Report from the field: Samba clustering with GlusterFS

sambaXP 2020

Anoop C S
<anoopcs@redhat.com>

Günther Deschner
<gd@samba.org>

SAMBA

Red Hat

Agenda

Overview

- Samba and GlusterFS
- Red Hat Gluster Storage

Deployment report

- Clustering issues
- Correctness problems
- Performance limitations

Moving forward

- Improvements in queue
- Future goals and SMB3

Samba, GlusterFS and Red Hat Gluster Storage (RHGS)

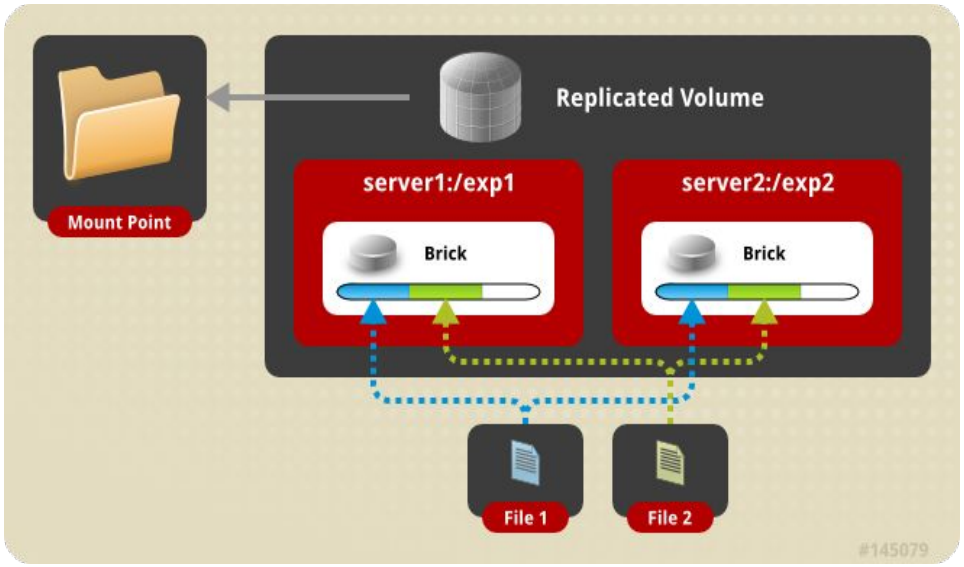
Red Hat Gluster Storage (RHGS)

Gluster

- Gluster is a free and open source software scalable network filesystem
- On-premise, public and private cloud deployments
- Replication, Quota, Snapshots, geo-replication
- Runs on every FS that support extended attributes
- FUSE fs client
- Ansible automation
- Supports variety of access protocols including NFS and SMB

Red Hat Gluster Storage (RHGS)

