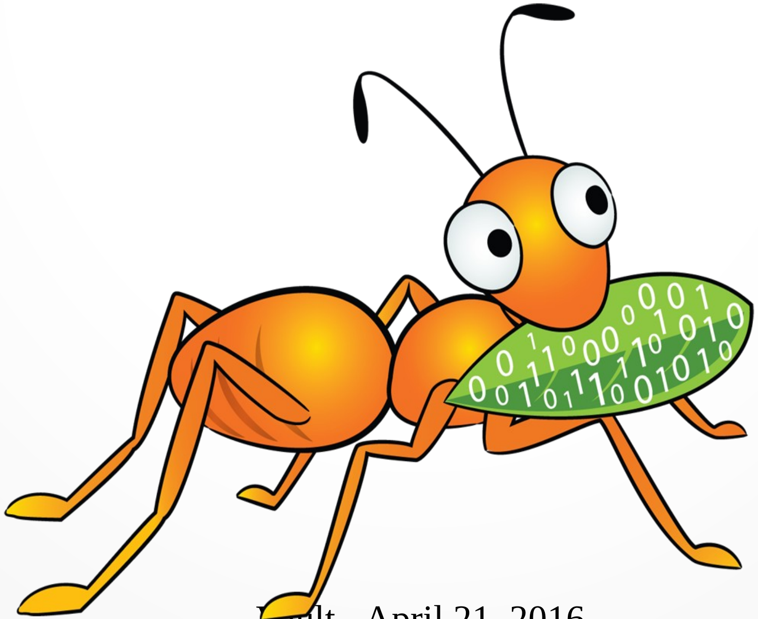
GlusterD 2.0



Atin Mukherjee

"An engineer by profession, a musician by passion"

Gluster Co Maintainer

Senior S/W Engineer @ Red Hat

Reach me @ IRC: atinm on #freenode, Twitter: @mukherjee_atin, mailto : amukherj@redhat.com, github: github.com/atinmu

Agenda

- GlusterFS – a brief overview
- GlusterFS concepts
- Introduction to legacy GlusterD (GlusterD 1.0)
- Why GlusterD 2.0
- High level architecture
- Components
- Upgrades consideration
- Q&A

GlusterFS

- Open-source general purpose scale-out distributed file system
- Aggregates storage exports over network interconnect to provide a single unified namespace
- Layered on disk file systems that support extended attributes

- No meta-data server
- Modular Architecture for Scale and Functionality
- Heterogeneous commodity hardware
- Scalable to petabytes & beyond

GlusterFS Requirements

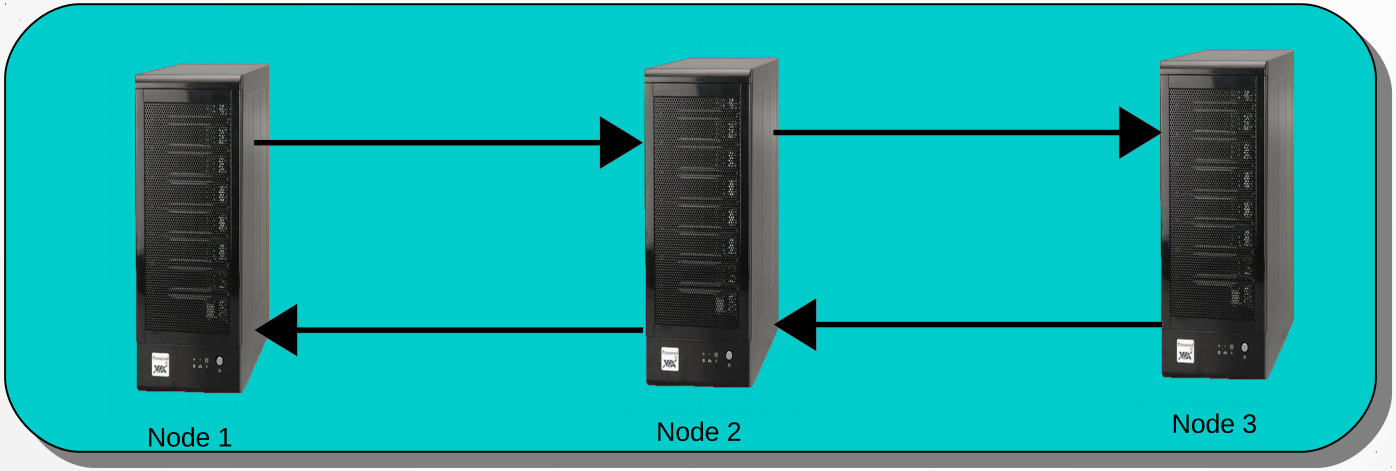
A node is server capable of hosting GlusterFS bricks

