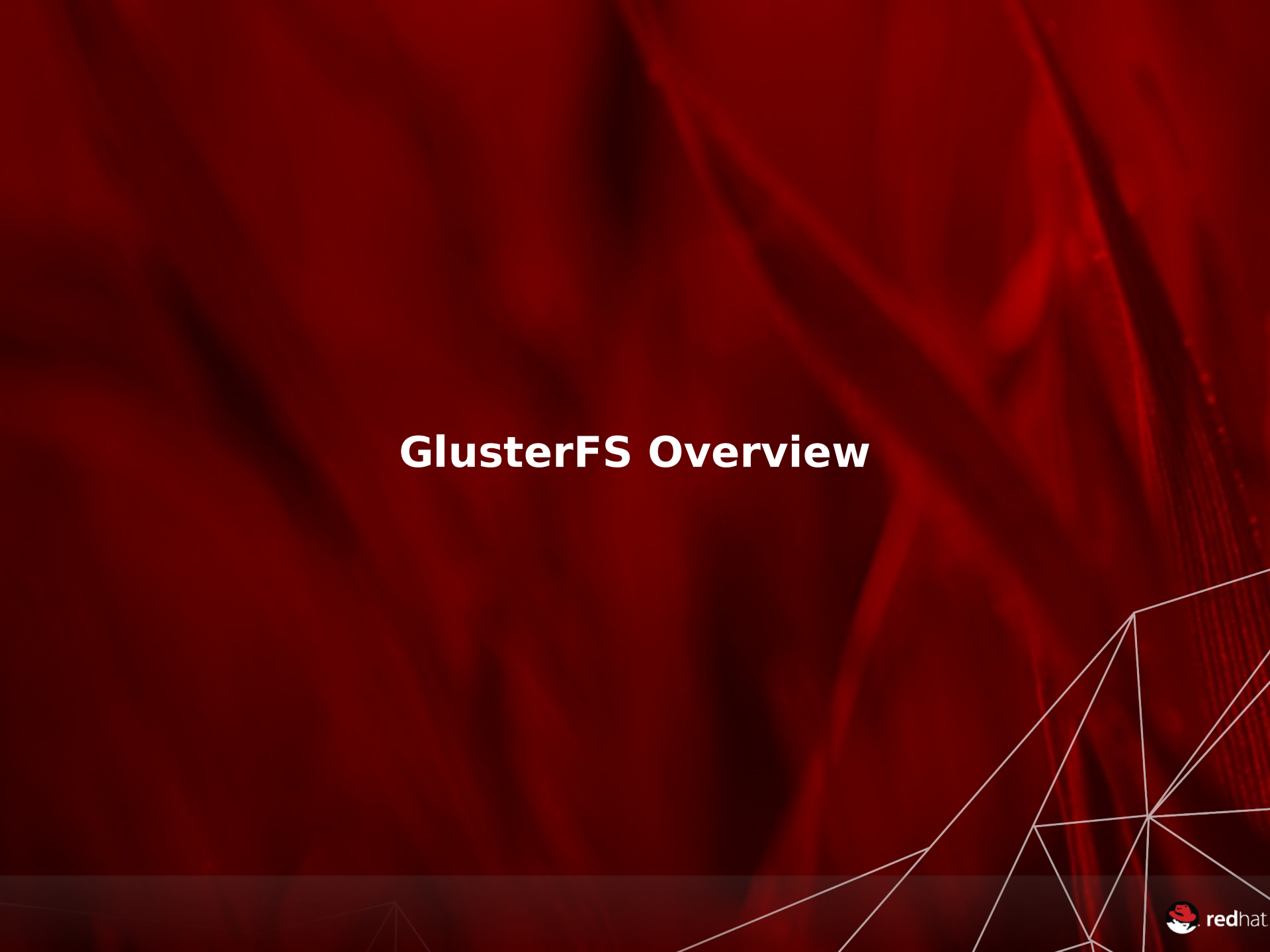
# Multi-Protocol Support in GlusterFS

**Ira Cooper / Poornima G.**
**Red Hat Storage SMB/SMB2 Team**
**September 15, 2014**

# GlusterFS Overview



# What is GlusterFS?

- Software Defined Storage
- “Shared Nothing” Clustered Filesystem
- Optimized for high throughput
- Common use cases:
  - HPC
  - Backup and Archival Storage
  - Virtual Machine Storage

