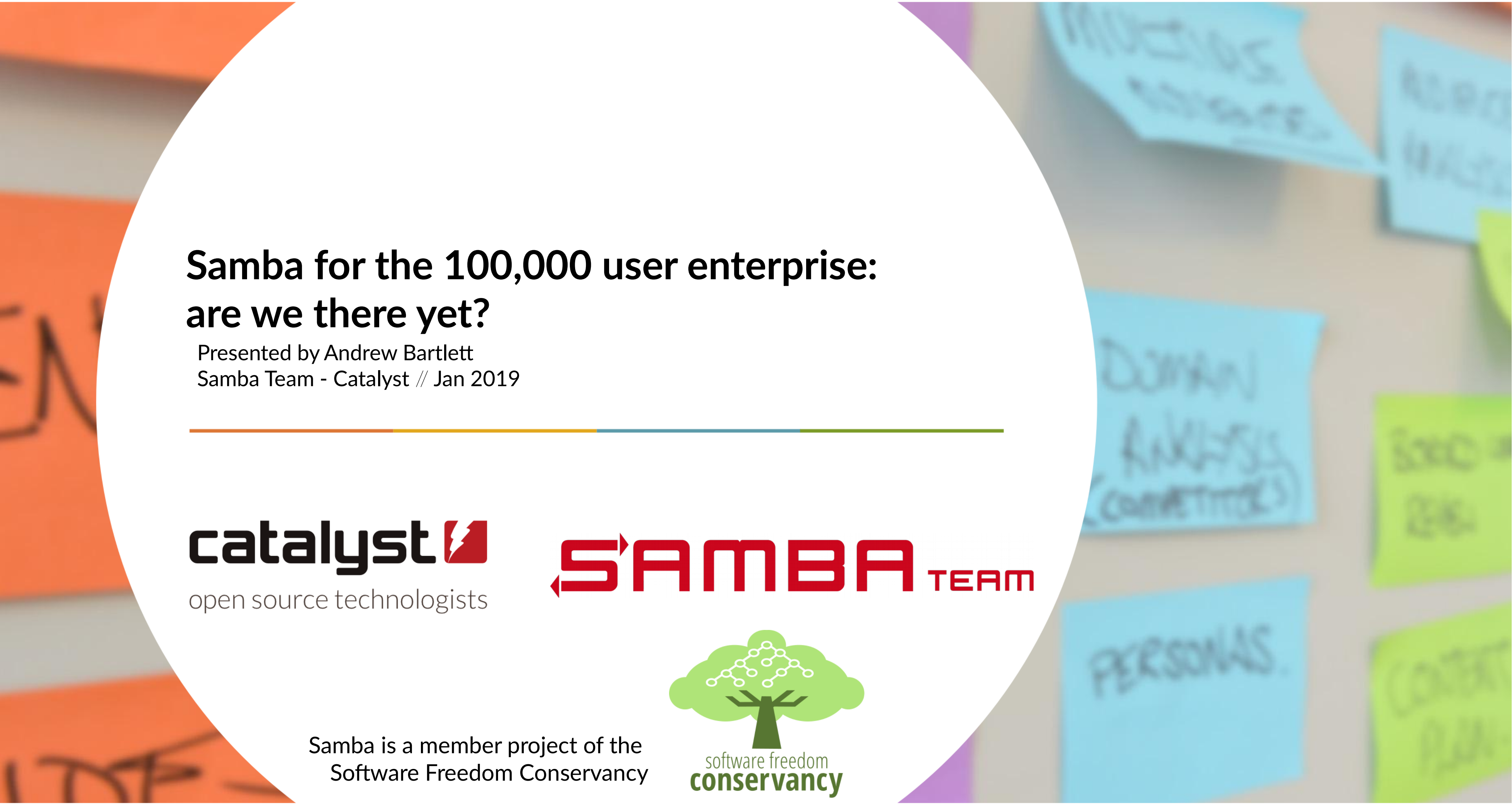


Samba for the 100,000 user enterprise: are we there yet?

Presented by Andrew Bartlett
Samba Team - Catalyst // Jan 2019

catalyst
open source technologists

SAMBA TEAM

Samba is a member project of the
Software Freedom Conservancy

software freedom conservancy

Andrew Bartlett and Catalyst's Samba Team

- Samba Developer since 2001
- Based in Wellington, New Zealand
- Team lead for the Catalyst Samba Team, including:

- **Garming Sam**
- **Douglas Bagnall**
- **Gary Lockyer**
- **Tim Beale**
- Joe Guo
- Aarron Haslett
- PM: Alessandro Dal Sasso

#movingtogitlab: Samba now developed on GitLab

- At least in part
- We tried GitHub for better contributor engagement
  - But Samba Team members didn't use it
  - Became the tool people were told not to use
- GitLab is Open Core
  - And we use gitlab.com, so the 'enterprise edition'
  - We can export to the Open Source GitLab CE however
- But regardless, we finally have uptake!

