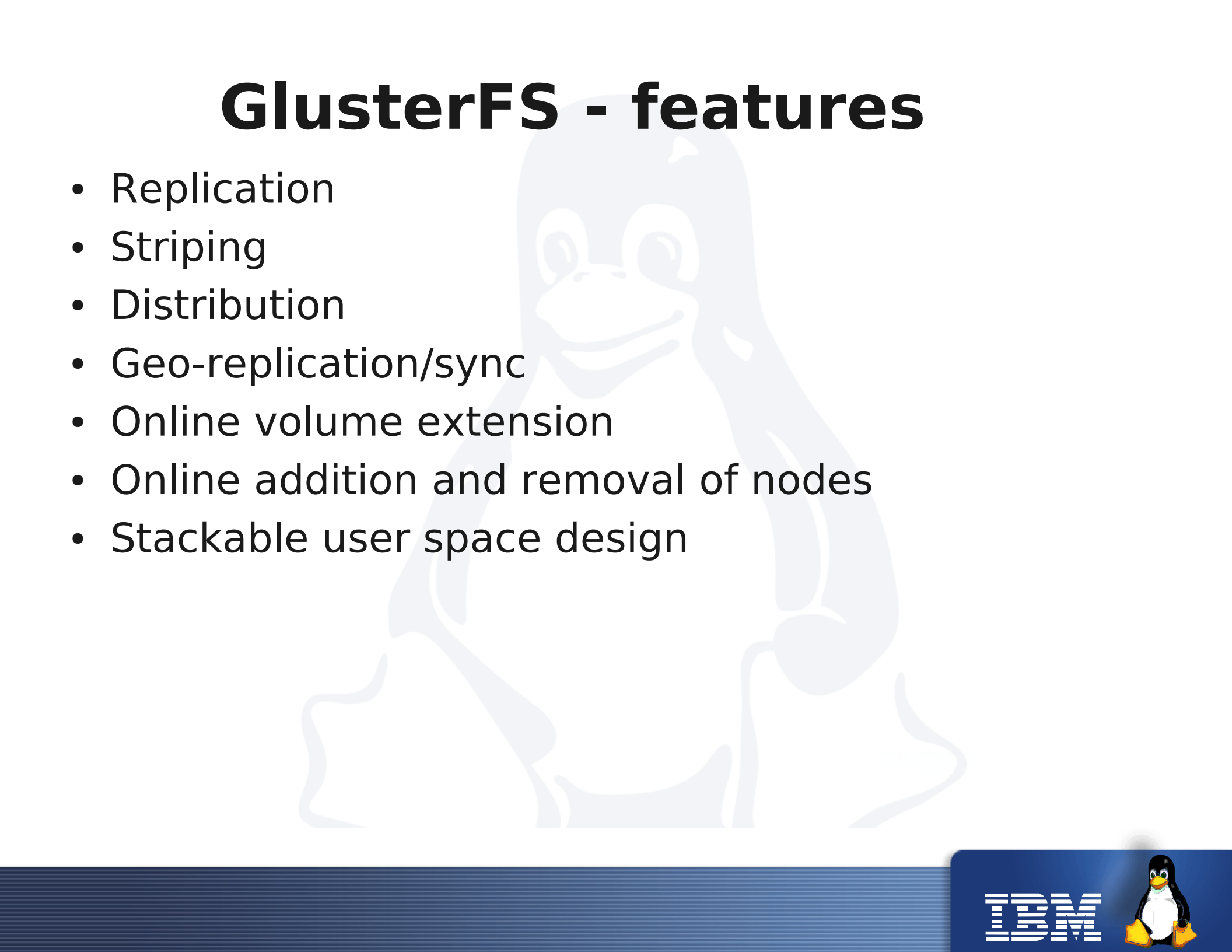
GlusterFS

- User space distributed file system that scales to several petabytes
- Aggregates storage resources from multiple nodes and presents a unified file system namespace



GlusterFS - features

- Replication
- Striping
- Distribution
- Geo-replication/sync
- Online volume extension
- Online addition and removal of nodes
- Stackable user space design