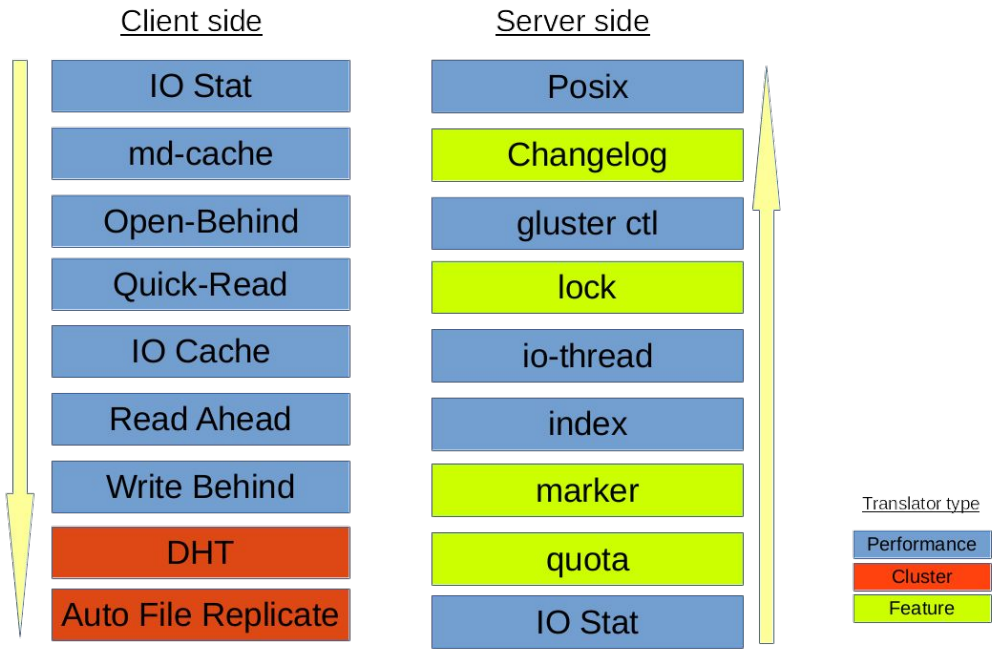
Gluster architecture I



- Bricks are low level components
- Multiple bricks → volumes
- Various different volume types:
  - Distribute
  - Replicate
  - Arbiter
  - Disperse
- volfile configuration

Red Hat Gluster Storage (RHGS)

Gluster architecture II

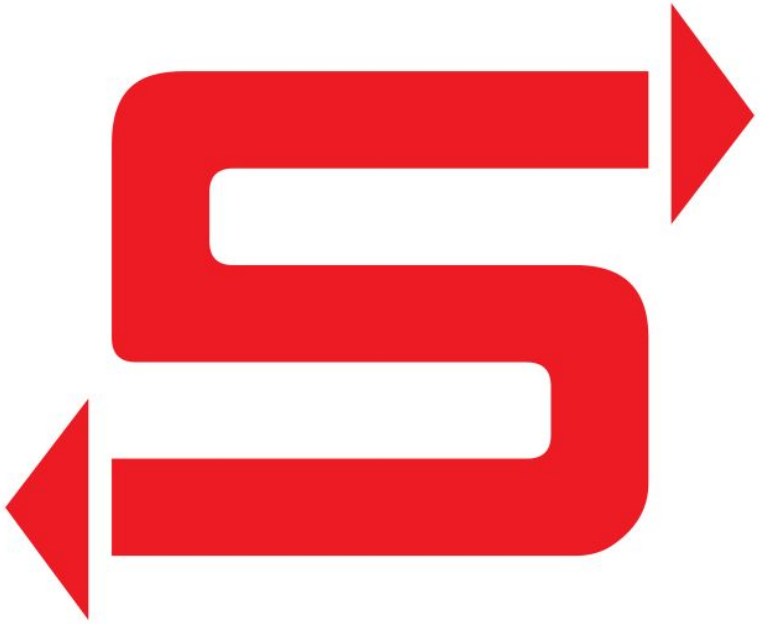


- Client and server translator stack
  (similar to the Samba VFS layer)
- md-cache translator primarily important for Samba
- Performance enhancements specific for Samba:

```
gluster volume set <volume>
performance.cache-samba-metadata on
```

Red Hat Gluster Storage (RHGS)

Samba & CTDB



- Layered product:

  Samba version *always* ahead of RHEL Samba version
- Two options for using gluster:
  - vfs_glusterfs module, consuming gfapi (default)
  - Fuse re-export
- macOS clients using vfs_fruit

  (now fully supported with non-local FS as well)
- CTDB for HA of Samba/Winbind and public IPs
- CTDB uses distinct glusterfs volume for recovery lockfile

Common problems from customer setups

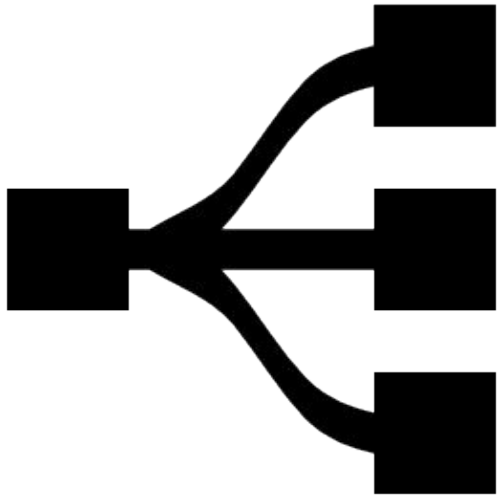
Failure to acquire recovery lock for CTDB

Chances of occurrence: *frequent*

- Node reboot
- Unavailability of common recovery lock
    - Accidental creation of recovery lock file locally
- Availability of shared storage post-reboot
- Automatic mounting of shared volume holding recovery lock(/etc/fstab)
    - Dependency on *glusterd*(GlusterFS daemon)
- CTDB systemd service file modifications

