Efficient Data Tiering in GlusterFS

Mohammed Rafi KC

About me

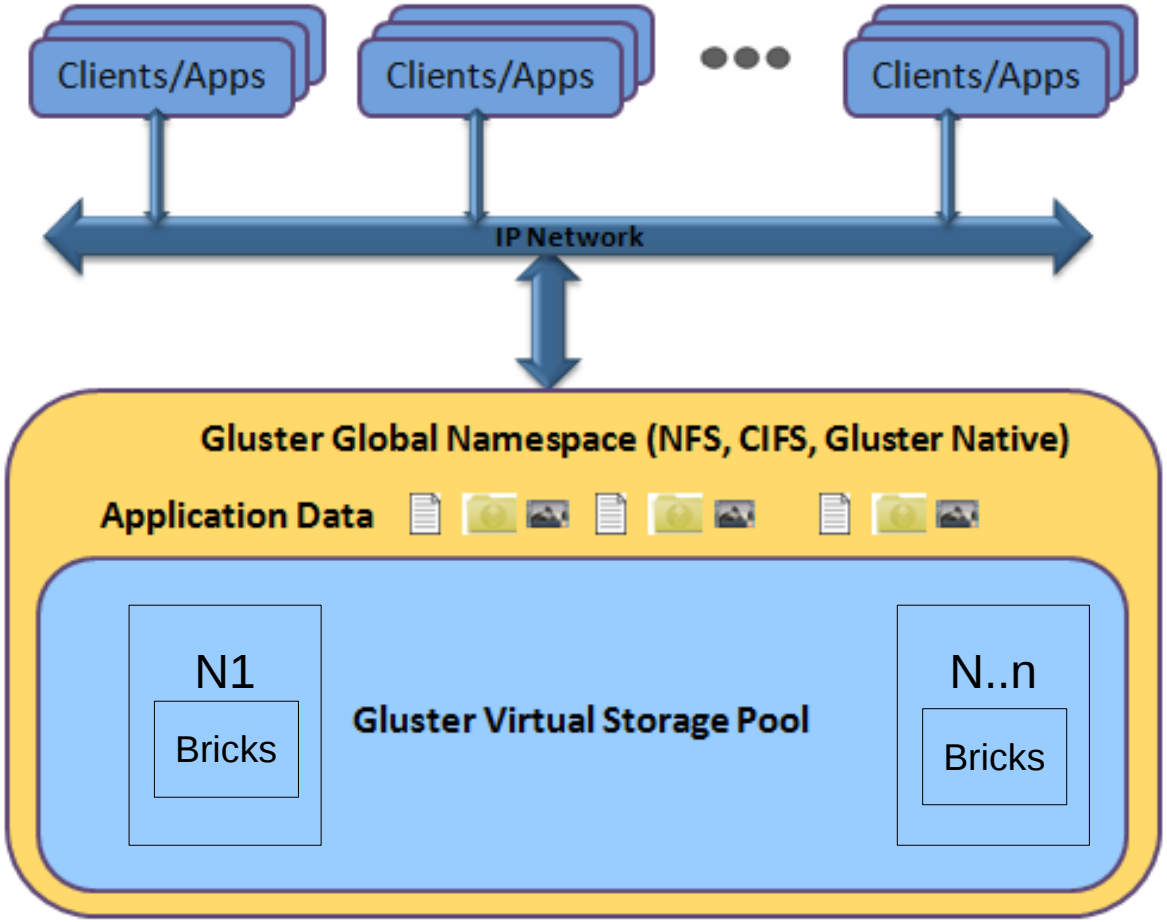
Mohammed Rafi KC (Software Engineer, Red Hat)

- Gluster-rdma, Gluster-snapshot, Gluster-tiering.

Agenda

- Quick GlusterFS Overview
- Data Tiering
- GlusterFS - Data Tiering
- Detailed Implementation
- Challenges

