Gluster can’t scale - Is it a reality or a past?

Atin Mukherjee

Engineering Manager, Red Hat Gluster Storage

Agenda

- Brief about GlusterFS
- Scaling bottlenecks in Gluster
- How did (and are) we overcoming them
- Projects/features like GlusterD2, brick multiplexing, sub-directory fuse mount, flexi sub vol.
- Q&A

What is GlusterFS

- A general purpose scale-out distributed file system.
- Aggregates storage exports over network interconnect to provide a single unified namespace.
- Filesystem is stackable and completely in userspace.
- Layered on disk file systems that support extended attributes.

Some key terminologies

- Trusted Storage Pool (TSP) - Trusted Storage Pool (cluster) is a collection of storage servers.
- Peer – Node in the cluster
- Volume - A volume is a logical collection of bricks
- Brick - A brick is the combination of a node and an export directory – for e.g. hostname:/dir

Present..

- RHGS cluster of ~150 nodes run in production deployments (with some caveats)
- Community version - ~200-250 nodes (with some caveats)
- With OpenShift Deployments, ~600 volumes on a 3 node cluster. (Yes! Thanks to brick multiplexing feature)

Scaling bottlenecks

- **Node scalability**
  - Is 150 nodes deployment in hybrid cloud sufficient with the growing storage demand?
  - 1000 nodes cluster even be a reality?
    - GD1 (GlusterD 1) ====> GD2 (GlusterD2)
    - GD2 (released as tech preview with GlusterFS 4.0) will become the default management plane for Gluster in coming releases

Scaling bottlenecks – Node scalability

- GlusterD - Manages the cluster configuration for Gluster

  Peer membership management

  Elastic volume management

  Configuration consistency

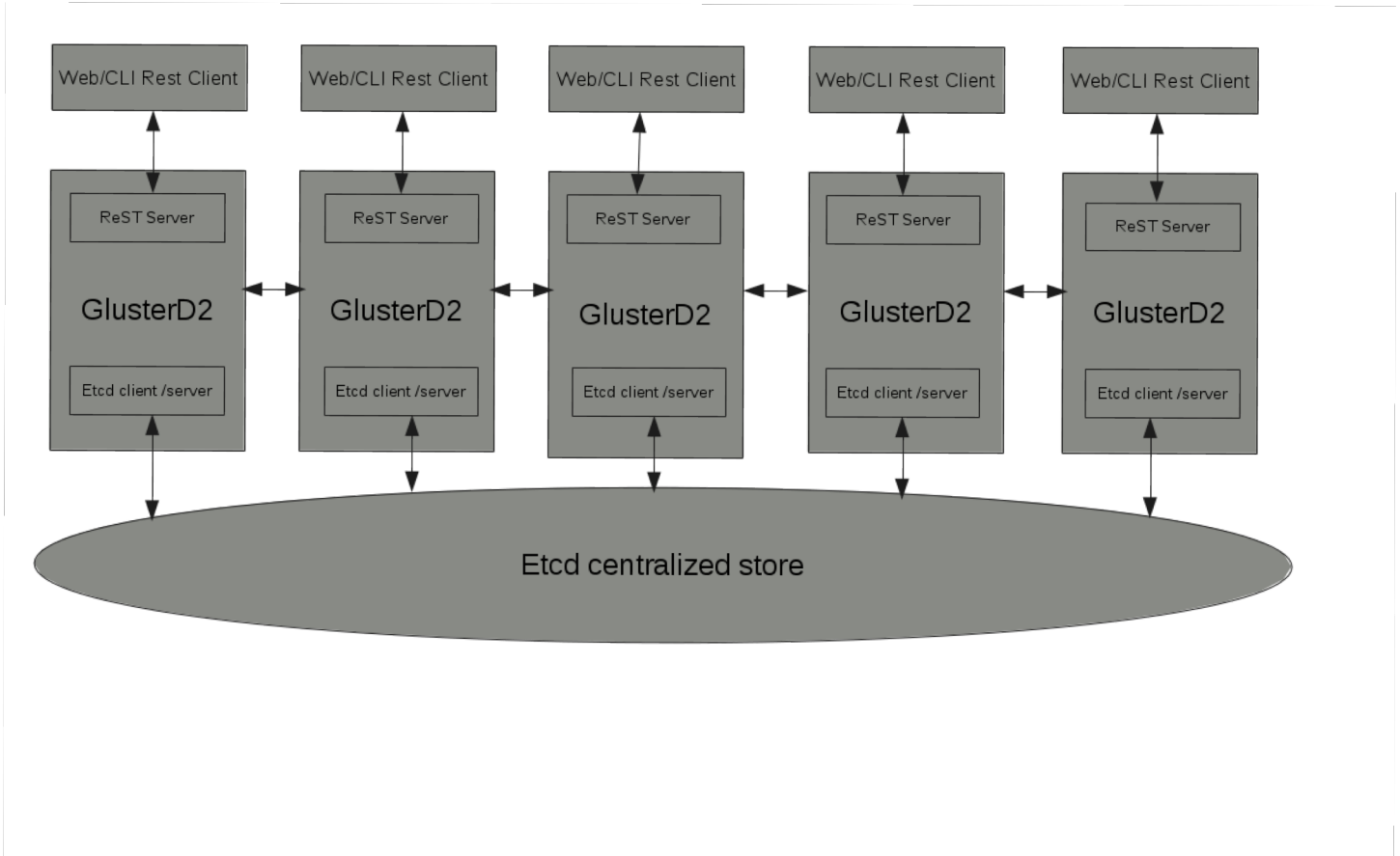
  Distributed command execution (orchestration)

  Service management (manages Gluster daemons)

Scaling bottlenecks – Node scalability

- Limitations of GD1
  - Non linear node scalability
  - Lack of native ReST APIs
  - Code maintainability and easier feature integrations in the form of plugins
- What's in for GD2
  - Centralized store (etcd)
  - Better transaction framework
  - ReST API support
  - Intelligent Volume Provisioning

Node scalability - GD2 Architecture



Scaling bottlenecks

- **Volume scalability**
  - Thousands of volumes in cluster?
    - Brick Multiplexing

Volume scalability – Brick Multiplexing

- Brick Multiplexing (GlusterFS 3.10 onwards)
  - 1:1 process & brick ===> 1:Many process & bricks
  - Reduced resource (Port, Memory, CPU) consumption
  - Not a performance enhancer!
  - One of the gluster core salient features for container native storage's success.

Other ways to scale

- Fuse based sub directory export (GlusterFS 3.12 onwards)
  - Single volume, sub directory based isolation for different clients
  - Namespace access control through auth.allow
  - Gluster snapshots need to work at volume level

Other ways to scale

- Gluster sub-vol (https://github.com/gluster/gluster-subvol)
  - Use subdirectories of Gluster volumes as persistent volumes in openshift.io
  - glfs-vol – a flex volume plugin to allow mounting Gluster subdirectories into containers.
  - Uses xfs quota for setting up quotas for the directory based space control.

Other scaling challenges

- Resource control
- Debugg-ability
- User experience

Q & A

Thank you!

- Reach us @
  - gluster-devel@gluster.org gluster-users@gluster.org
  - IRC : #gluster, #gluster-dev @freenode