Misconfigured public/private network separation

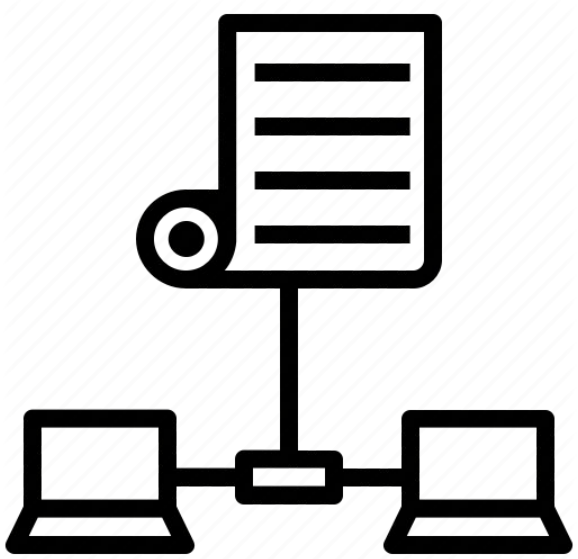
Chances of occurrence: *rare*



- Cable unplugged and node down/reboot scenarios
- Durable handle reconnect
- Separate GlusterFS and CTDB traffic
  - Network teaming?
- Resolving hostnames, if used, to correct network interface
- External client facing IP on a different network

Diverging netbios namespaces on cluster nodes

Chances of occurrence: *rare*



- AD Domain membership:

  machine account credentials in CTDB
  - Only join AD on one node!
  - Not having same “netbios name” set splits common cluster account into individual node accounts
- Standalone setup:

  local SAM is shared via CTDB
  - Not having same workgroup and netbios name on all nodes creates diverging SIDs
  - Result: ACL authorization failures

Incorrect usage of POSIX permissions

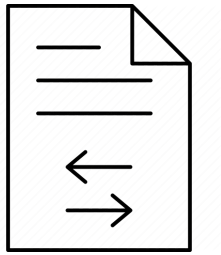
Chances of occurrence: *common*

- Setting up ACLs on share root
  - Enabling *-o acl* mount option of GlusterFS FUSE mount
- Existence of default ACLs on directory
- Using *vfs_acl_xattr*
- Special treatment with *'ignore system acls = yes'*
- Problems with switching same share with and without *vfs_acl_xattr*

Understanding performance bottlenecks

Primary focus areas

Broadly classified into IO and metadata related workloads



- Basically involves read/write operations
- Writes should reach bricks if online
- Previous implementation with libgfapi async APIs
- Limitations
- Using *pthreadpool* infrastructure

Takeaway: *Parallelism and asynchronous nature in scheduling read/write operations*

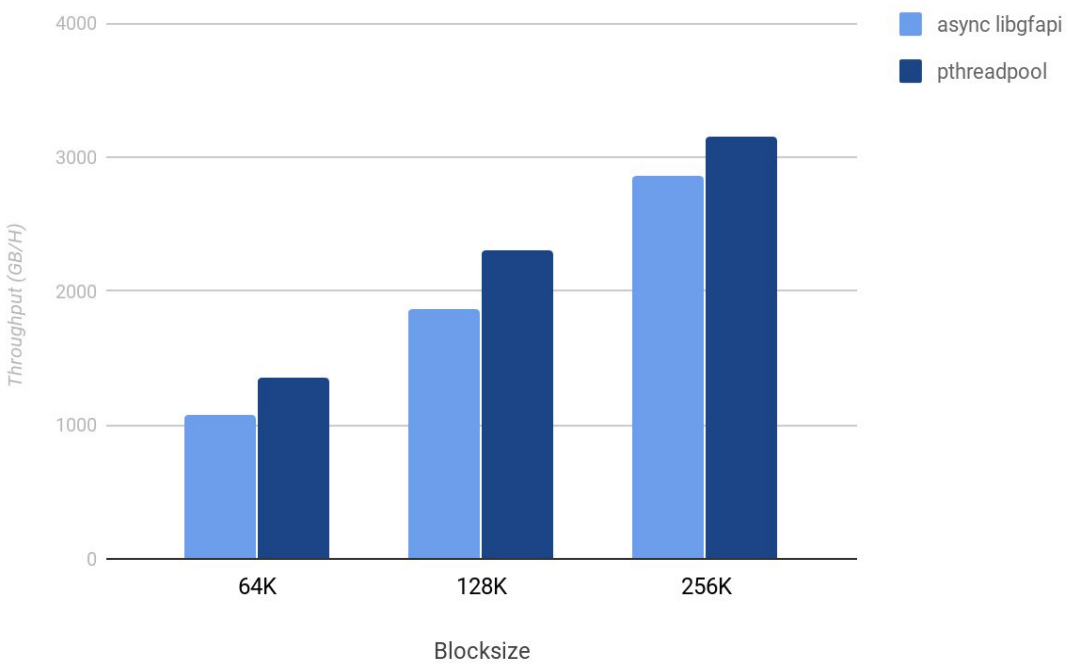
- Mostly *getxattr* and *stat* calls
- Frequent invocations
- Working with a distributed file system underneath
- Improvements with *get_real_filename* and *md-cache* translator

