# The CTDB Report

Martin Schwenke
<martin@meltin.net>

Samba Team · DDN

SambaXP 2025

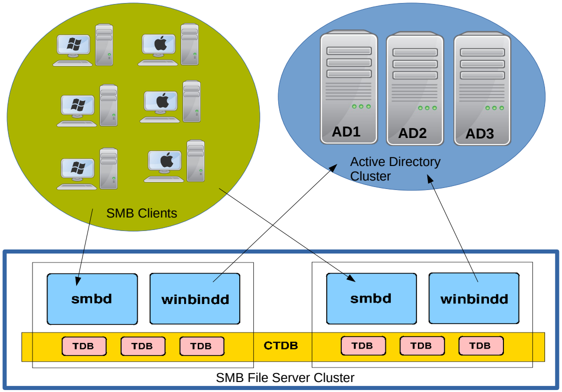
## Overview

## Audience

Everyone!

. . . a development-focused talk, but not just for developers

## Clustered Samba



## What is CTDB?

- Clustered database for Samba metadata
  - Distributed, volatile TDBs
  - Replicated, persistent TDBs
- Cluster-wide messaging transport
- Cluster management — leadership, membership
- Dynamic IP address failover
- Service management (`smbd`, `winbindd`, NFS, ...)

## Progress

## Authors

| | |
|---|---|
| 251 | Martin Schwenke |
| 21 | Volker Lendecke |
| 16 | Vinit Agnihotri |
| 14 | John Mulligan |
| 14 | Andreas Schneider |
| 13 | Stefan Metzmacher |
| 9 | Jennifer Sutton |
| 3 | Michael Tokarev |
| 2 | yogita72 |
| 2 | Günther Deschner |
| 2 | Anoop C S |
| 347 | |

## Reviewers

| | |
|---|---|
| 139 | Amitay Isaacs |
| 79 | Volker Lendecke |
| 65 | Martin Schwenke |
| 37 | Anoop C S |
| 20 | Jerry Heyman |
| 12 | Andrew Bartlett |
| 12 | Andreas Schneider |
| 12 | Günther Deschner |
| 9 | John Mulligan |
| 8 | Ralph Böhme |
| 5 | Stefan Metzmacher |
| 3 | Noel Power |
| 2 | Jeremy Allison |
| 2 | Douglas Bagnall |
| 2 | David Disseldorp |

## Main areas

- Improve cluster lock reliability
- Add ability to read nodes list from command
- Improve NFS lock recovery/reclaim
- Script cleanups - `shfmt`, `shellcheck`, ...
- Test additions and improvements

## Cluster lock reliability

- Standard helper is `ctdb_mutex_fcntl_helper`
- It takes a lock in a specified file in cluster filesystem
- Add child process to do some sanity checks:
  - Recheck lock file by taking another lock at a different offset
  - Check that inode number of lock file hasn't changed
- Main process and child ping each other to ensure that neither is wedged

## Read nodes list from command

- Historically `/etc/ctdb/nodes` (for example)
- The Samba in Kubernetes project needs more flexibility
- So, allow a command to generate a notes list
- John Mulligan
- TODO: Fix issue where blocking command blocks `ctdbd`

## NFSv3 lock recovery/reclaim

- Drop `statd-callout` re-invocation via `sudo`
    - Public address cache for scripts
- Split `statd-callout`: fast C program + helper script
    - `rpc.statd` is single threaded, calls HA callout synchronously
- Allow `statd-callout` to write client state to shared filesystem storage
    - Optional revert of 2010 move to persistent TDB?
    - Persistent TDB is still default
    - Persistent TDB dequeuing done in `monitor` event — lossy
- Use NFS-utils `sm-notify`
    - ...instead of home-grown, IPv4-only `smnotify`

## NFSv4.x lock recovery/reclaim

- Add `startipreallocate` event to force *all* nodes into grace at start of failover (Vinit Agnihotri)
- Drop `06.nfs` event script and associated `releaseip-pre`, `takeip-pre` pseudo-events (Vinit)
- `nfs-ganesha-callout`
  - Use `startipreallocate` (Vinit)
  - Avoid failing to enter grace period during startup
  - Support Lustre filesystem (WIP)
  - Support grace timeout for those who don't think they care. . . (WIP)
  - Rewrite in Python (WIP — Peter Schwenke)
  - Support IP-based recovery (WIP, also add NFS-Ganesha support for `recovery_fs` — Peter)
  - `ctdb/doc/examples/...` → supported, soon?

## Miscellaneous

- Improved, non-crashy handling of connection tracking for SMB multichannel (Stefan Metzmacher)
- Improve NFS monitoring: RPC timeout sanity checking against NFS statistics (implemented for NFS-Ganesha)
- Track connections to public IP addresses on all ports
  - Not just TCP port 2049
  - e.g. NFSv3 `lockmgr`, `sftp`
- Optionally kill server end of connections using `ss -K`
  - Can replace troublesome `ctdb_killtcp`
- Improve `ctdb_mutex_ceph_rados_helper` (Günther Deschner, John Mulligan)
- Add `UNKNOWN` node status (queued in 2022 — Vinit& I)

## Queue

## I have a lot of WIP branches

| | |
|---|---|
| ctdb-queued-db | Transactions are too slow, so this queues updates and dequeues them on a timer |
| ctdb-conntrack | Separate connection tracking, needs rewriting with `ctdb-queued-db` |
| ctdb-nftables | Update our failover-time firewalling to use `nftables` |
| ctdb-prio | Use elevated nice setting instead of real-time scheduling |

## Plans

## Failover on graceful shutdown

- NFS does not go into grace on graceful shutdown
- Node that is shutting down just releases IPs
- Either:
  - Run `startipreallocate` — doesn't help SMB
  - Do a whole "takeover run"
- Add an optional `shutdown timeout` option?
- Sorry Ralph...
  - Ralph Böhme proposed this in a MR 4 years ago
  - He implemented it in `ctdbd_wrapper` by optionally running `ctdb disable` on shutdown
  - ...but we removed `ctdbd_wrapper`
  - Seemed too hard any other way...

## ctdb-processd

- CTDB starts a lot of processes
  - Long-lived: daemons, cluster lock helper, . . .
  - Short lived: TDB lock helper, event scripts, node list command, . . .
- Centralise in a daemon to get common handling of:
  - Output
  - Exit code
  - Logging
  - Timeouts
  - Process monitoring
- Small enough to avoid `vfork` + `execv`?
- REST API over Unix domain socket?
- JSON results?
- Not `systemd` `:-)`

## How about that redesign/rewrite?

- Use (my version of?) Amitay Isaac's 2019 `ctdb-transportd`
- Write some transport tests
- Merge my `ctdb-tunnel` branch to tunnel (most) new protocols over the legacy protocol
- It looks like it should be easy to parameterise `transport_api` so it can use either `lib/messaging` or legacy tunnelling
- Switch the internal protocol (e.g. uptime, ping, ...) to JSON
- Add Python bindings for the transport API
- Write components in Python with JSON protocols: `ctdb-clusterd`, `ctdb-failoverd`, . . .
- Restructure the `ctdb` CLI to match our components
- Rewrite CLI in Python?
- Machine readable format is usually just JSON wire format?

Questions?