- Server
  - Intel/AMD x86 64-bit processor
  - Disk: 8GB minimum using direct-attached-storage, RAID, Amazon EBS, and FC/Infiniband/iSCSI SAN disk backends using SATA/SAS/FC disks
  - Memory: 1GB minimum
- Networking
  - 10 Gigabit ethernet
  - Infiniband (OFED 1.5.2 or later)
- Logical Volume Manager
  - LVM2 with thin provisioning
- File System
  - POSIX w/ Extended Attributes (EXT4, XFS, BTRFS, …)

GlusterFS Concepts – Trusted Storage Pool

A collection of storage servers (nodes)

- Also known as cluster
- Trusted Storage Pool is formed by invitation – ***“probe”***
- *Members can be dynamically added and removed from the pool*
- Only nodes in a Trusted Storage Pool can participate in volume creation



GlusterFS Concepts – Bricks

A unit of storage used as a capacity building block

- A brick is the directory on the local storage node
- Layered on posix compliant file-system (e.g. XFS, ext4)
- Each brick inherits limits of the underlying filesystem
- It is recommended to use an independent thinly provisioned LVM as brick
  - Thin provisioning is needed by Snapshot feature

GlusterFS Concepts – Volume

A volume is a logical collection of one or more bricks

- Node hosting these bricks should be part of a single Trusted Storage Pool
- One or more volumes can be hosted on the same node

What is GlusterD

- Manages the cluster configuration for Gluster
  - ✓ Peer membership management
  - ✓ Elastic volume management
  - ✓ Configuration consistency
  - ✓ Distributed command execution (orchestration)
  - ✓ Service management (manages Gluster daemons)

Issues with legacy GlusterD design

- N * N exchange of Network messages for peer handshaking
- Not scalable when N is probably in hundreds or thousands
- Replicate configuration (management) data in all nodes
- Initialization time can be very high
- Can end up in a situation like “whom to believe, whom not to” - popularly known as split brain
- Lack of transaction rollback mechanism

Why GlusterD 2.0

“Configure and deploy a 'thousand-node' Gluster cloud”

Why GlusterD 2.0 (ii)

- More efficient/stable membership
  - Especially at high scale
- Stronger configuration consistency
- Non trivial effort in adding management support for Gluster features (modularity & plugins)

