GlusterFS

For KVM Users and Developers

Stefan Hajnoczi <stefanha@redhat.com>

Red Hat

November 2012

Agenda

- What is GlusterFS?
- High-level concepts
- Architecture
- KVM integration
- Future of GlusterFS and KVM

What is GlusterFS?

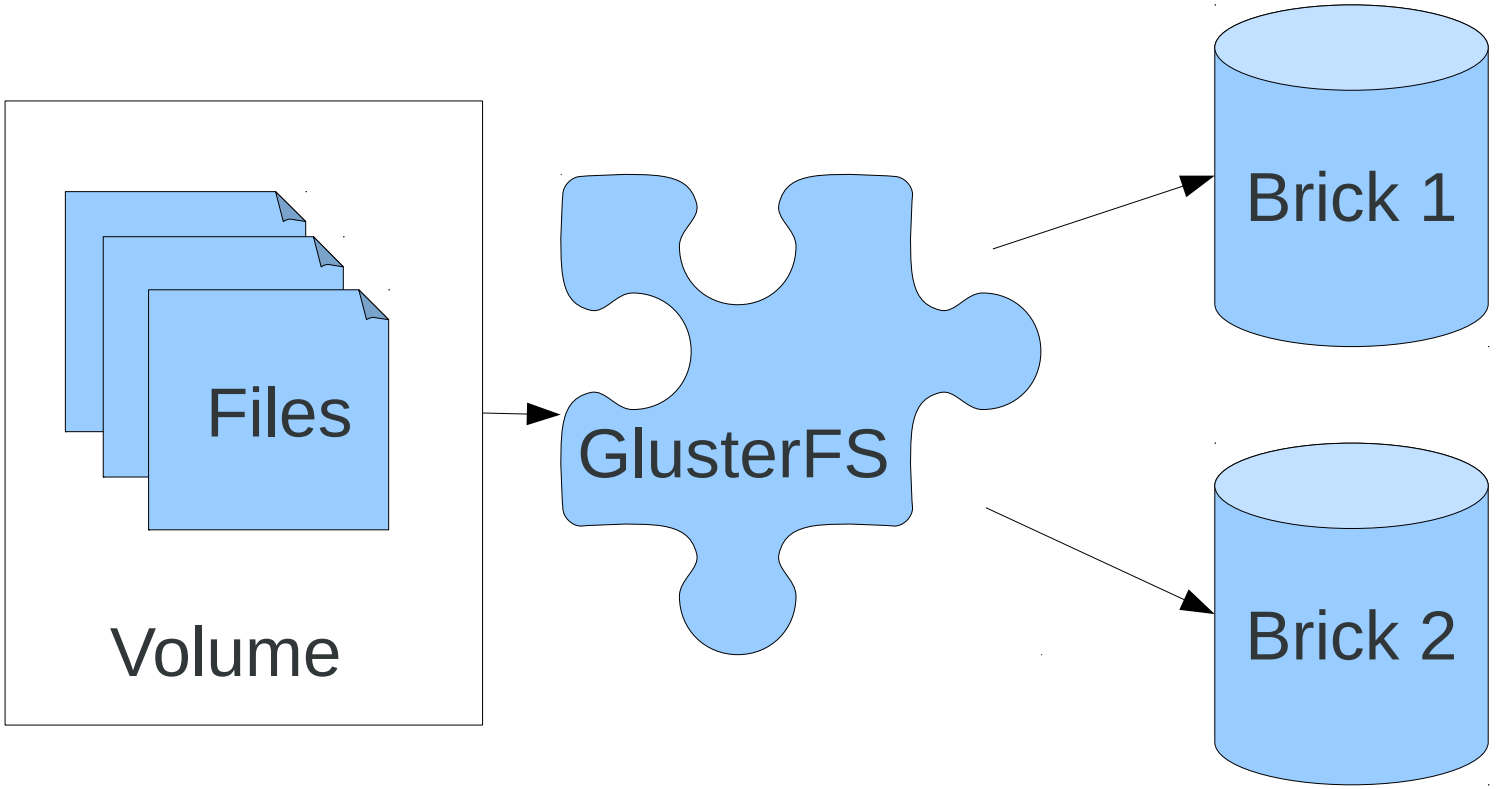
- Distributed file-level storage
- Commodity hardware
- Open source software
- Clients: NFS, CIFS, FUSE, shared library
- No central metadata server
- Cluster features for failures and self-healing
- Scale-out design for adding/removing storage

Why GlusterFS and KVM?