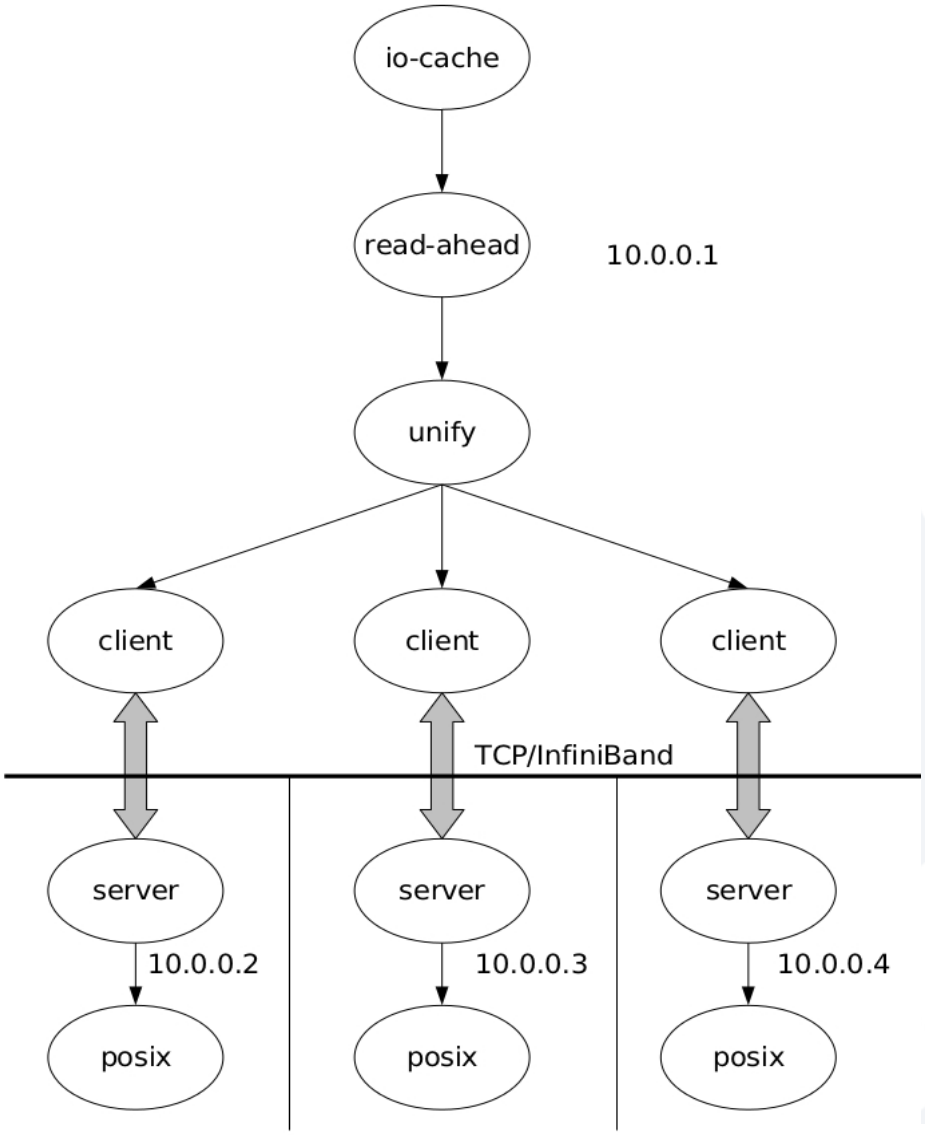


GlusterFS Translator

- Converts requests from users into requests for storage (*)
  - A shared library that implements file system calls
- Multiple translators can be stacked to form a translator tree
  - Every file system call to gluster will pass on via this tree
- Each translator provides a distinct functionality
  - storage/posix.so, performance/io-cache.so
  - protocol/client.so, protocol/server.so
- Modularity
  - Just enough translators to achieve the desired functionality
  - Dynamic addition and removal of translators