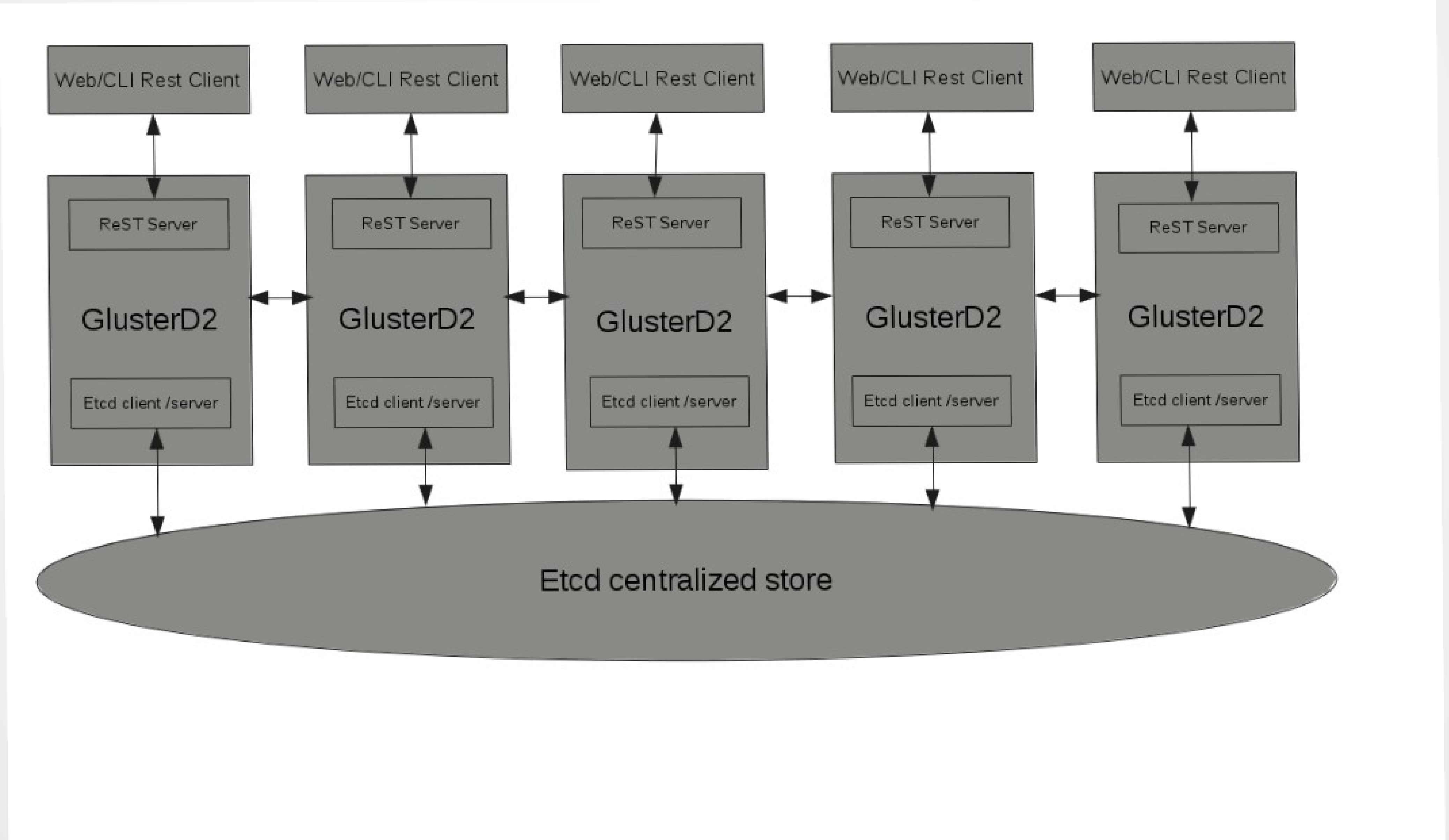
Language choice

- Legacy GlusterD is in C

  “*It wasn't enough just to add features into existing language because sometimes you can get more in the long run by taking things away...They wanted to start from scratch and rethink everything.*” - Robert.C.Pike

- GD2 is in Go and written from scratch!
  - Suits best in writing a management plane of a file system (distributed)
  - Garbage collection
  - Standard libraries support
  - One binary approach
  - Built in rich features like go routines, channels for concurrency

High Level Architecture



Component breakdown

- ReST interfaces
- Central store - etcd management & bootstrapping
- RPC Mechanism
- Transaction framework
- Feature plug-in framework
- Flexi volgen

ReST interface

- HTTP ReST Interface
- ReST API support
- CLI to work as ReST Client

Central store - etcd

- Configuration management (replication, consistency) handling by etcd
- etcd bootstrapping
  - etcd store initiatilization at GD2 boot up
  - Modes – client(proxy)/server
  - Interface to toggle between client to server and vice versa
- External etcd integration too

RPC Framework

- Brick to client (and vice versa) communication over existing sun rpc model
- Protobuf RPC for GD2 to GD2 / daemons communication
- Considerations
  - Language support for both C & Go
  - Auto code generation
  - Support for SSL transports
  - Non-blocking I/O support
  - Programmer friendly implementation pattern (unlike thrift using Glib style)

Transaction framework

- Drives the life cycle of a command including daemons
- Executes actions in a given order
- Modular - plan to make it consumable by other projects
- To be built around central store
- execution on required nodes only
- Originator only commits into the central store

Feature plug-in framework

- Ease of integration of Gluster features
- Reduce burden on maintenance & ownership
- Feature interface aims to provide
  - insert xlators into a volume graph
  - set options on xlators
  - define and create custom volume graphs
  - define and manage daemons
  - create CLI commands
  - hook into existing CLI commands
  - query cluster and volume information
  - associate information with objects (peers, volumes)

Flexi volgen

- Volfile – source of information by which xlator stack is built up
- Currently pretty much static in nature
- Goal - make it easy for devs/users to add/remove xlators
- SystemD-units style approach for the new volgen (Still under discussion)

Other improvements

- Transaction based logging, probably centralized logging too!
- Unit tests
- Better op version management
- Focus on improved documentation

Upgrades consideration

- No rolling upgrade, service disruption is expected
- Smooth upgrade from 3.x to 4.x (migration script)
- Rollback - If upgrade fails, revert back to 3.x, old configuration data shouldn't be wiped off

References

- Design documents - https://github.com/gluster/glusterfs-specs/tree/master/design/GlusterD2
- Source code - https://github.com/gluster/glusterd2
- Reach us @:

  gluster-devel@gluster.org, gluster-users@gluster.org

  #gluster-dev, #gluster on IRC

Q & A

THANK YOU