- KVM deployments increasingly need distributed storage:
  - Live migration without a SAN
  - Cloud scenarios where any host can run any VM
  - VMs must be available across host failure
- GlusterFS strengths:
  - Covers many disk image use cases today
  - Flexibility (more on this later)
  - Track record of successful integration with Hadoop and OpenStack Swift

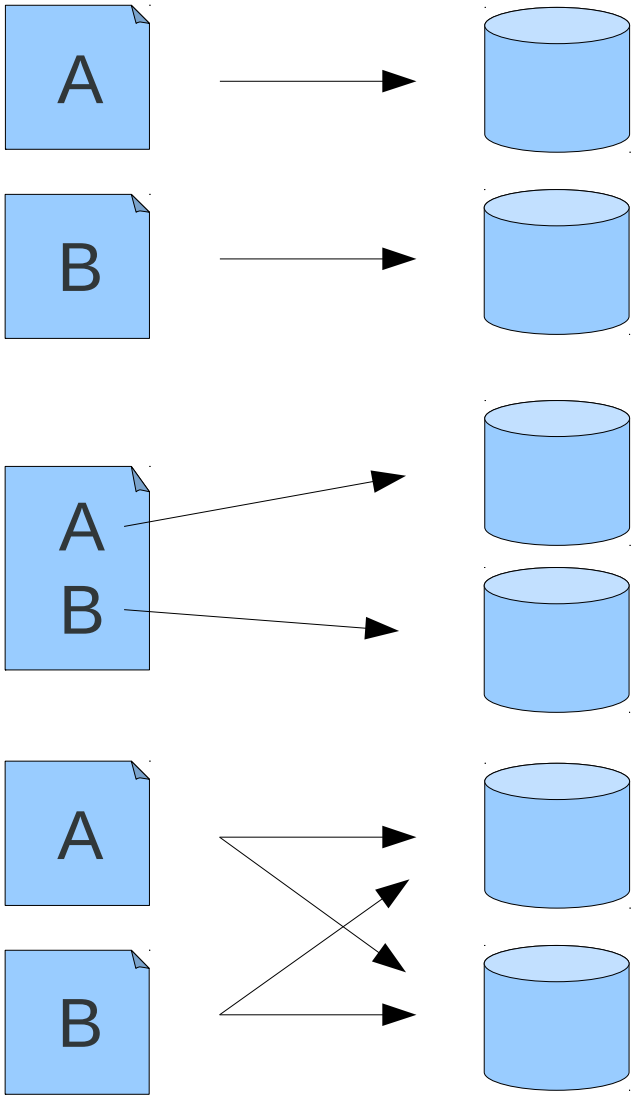
GlusterFS concepts

- *Volume* provides namespace for files
- Data can be spread across storage entities aka *bricks*