# GlusterFS Architecture

- GlusterFS is basically a stacked VFS.
- Each “layer” is called a translator
- Example translators:
  - Client/Server – Network traversal
  - AFR – Automatic File Replication
  - POSIX – Back end to a POSIX file system
  - DHT – Distributed Hash Table

**Multi-Protocol Support...**

**Why is this so hard? :(**

# Protocols Supported

- SMB/SMB2 – Samba
  - 3.6.9 + Patches
  - Work in progress to support newer versions
- NFS
  - NFSv3 via GlusterFS specific server
  - NFSv4 work in progress with Ganesha
- FUSE
  - Linux
  - FreeBSD
  - OS X

# NFS / FUSE Semantics

- Case sensitive
- Byte Range Locking is discretionary
- File Locking is discretionary
- Namespace operations have no locks
- Delegations / Layouts in NFSv4
  - Directory delegations in NFSv4.1+
  - “5 second rule” for NFSv3/FUSE

# SMB/SMB2 Semantics

- Case Insensitive
- Byte Range Locking is enforced
- “Share Modes” exist, and are enforced
- Oplocks / Leases
- Directory Leases / Notify

# Design

# Philosophy

- SMB Driven, mainly
  - Large differences in semantics
  - Not the way it is usually done
- FUSE + NFS kind of works already

# Imperfection is Expected

- SMB, SMB2, FUSE, NFSv3, and NFSv4.x can not be perfectly mixed
  - This has been discovered through years of experience
- The goal is to give an acceptable user experience
  - No data loss/corruption
  - Minimal to no mishandled permissions
  - Semantics that are as good as possible
  - Allow the user to make tradeoffs.

# General Design Principles

- State recovery as in SMB2+ will be built into the file system
- State replication will be assisted by replication / erasure coding translators
- Everything that follows is a translator or set of them
- Support will be added to Ganesha, Samba and FUSE for these new concepts

# ACLs

- We've decided to accept the Rich ACL proposal
- NFSv4 ACLs and SMB look the same
  - NFSv4.1 was supposed to help here
- SIDs will not be stored in the filesystem
- Support for NFSv3 must be maintained
- Support for POSIX ACLs may be needed for FUSE and NFSv3

# Locking

- Byte Range
- Share Modes
- Oplocks, Leases and Delegations

# Byte Range Locking

- Two different types of locks
- POSIX
  - Discretionary
  - Range merge and split
  - Tend not to be trusted over NFS
- SMB/SMB2
  - Mandatory
  - Do not range merge or split
  - Tend to be used/abused

# Byte Range Locking (2)

- Separate state for mandatory and discretionary locks
- Kept in separate translators
  - Makes the semantics clearer, and easier to enforce
  - The locks will not interfere with each other
- Locking enforcement rules will be tunable

# Share Modes/Reservations

- SMB/NFSv4 feature
- Permission on open to “allow/deny” others
  - To read
  - To write
  - To delete
- POSIX explicitly has no such concept

# Goplocks

- **G**lusterFS **Oplocks**
- Mixes SMB, SMB2, and NFSv4 semantics, for opportunistic locking
- Requirements:
  - Reader, Writer, Handle, Once, None, file based caching
  - Callbacks are registered by clients

# Future Directions

- Directory Goplocks
  - Will be need for full metadata consistency
- Improved ACLs
- Better atomicity for open/create
- Disconnected clients, dead clients, and other issues will have to be worked through

# Questions?

# Thanks for Attending!