STORAGE DEVELOPER CONFERENCE
SDC 23
BY Developers FOR Developers

net use //samba/cloud: Scaling Samba

Ralph Böhme, Samba Team, SerNet
2023-09-20

Outline

Scaling Samba in the Cloud

Distributed DB: ctdb or what?

Benchmarks

Results and Conclusions

Q&A

Scaling Samba in the Cloud

Samba in the Cloud

Dreaming of a scalable Samba Cloud SMB server

- Highly scalable Opensource Cloud SMB with Samba
  - hundreds of nodes
  - hundreds of thousands of clients
- Elasticity: adding/removing nodes must be cheap
- Availability: multi-datacenter, multi-region support
- Migrate data to the cloud while keeping applications working

The problem

- Samba's `ctdb` has consistency, scalability and elasticity limitations
  - data is not replicated, required for SMB3 Persistent Handles
  - usecase is high-performance NAS in a single DC
  - Suitable for cloud SMB at scale?
- Real world scalability: production max 16 nodes, 50k clients
- Elasticity: changing node count causes an expansive database redistribution

Building Blocks

Building blocks of a scalable Samba Cloud SMB server

- Clustered Filesystem
  - CephFS, GPFS, GlusterFS, Lustre, S3, . . . ?
- Distributed Database for session and handle state
  - `ctdb`, . . . ?
- This time we only look at the database component

Distributed DB: ctdb or what?

ctdb

ctdb **features**

- fast, fast, fast
- I've seen 66k ops/s in the open/close benchmark

ctdb **limitations**

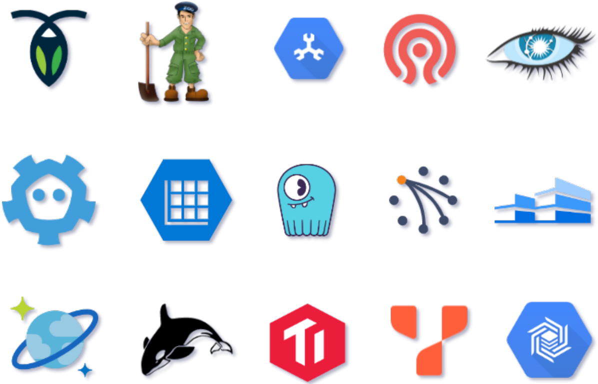
- ctdb has consistency, scalability and elasticity limitations
  - data is not replicated, required for SMB3 Persistent Handles
  - usecase is high-performance NAS in a single DC
  - not suitable for cloud SMB at scale
- Real world scalability: production max 16 nodes, 50k clients
- Elasticity: changing node count causes an expansive database redistribution
- Availability: no support for mulit-DC and AZ

Fix ctdb? Are there alternatives?

- There are many scalable Open Source distributed databases out there
- Can any of those fit the bill?

STORAGE DEVELOPER CONFERENCE SDC 23 BY Developers FOR Developers

Zoo of Distributed Databases



CockroachDB, Zookeeper, Google Spanner, Ceph, Cassandra
etcd, Azure Table, Scylla, Riak, FoundationDB

Azure CosmosDB, Apache Hbase, TiKV, Yugabyte, Google Bigtable

Requirements: Consistency

Consistency

- Samba needs a database with strong consistency quarantees
  - for a non-transactional key/value store this means linearizability
  - for a transactional databases this means strict serializability
- low the database behaves like a single copy and all operation appear in real-time order

Distributed Locking

- locking is needed to serialize and isolate access to two resources: filesystem and database with file-handle state
- To implement locking we need either: transactions or atomic compare-and-set

Requirements: Performance

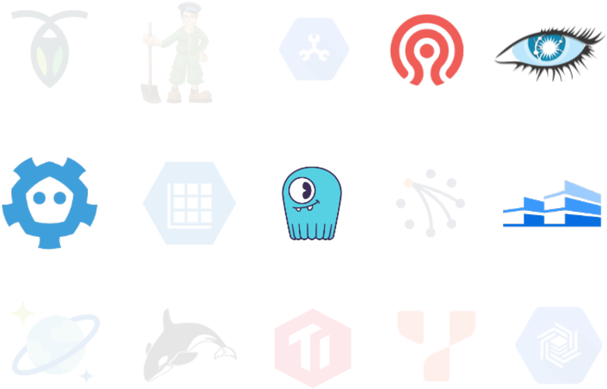
Performance

- Due to its non-replicating design `ctdb` has a very high throughput and low latency
- For many workloads low latency is not first priority:
  - remote office collaboration opening an .doc file:
    takes 200 ms longer to open? Probably doesn't matter!