What is GlusterFS



Distributed File System

Software Define NAS

TCP/IP or RDMA

Native Client, SMB, NFS

Automated Data Tiering

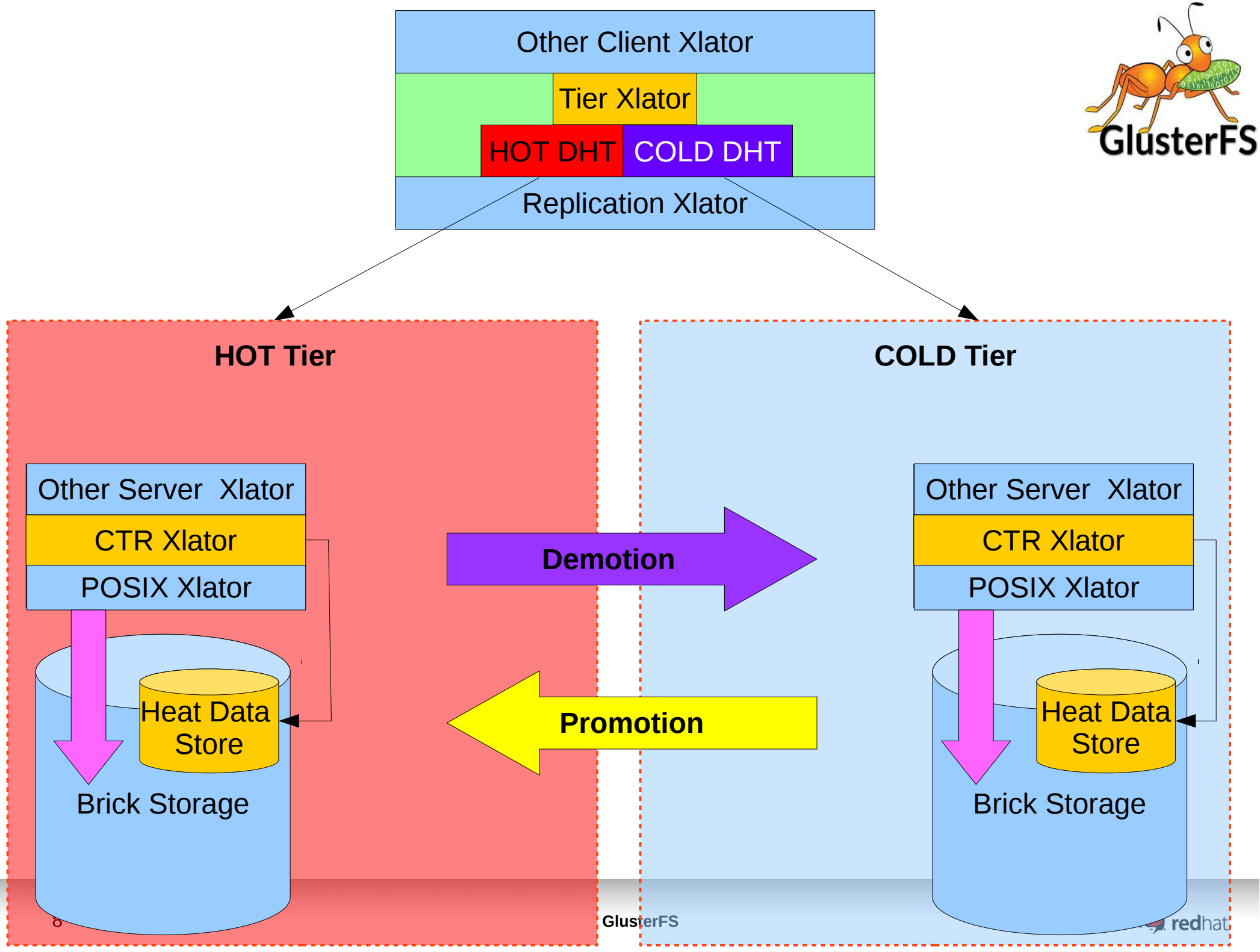
- A logical volume composed of diverse storage units
  - Fast / slow
  - Secure / nonsecure
  - Expired hold time / expired
  - compressed / uncompressed,
  - Cloud expensive elastic storage / cheap
  - Etc.
- Data moves automatically across tieres
- Efficient use of different storage tieres

GlusterFS – Data Tiering

- Two Tiers
  - HOT and COLD
  - Fa$t SSD, slow HDD
  - Fast 2X replicated, slow erasure coded
- Files will be moved across HOT and COLD tiers

Policies for Smart Migration

- File size
- Access rate
- Migration frequency
- Water mark
- Break files into chunks
  - Gluster “sharding” feature


Other Client Xlator
Tier Xlator
HOT DHT
COLD DHT
Replication Xlator
GlusterFS
HOT Tier
COLD Tier
Other Server Xlator
CTR Xlator
POSIX Xlator
Heat Data Store
Brick Storage
Demotion
Promotion
Other Server Xlator
CTR Xlator
POSIX Xlator
Heat Data Store
Brick Storage

Challenges in Data maintenance

Data Maintenance has a overhead on CPU, Memory, Storage, Network.. Therefore..