Takeaway: *GlusterFS client side caching for metadata heavy requests*

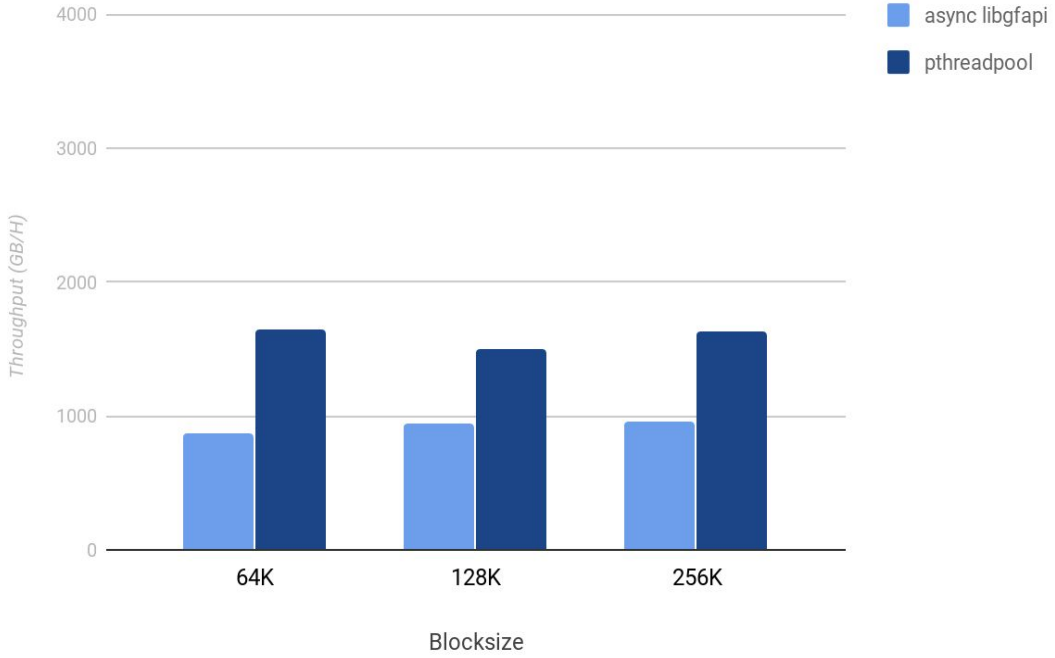
IO benchmarking with Disk Performance tool

