HANDLING PERSISTENT PROBLEMS: PERSISTENT HANDLES IN SAMBA

Ira Cooper
Tech Lead / Red Hat Storage SMB Team
May 20, 2015 – SambaXP
Who am I?

- Samba Team Member – SMB2/SMB3 focused.
- Tech Lead – Red Hat Storage SMB Team
  - Focus on Samba Integration with Linux technologies
    - Gluster
    - Ceph
    - XFS
    - Rich ACLs
    - To name a few…
What is a Persistent Handle?

- A Durable Handle with extra guarantees.
  - How long it is valid for.
  - Where it is valid.
- Connected to the witness protocol.
  - Tells clients where the handles are valid.
  - When failures occur – Replaces tickle ACKS?
    - Really, they compliment each other.
Why Persistent Handles?

- It enables CA – Continuous Availability.
  - Allow re-connection after server/network failure.
  - Allows clients to take full advantage of SMB3 features.
- Applications that require CA:
  - HyperV – Allows .vhd files to be served over SMB3.
  - MSSQL – Allows for reliable operation of databases over SMB3.
- Applications that benefit from CA:
  - Long running batch jobs.
  - Standard user workloads.
General Design Thoughts.

- Two major designs.
  - Clustered filesystem.
    - Gluster.
    - Ceph.
    - GPFS.
  - Shared storage, unclustered.
    - SAS.
    - Fiber Channel.
- Our focus today is on Clustered.
- But we should look at shared storage!
What Do We Need To Do?

- Save and Enforce the Handle State.
  - Byte Range Locks.
  - Leases / Oplocks.
  - Share Modes.
  - Persistent Handle ID → File Name.
  - Client/Connection GUID?

- Where?
  - In CTDB?
  - In the filesystem?
  - On the filesystem?
On the Filesystem or Shared Storage.

- Currently no major work in this area.

- Low complexity.

- Low Risk?
  - Will TDB be safe enough for this?

- Performance could be quite good.
CTDB and Persistent Handles.

- Persistent Handles guarantees run counter to CTDB.
  - CTDB is designed to have weak consistency
    - This is GREAT for performance.
    - Not so good for guaranteeing cross node persistence.

- RAFT should help.
  - Better durability guarantees.
  - More organized.

- What failures should we survive?
In the Filesystem.

- This is the current choice we made in Gluster.
  - Also used by Red Hat Gluster Storage.

- Persistent Handle recovery is assisted by the filesystem.

- This requires heavy support from the filesystem!

- Not portable to other filesystems.
Why In the Filesystem?

- Interoperability.
  - Samba doesn't have an NFS server.
  - Or iSCSI.
  - Or ftp
  - Or FUSE
  - Or....

- Regardless of what we think, the world is more than SMB.
Persistent Handle in the Filesystem?

- It depends on the design of your filesystem.

- For this presentation, we'll discuss Gluster.
  - Similar problems will be faced in any system.
  - Design of the solution will vary greatly.

- But Gluster is a great example!
Gluster Basics.

- Clean, extensible architecture.
- Design is a giant stacked VFS.
- Everything is a module aka translator (xlator).
  - Even network communication.
  - Replication.
  - POSIX Locking.
  - Read Caching.
  - Write Caching.
  - Etc…
Gluster + Samba

“Real Client”

Win8

Samba

Samba

Samba

Gluster Client

Gluster “Brick”

XFS

Gluster “Brick”

XFS

Gluster “Brick”

XFS

POSIX
Gluster “xlator stack” Client (example)
Gluster “xlator stack” Server (example)
New Xlators for SMB3 and Multiprotocol.

- Upcall dedupe.
- Lease-lk.
  - Leasing semantics.
- Sharemode.
  - Sharemodes.
- Mandlock – Mandatory Locking.
  - Byte Range Locks, SMB Style.
Lease-Ik design.

- Lease-Ik has callbacks from the filesystem to userland.
  - We can get notified on a Lease break, and ack it now!
  - This is an unusual feature.
    - Unique even?
  - Allows “users” to make sure their leases are honored.
  - Enforced by the filesystem.
  - Allows for multiprotocol designs.
Sharemodes design.

- Enforces sharemodes, on all opens/accesses.

- Allows “users” to make sure their locks sharemodes work.
  - As best as possible. Some semantics can't be enforced.
  - Can't always block open.

- Enforced by the filesystem.

- Allows for multiprotocol designs.
Mandatory Locking design.

- Currently being debated if it is separate from POSIX.
- True SMB Locking semantics will be enforced.
  - There will be tunables to tune it down.
- Issues like stacking locks, etc are handled properly.
- Allows “users” to make sure their locks are honored.
- Enforced by the filesystem.
- Allows for multiprotocol designs.
Problems?

- Who is the client?
  - Client xlator?
  - SMB client?
- Both are right!
  - Need to record info about both.
  - “Remote” client GUID.
- Need to time out data after a “client” disconnects.
  - Both SMB and Gluster.
Neat Things!

- NFSv4 and SMB3 share technology!
  - SMB3 – Persistent Handles.
  - NFSv4 – Grace.
  - Share Modes.
  - Leases or Delegations.

- Truly multiprotocol and SMB3 design, from day 1.
Notes.

- Do not think names, and exact details are accurate.
- We are working on the VFS changes needed for Samba.

- The overall plan is accurate!

- The architecture you learned IS accurate roughly.

- I encourage you to go dig into Gluster!
Questions?
Thanks for Attending!