Fast Search

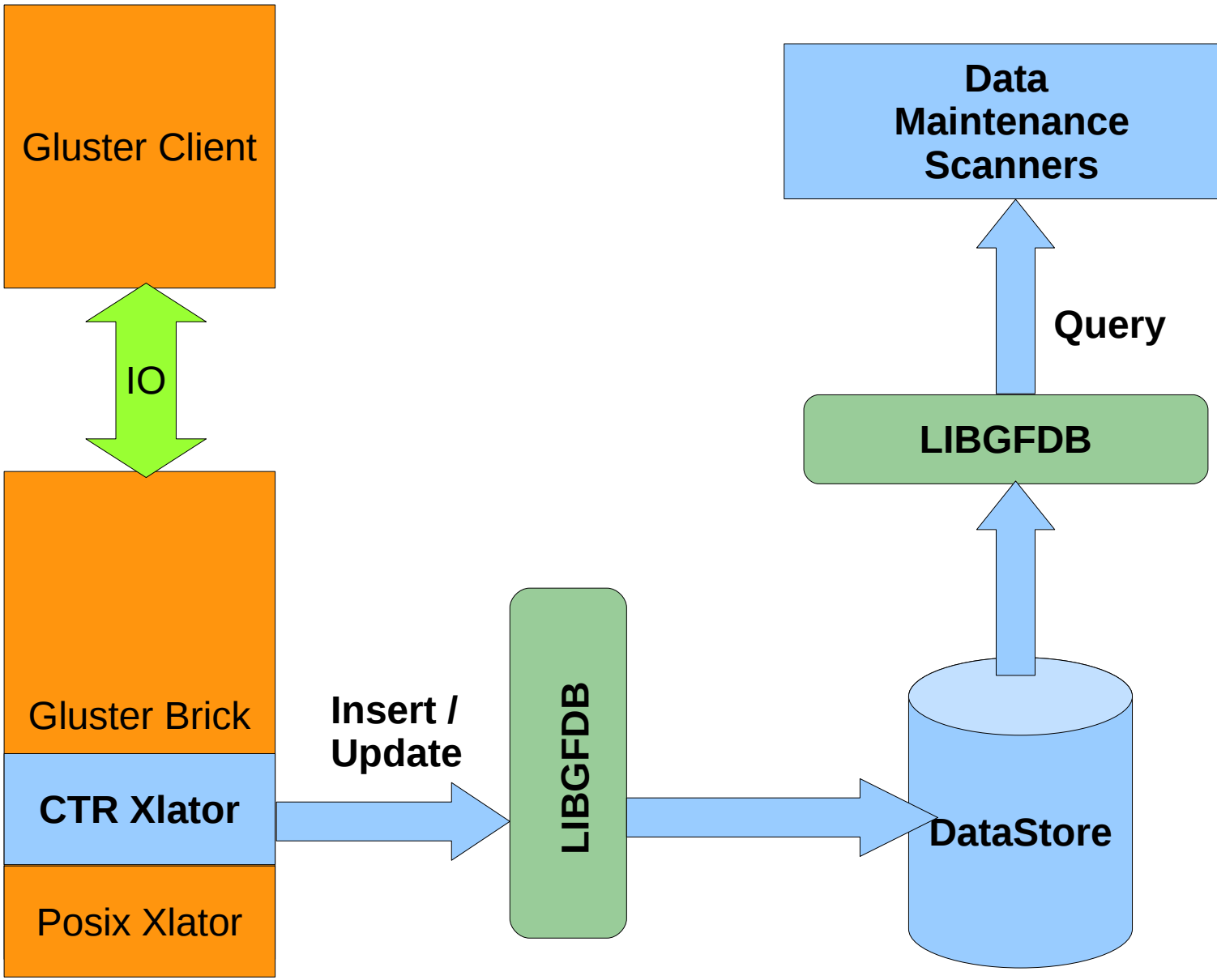
Rich Metadata

Distribute

Load balancing

Implementation

- Change Time Recorder
- Libgfdb
- Migration Daemon


Gluster Client
IO
Gluster Brick
CTR Xlator
Posix Xlator
Insert /
Update
LIBGFDB
DataStore
LIBGFDB
Query
Data
Maintenance
Scanners

LibgfDB

- API Abstraction
- Rich Search Filters
- Performance optimization options

Optimized DB for GlusterFS

"Record now , consume later"

Database optimized to record fast

Good Querying Capabilities

Embedded Database

Datastore Optimization: Sqlite3

- **PRAGMA page_size:** Align page size
- **PRAGMA cache_size:** Increased cache size
- **PRAGMA journal_mode:** Change to WAL
- **PRAGMA wal_autocheckpoint :** Less often autocheck
- **PRAGMA auto_vacuum :** Set to NONE

Lesson Learned :

- DB updates can be expensive
- DB query may have scalability problems
- Durability (ACID semantics) is expensive

Feature Page

http://www.gluster.org/community/documentation/index.php/Features/data-classification

Gluster Github:

https://github.com/gluster/glusterfs

Email:

Mohammed Rafi KC <rkavunga@redhat.com>

THANK YOU

What's next:

- Libgfdb Performance options :
  - iMeTaL : in-Memory Transaction Log
  - PeTal : Persistent Transaction Log
- Sqlite3 Database Sharding

Existing Solutions

- File system crawl
- File system log
- Metadata databases
- In-memory inode caches

DataStore Optimization: Sqlite3