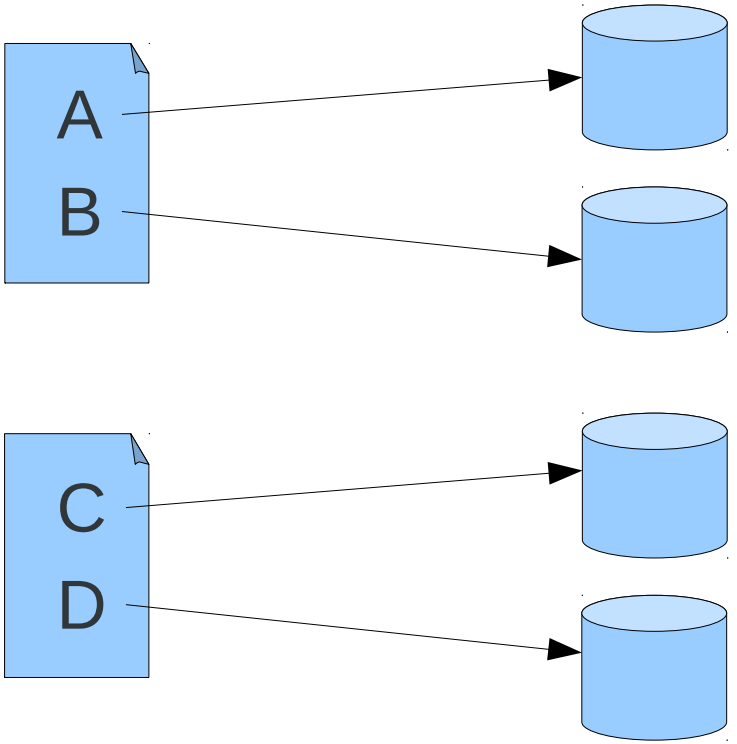
Volume types

- *Distributed*
  - Files placed across bricks
- *Striped*
  - Data striped across bricks
- *Replicated*
  - Files mirrored across bricks



Combining volume types

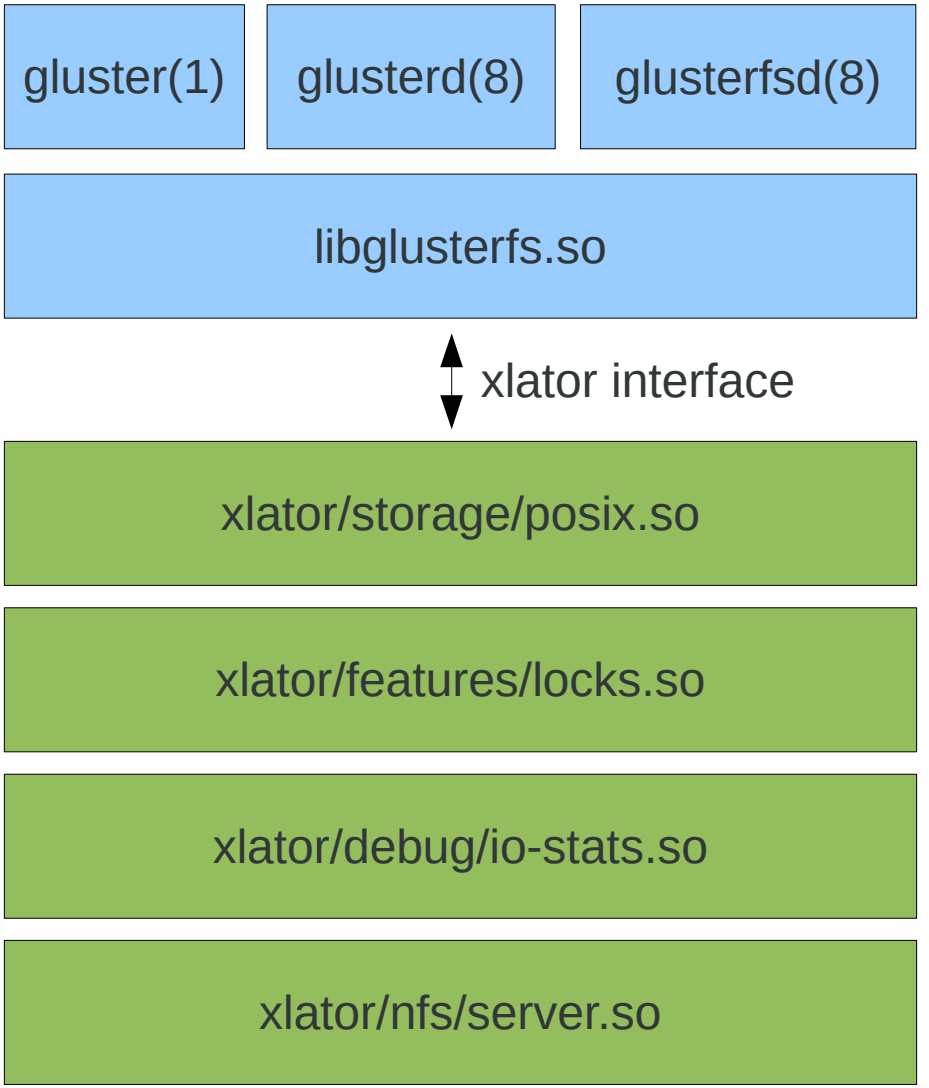
- *Distributed Striped*
  - Files placed across bricks and file data striped across bricks