Assume SMB workload with mostly non-concurrent file access

- the resulting database access pattern is also non-concurrent
- this allows good horizontal scalability

Distributed Databases Tested



dbwrap_py

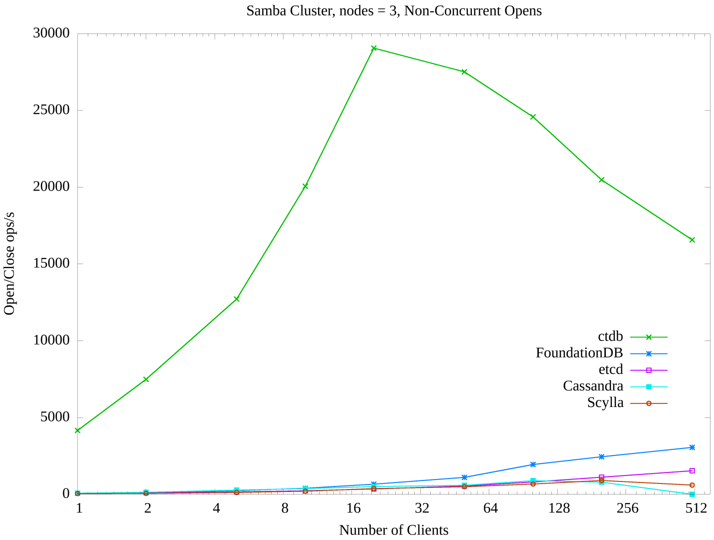
dbwrap_py

- Simplify database adapter development: use Python
- Just 1000 lines of C code (without txn support)
- Using Python for the backend allows rapid prototyping and testing

```
$ wc -l python/samba/samba3/dbwrap_py_*
  338 python/samba/samba3/dbwrap_py_cassandra.py
  414 python/samba/samba3/dbwrap_py_etcd3.py
  303 python/samba/samba3/dbwrap_py_fdb.py
   47 python/samba/samba3/dbwrap_py_tdb.py
```

Benchmarks

Performance: initial evaluation at SambaXP 2023



FoundationDB

And the winner was: FoundationDB

- Key-value store with transaction support
- Supposed to be highly scalable, used by Apple and Snowflake for cloud workloads
- One of the very few open source distributed DBs with a C client
- . . . and Python which we're using for rapid prototyping

Testing with larger FoundationDB cluster

- At SambaXP 2023 I concluded with "we need tests on larger clusters"
- Here are the results. . .

Benchmark

Before the results: how did we test?

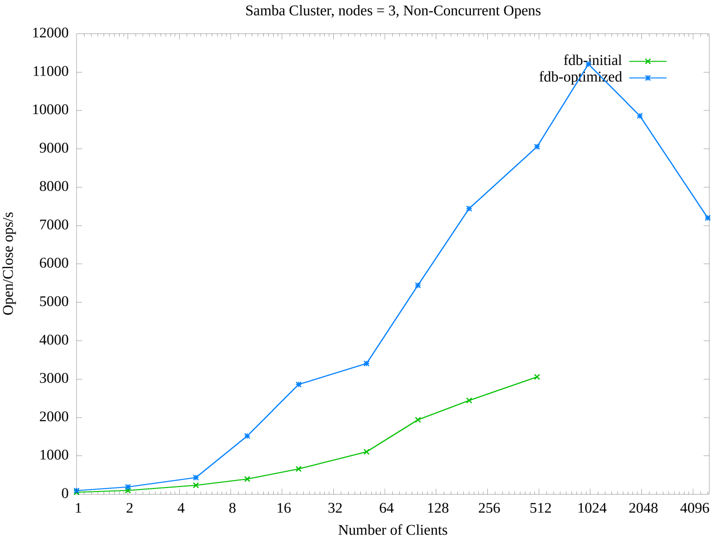
Server: deploy with terraform

- Deploy Samba ctdb cluster and FoundationDB cluster in Azure with terraform
- 1 client VM, 3 ctdb nodes, 1-32 FoundationDB nodes
- thanks to Jule Anger for working on the terraform tooling!

Client: open/close in a loop

```
$ smbtorture //172.18.111.10/test -U slow%x \
  smb2.bench.path-contention-shared \
  --unclist unclist-test.txt \
  --option=torture:timelimit=10 \
  --option=torture:nprocs=[1-500]
```

Performance: ctdb vs FoundationDB



Results and Conclusions

Results

Mixed bag. . .

- 44,000 write (!) txn/s on cloud VMs (`Azure Standard_D3_v2`) with IOPS capped disks
- But it still achieves only 10% max throughout compared to `ctdb`

Conclusions

And the winner is. . .

- FoundationDB for performance and features
- Simple PAXOS or RAFT based databases do not scale well
- Databases which avoid a leader bottleneck scale better
- FoundationDB scales significantly better then any other tested DB

Or write our own?

- Writing a scalable distributed database is very hard
- Single shard PAXOS and RAFT are trivially implemented but do not scale
- using a consensus group per shard solves this but:
  - now you need consensus key ranges
  - changing the ranges when adding or removing nodes becomes a hard problem
- TiKV does this, so it's doable
  (unfortunately TiKV has neither C nor Python bindings)
- Advanced features like multi DC / AZ support doesn't make it easier. . .

Outlook

Outlook

- Highly anticipating the release of Apache Cassandra 5.0
- Cassandra is kind of the Open Source industry standard for eventually consistent databases
- 5.0 ships with strong consistency based on a new consensus protocol ACCORD

QUIC support becomes more important

- Candidate library to add QUIC support to Samba: https://github.com/litespeedtech/lsquic/

Summary

tl;dr

- Still a lot to investigate
- Interested? Join the effort!

Q&A

Q&A

Thank you!
Questions?

Ralph Böhme
slow@samba.org
rb@sernet.de