(*) Borrowed from Jeff Darcy's Gluster workshop slides

Translator tree example

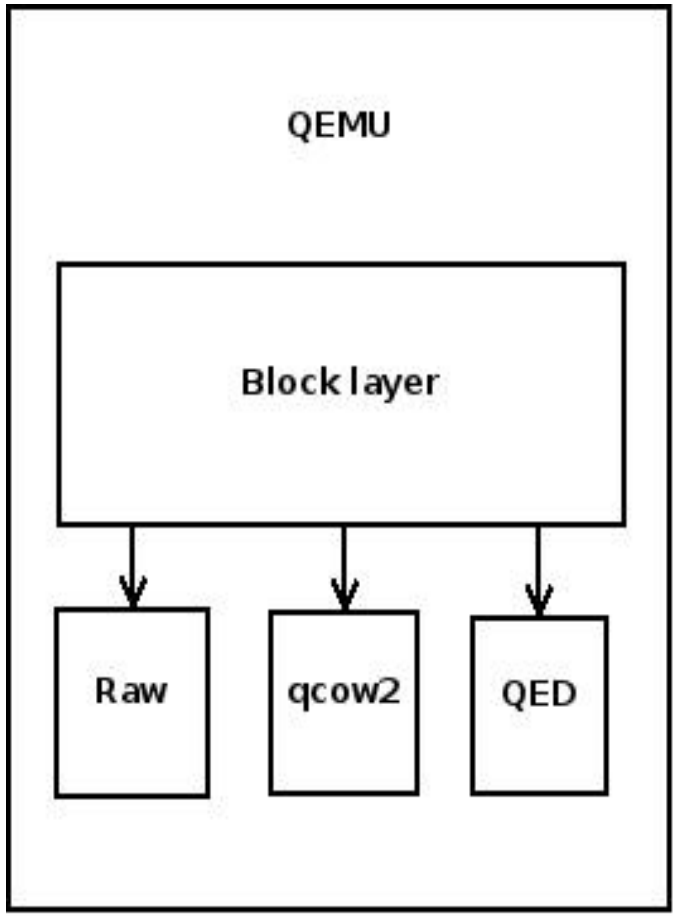


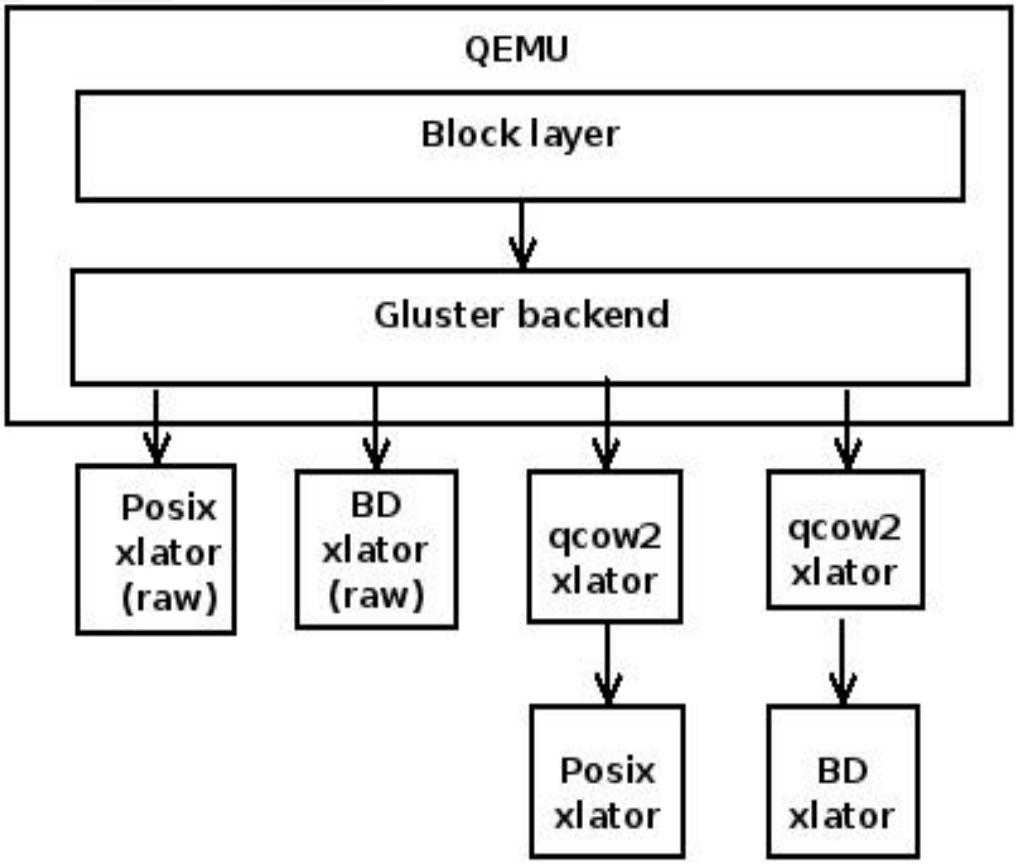
Source: gluster.org

Enabling GlusterFS for Virtualization use

- QEMU-GlusterFS integration
  - Native integration, no FUSE mount
  - Gluster as QEMU block back end
  - QEMU talks to gluster and gluster hides different image formats and storage types underneath
- Block device support in GlusterFS via Block Device translator
  - Logical volumes as VM images