- Additional volume type combinations described in Administration Guide:
  - See http://gluster.org/ for supported combinations

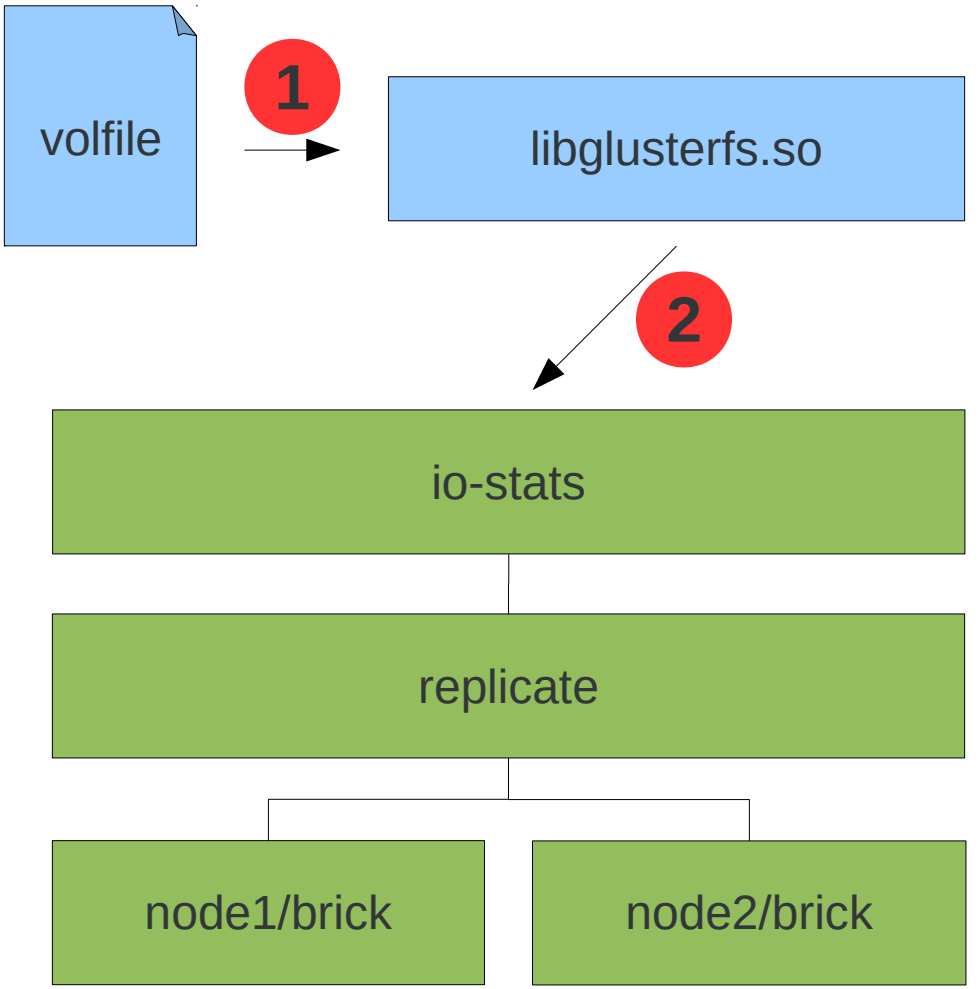
GlusterFS architecture

- gluster(1) mgmt CLI
- glusterd(8) mgmt daemon
- glusterfsd(8) storage daemon
- libglusterfs.so storage engine library
- Functionality is provided by xlator plugins
- xlator interface is roughly like kernel VFS