Throughput measured for read-write operations

READ



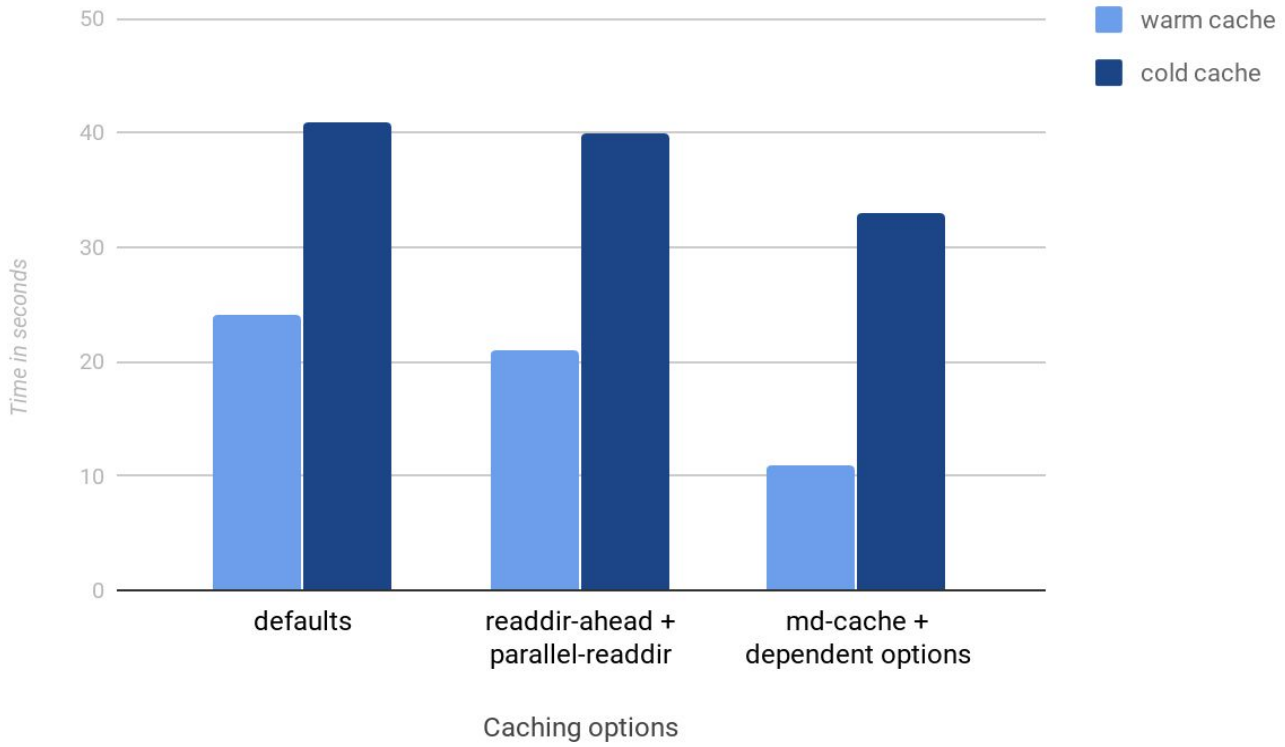
WRITE



Samba on top of 12x(2+1) Arbiter volume spread across 3 nodes with CTDB
Block count = 16K, Thread count = 12, Mode = random

Plain listing of large directories

Delay in displaying contents of a directory with large number of small files



Samba on top of 12x(2+1) Arbiter volume spread across 4 nodes with CTDB
Client = *smbclient*, Number of 1KB files = 16K

Recursive listing of large directory tree

Time taken for file system crawl

- Relatively large directory structure
  - with depth level of 10, 100 etc..
- Proposed solution from GlusterFS
  - implemented within distribution layer
  - prefetching logic to fill readdir buffer
  - *performance.readdir-cache* volume set option
- Native client improvements around 100%
- Samba integration?
  - yet to explore :-) but hopeful

Miscellaneous improvements

New glusterfs_fuse VFS module

Added advantage of *get_real_filename*

- Motivation: absence of get_real_filename previously
- Performance improvement with creation of files
  - Exposed in gluster via xattr: `glusterfs.get_real_filename:<filename>`
- Problems with file_id calculation
  - Difference in Device-Id on FUSE mounts from cluster nodes
- Reuse of existing logic (vfs_fileid)
- New module
  - Last one in the stack, No additional options
  - Easy to use without any overhead
- Still involves FUSE context switches

New VFS interface for *fcntl()*

Directed towards handling of open file descriptor flags

- Motivation: actually a regression seen with GlusterFS
  - O_NONBLOCK set bypassing VFS
- Contact VFS for handling fd flags
- Introduction of SMB_VFS_FCNTL
  - Complexity involved with different types of *fcntl* flags
  - Automatic detection of flags
  - May be the only VFS interface macro with variable arguments !
- Just one caller?
- Consumed by *vfs_glusterfs* with a hack

Ongoing developments around Samba integration

SAMBA

Red Hat

Next steps

General upcoming developments

- Correctness in durable handle reconnect with GlusterFS
- Slow directory listing (large number of entries)
- Proxy mode with vfs_glusterfs/gfapi to limit memory consumption
- SMB3 Multichannel
  - Oplock/lease replay
  - Enabling socket_wrapper for multichannel self test
  - Integration with CTDB
- Witness
  - Dependency on async DCE/RPC server
  - Prototype implementation done in 2015
- Automation/CI

Questions?

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

SAMBA

Red Hat