GlusterFS back end in QEMU





QEMU-GlusterFS integration

- New block driver in QEMU to support VM images on gluster volumes
  - Uses libgfapi to do IO on gluster volumes directly w/o FUSE mount
- Usage

  -drive file=gluster://server[:port]/volname/image[?transport=...]
- FIO Numbers (Seq read, 4 files with direct io, qemu options: if=virtio, cache=none)

	Aggregate BW(kB/s)	Min BW(kB/s)	Max BW(kB/s)
Base	63076	15769	17488
FUSE mount	29392	7348	9266
QEMU-GlusterFS native integration	53609	13402	14909
QEMU-GlusterFS native with custom client side volfile	62968	15742	17962

GlusterFS BD xlator

- BD xlator exports block devices at server side as files to gluster clients
  - Currently supports LVMs only
  - Exploring exporting LUNs as files (Future)
- Advantages
  - Direct block device access, no FS overhead
  - Provides VM thin provisioning and snapshots by leveraging thin provisioning and snapshot features of LVM
  - Ease of use and management with block device backed VM images as files
  - Inherently thin provisioned images using dm-thin targets (WIP)
- Fitting GlusterFS in SAN environment

… BD xlator

- Leaf (server side) translator
- Exports LVM volume group as directory and logical volumes within it as files
  - VM image is a file which in turn is an LV
- Posix calls mapping
  - create – LV creation
  - link – Full clone
  - soft link – Linked clone/snapshot
  - truncate – LV resize

Using BD xlator

- Creating gluster volume with BD backend
  - # gluster volume create <volname> device lv <host>:/<vg-name>
- Creating a VM image on BD backend
  - # gluster bd create <volname>:/<vg-name>/<lv-name> <size>
- Clone and snapshot
  - # gluster bd clone <volname>:/<vg-name>/<lv-name> <new-lv>
  - # gluster bd snapshot <volname>:/<vg-name>/<lv-name> <new-lv>
- Commands from gluster mount point
  - # cd /gluster-mount-point/vg-name
  - # touch lv1 /* create an LV */
  - # truncate -s <size> lv1 /* sets the size of LV */
  - # ln lv1 lv2 /* full clone of lv1 in lv2 */
  - # ln -s lv1 lv2 /* linked clone of lv1 in lv2 */

QEMU-GlusterFS advantages

- VM images as files in all scenarios (esp SAN using BD xlator)
  - Ease of management
  - File system utilities for backup from GlusterFS FUSE mount (Future)
- Off-loading QEMU from storage/FS specific work
  - File system driven snapshots, clones (via BD xlator)
- Storage migration that is transparent to QEMU
  - Driven by GlusterFS (Future)
- Translator advantages
  - User space pluggable VFS, modularity
  - Lean storage-stack

libvirt support for GlusterFS

- RFC patches out on libvirt mailing list to support gluster drive specification in QEMU
  - https://www.redhat.com/archives/libvir-list/2012-August/msg01625.html
- Libvirt XML specification

```
<disk type='network' device='disk'>
  <driver name='qemu' type='raw'/>
  <source protocol='gluster' name='volume/image'>
    <host name='example.org' port='6000' transport='socket'/>
  </source>
</disk>
```

oVirt and VDSM

- oVirt
  - Virtual data center management platform
  - KVM based virtualization environment
  - VM life cycle, storage, network management
  - Self service portal
  - Depends on VDSM
- VDSM
  - oVirt node agent
  - Node virtualization management API
  - Uses libvirt/QEMU for VM management
  - Responsible for storage, network, host, VM management etc

VDSM storage domains

- Storage domain
  - Standalone storage entity
  - Stores images and associated data aka disk image repository
- Domain types
  - File domain
    - NFS and localFS
    - PosixFS – support for posix complaint storage back end
  - Block domain
    - iSCSI and FCP