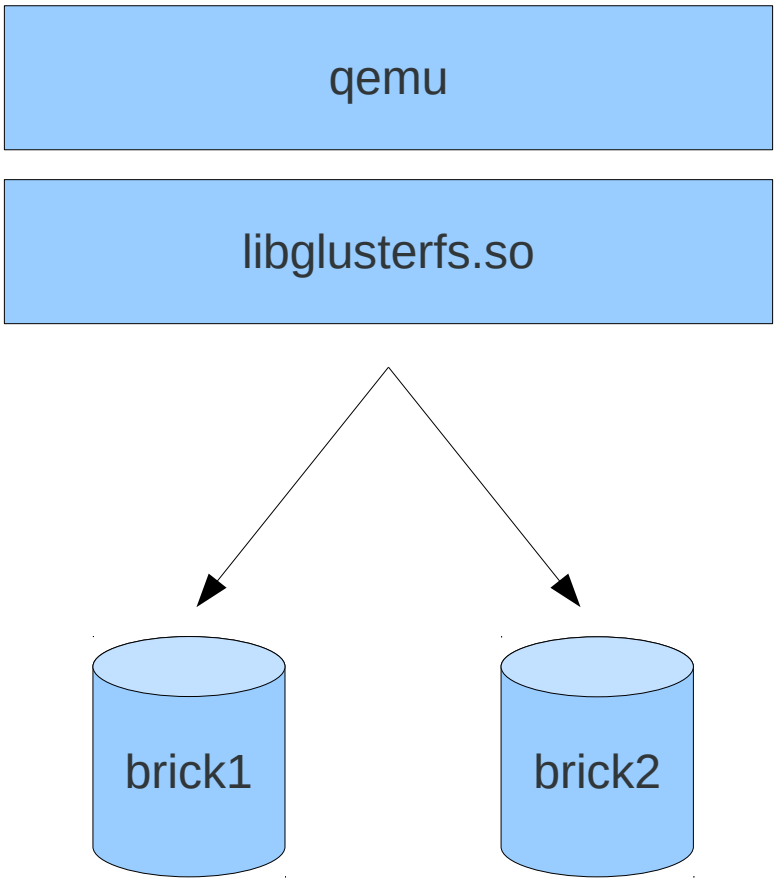
Volfiles - xlator graph configuration

- Volfile describes an xlator graph configuration
- libglusterfs.so creates xlator graph given a volfile
- Volfile is generated by glusterd(8)...
- ...but can be hand-written by advanced users



Example: disk image on replicated volume

- Host issues mirrored writes directly to brick1 and brick2
  - No single point of failure
  - Great scalability for striped volumes



For developers: extending GlusterFS

- Write an xlator to extend GlusterFS
- xlator interface includes VFS operations plus more:
  - Open, read, write, close, ...
- If your storage feature is not QEMU-specific, GlusterFS may be a good place to implement it
  - Quorum policies (voting on read operation results)
  - Live migration of image files
- An xlator could even expose qcow2 files and their snapshots

Missing feature: snapshots

- Snapshots are a key feature for disk images:
  - Cheaply create new VM from master image
  - Preserve disk image before trying risky operation
  - Backup disk image by taking a consistent copy
- Snapshot concept missing from GlusterFS<=3.3
- Workarounds:
  - Use qcow2 on GlusterFS
  - Use LVM or btrfs snapshots underneath GlusterFS (risky, requires understanding of brick layout)

Upstream status of GlusterFS integration

Component	Status	Notes
GlusterFS	Merged for 3.4	KVM depends on new glfs.h API
QEMU	Merged for 1.3	
Libvirt	Patch on mailing list	

Thank you!

- GlusterFS community: http://gluster.org/
- QEMU community: http://qemu.org/

- Blog: http://blog.vmsplice.net/
- Email: stefanha@redhat.com
- IRC: stefanha on #qemu (oftc) and #gluster (freenode)

Object Storage (REST API)

- New in GlusterFS 3.3
- Amazon S3-style object storage has become popular
- GlusterFS works with OpenStack Swift object storage
- Use cases:
  - Allow customers to upload VM templates
  - REST API for exporting disk images
  - Use “cloud” tools with KVM and GlusterFS cluster