GlusterFS storage domain in VDSM

- PosixFS approach via GlusterFS FUSE mount is used currently
- Support in VDSM to exploit QEMU-GlusterFS native integration
  - PosixFS + VDSM hooks approach
    - Modifies libvirt XML to support gluster specification in QEMU
    - Non-standard, hooks not shipped with VDSM rpm
  - GlusterFS as network disk type under PosixFS
    - Adds GlusterFS as network disk in libvirt part of VDSM
    - Not ideal fit, not future-proof
  - GLUSTERFS_DOMAIN approach - preferred
    - New storage domain type, inherits mostly from NFS domain, Patches under review

GlusterFS support in oVirt/VDSM

- GUI and REST API for managing gluster clusters
  - Create, expand, shrink Gluster clusters
  - Create and manage Gluster volumes
- Leveraging oVirt platform
  - Gluster related verbs in VDSM
    - vdsm-gluster plugin – separate rpm
  - Gluster related commands and queries in oVirt engine backend
  - Gluster specific UI changes and REST APIs
  - Configurable Application Mode: virt only / gluster only / virt + gluster

...GlusterFS support in oVirt

- Completed
  - Enable gluster on a cluster in oVirt
  - Create and delete volumes
  - Manage volume lifecycle: start/stop,add/remove bricks, set/reset options
  - Audit logs
  - Advanced Volume search with auto-complete
- Future work
  - CIFS export
  - Option to configure volume to be used as storage domain in oVirt
  - Support for Bootstrapping and SSL
  - Import existing Gluster cluster into oVirt engine
  - Async tasks (rebalance, replace-brick, etc)
  - Geo-replication
  - Top / Profile
  - UFO (Unified File and Object Storage)

Storage Array integration

- Exploiting storage array capabilities from the virtualization stack
- Need for a stable programming interface for managing storage hardware
- Taking advantage of storage array off-load features like
  - Thin provisioning
  - Snapshots
  - Array assisted copy

libstoragemgmt

- Library to programmatically manage storage hardware in a vendor-neutral way
- C APIs for storage management, python bindings
- Manages SAN and NAS
- Exploits storage array off-load capabilities
- Plugins for vendor-specific storage
- Example usage
  - Create LUN
  - Enumerate LUNs
  - List capabilities

VDSM-libstoragemgmt integration

- Goals
  - Ability to plugin external storage array into oVirt/VDSM virtualization stack, in a vendor neutral way
  - Ability to list features/capabilities and other statistical info of the array
  - Ability to utilize the storage array offload capabilities from oVirt/VDSM
    - Array assisted thinp, copy, snapshot
- RFC posted and discussed in the community - https://lists.fedorahosted.org/pipermail/vdsm-devel/2012-June/001105.html
  - Needs more investigation on how libstoragemgmt can fit into VDSM repo engine
  - Needs more discussion in the community

Future Work

- T10 SCSI extensions (xcopy, writesame)
  - VFS interfaces, FS support
- Storage integration
  - Storage off-loads
  - libstoragemgmt plugins
- GlusterFS storage domain in VDSM
  - Drive to completion
- Mapping VM's to LUN's
  - Extending GlusterFS BD xlator to support LUN's in the back end
- dm-thin support
  - dm-thin support from GlusterFS BD xlator

References

- Latest QEMU-GlusterFS patches (v6)
  - http://lists.gnu.org/archive/html/qemu-devel/2012-08/msg01536.html
- Mohan's Block device xlator patches
  - http://review.gluster.org/3551
- Harsh's RFC patches for libvirt support
  - https://www.redhat.com/archives/libvir-list/2012-August/msg01625.html
- Deepak's Patches that add VDSM support
  - http://gerrit.ovirt.org/6856
- Video demo of using QEMU with GlusterFS
  - http://www.youtube.com/watch?v=JG3kF_djclg
- QEMU git tree – git://git.qemu.org/qemu.git
- GlusterFS git tree – git://git.gluster.com/glusterfs.git
- QEMU-GlusterFS Benchmark details
  - http://lists.nongnu.org/archive/html/qemu-devel/2012-07/msg02718.html
  - http://lists.gnu.org/archive/html/gluster-devel/2012-08/msg00063.html

References

- oVirt and VDSM
  - http://www.ovirt.org
- libstoragemgmt
  - http://sourceforge.net/projects/libstoragemgmt/

Legal Statement

- This work represents the view of the authors and does not necessarily represent the view of IBM.
- IBM, IBM(logo) are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademark or service marks of others.
- There is no guarantee that the technical solutions provided in this presentation will work as-is in every situation.