GitLab CI

- Samba’s full test suite run in parallel
- Split between:
  - free servers from gitlab.com
  - Samba Team servers with Rackspace (on a credit)
- CI was a big factor
  - showing tests would pass before final submission
- GitHub mirror now points contributors to GitLab
- Final code build and merge is still on samba.org hardware

Python 3 support in Samba 4.10

- Samba 4.10 will build with Python3 by default
  - Previous versions had partial support in some bindings (eg used by FreeIPA)
  - Challenge will be killing the python2 support!
- RHEL 8 will finally have python3
  - But RHEL7 has only python2.7…
- Python 2.7 will still be available for building a pure file server
  - Samba 4.10 AD DC still supports Python 2.6 and 2.7

The goal

- For an important customer
  - 120,000 user domain
  - 100,000 computer accounts
- But also for everyone else
  - I hear causally of a 50,000 user domain in production at an Itallian University
  - If only they knew what pain they would have had only a few versions ago!
- Auditing tools and logging are important for everyone

Supporting more connections on each DC

- Samba 4.6 removes single-process restrictions on NETLOGON
  - Really important for 802.1x backed authentication
- Samba 4.7 supports a multi-process LDAP server
  - Actually reduces number of connections you can fit in memory (oops)
- Samba 4.8 adds a prefork mode for LDAP
  - Great for a big AD DC with many, many clients
- Samba 4.10: prefork for more services:
  - NETLOGON, KDC
- Samba 4.10: Process limits for (standard) per-connection forking model

Audit Logging

- Authentication and authorization (4.7)
- Database changes (4.9 and 4.10)
- Human readable and JSON
  - Turns out we should have just used JSON

Fine grained password policy

- Allow some users and groups to have different policies
- Previously it was one policy for the whole domain

Backup tools

- Replaces unlocked backup of the DB with a shell script
- Online and offline backup
  - Offline is locked read of the raw DB records
  - Online is DRS replication and SMB download of sysvol (Group Policies etc)
- Restore tool to recreate the domain
  - Authoritative restore (re-create the first DC)
- If you still have a working DC, just join another DC, don't restore!

Lab domain creation tools

- Able to rename the domain so you can put it in the lab
  - Avoids requiring the creation of a layer-2 isolated network
- Create a realistic preproduction domain!

AD DC Operation at scale

- Practical operation at scale and under load
- Traffic replay tool improved
  - Now can pre-create a 'realistic' DB
  - Able to simulate much more traffic
  - Also operates against Windows

Replication at scale

- Scale is not just filling the database
- Helps if you can actually create the second replica!
- Found while trying to build a large network in our lab
- Lots of small but practical fixes made a massive difference
  - Group memberships were slow at > 10,000 group memberships
  - 10 hours down to half and hour

Inter-forest trusts

- A continuing project
  - Principally by Stefan Metzmacher of SerNet
- Now possible to trust other forests
- Still one domain per forest however
- Also still only suitable for fully-trusted domains
  - Not a security boundary

Replication diagnostics

- Visualised (4.8)
- Human realiable text
  - Inspired by CEPH
- JSON (4.10)

samba-tool improvements

- New ‘ou’ subcommand for Organisation Units
- New ‘computer’ subcommand for trust account management
- Improved ‘dns’ subcommand to be more friendly on failure
- New ‘group stats’ subcommand
  - Number of group memberships is an important scale factor
  - But most organisations only report number of users and groups

Customer request: 64-bit DB

- Concerned that the 4GB DB could be filled too quickly
  - Wanting to store > 100,000 users in a single domain!
- Main concern is the hard limit of TDB
- LMDB chosen as a modern key-value store
  - Used in OpenLDAP

LMDB

- LMDB pretty much did what it said on the tin
- Instead LMDB taught us about Samba and LDB
- Numerous locking issues found and fixed

A new approach: Key/Value layer

- Garming and I decided to add a key-value layer
  - Avoid code duplication
  - Allow more than just LMDB (perhaps LMDBx, LevelDB)
  - Share performance and correctness improvements with ldb_tdb
- And so, so many tests
  - Firmly locking down the semantics

First Hurdle: Locking

- Even the prototype found issues!
  - Demonstrated the lack of whole-DB locking
  - Fixed for Samba 4.7
- Probably behind many of our replication issues

Second Hurdle: More Locking!

- It just wouldn't pass make test!
  - More strange failures in replication
- Unlock ordering issues in replication
  - highestCommitedUSN visible before the data
  - Fixes proposed for backport to Samba 4.7 and 4.8
- Modification without locks (at startup) in Samba 4.8
  - DB-init time only, but not good
  - Added checks to key-value layer to prevent re-occurrence

Third hurdle: Maximum key size

- TDB has an unlimited key size
- LMDB is limited to 511 bytes
- LDB traditionally used the DN as the key
  - Addressed by the new GUID key system
  - Special handling needed for index keys (truncation)

And what about performance?

- Three performance tools measured so far:
  - Make perftest on our Hardware test server
    - Old AMD Athalon!
  - Traffic replay tool in the cloud
  - Adding users and users into groups of my workstation

Make perftest

- First performance numbers were, well, a disappointment
- 30% performance loss!
  - LMDB uses write(), and a read-only mmap()
  - socket_wrapper intercepts write()
- Workaround:
  - Use Linux userspace namespaces instead of socket_wrapper
  - Patches to upstream this still pending
- End result is no major change, perhaps 10% slower

Traffic replay

- This is a tool to replay an amplified anonymised traffic capture
- Similar numbers to TDB
- Need to re-try with a larger DB
  - We think LMDB will show most strength at large sizes

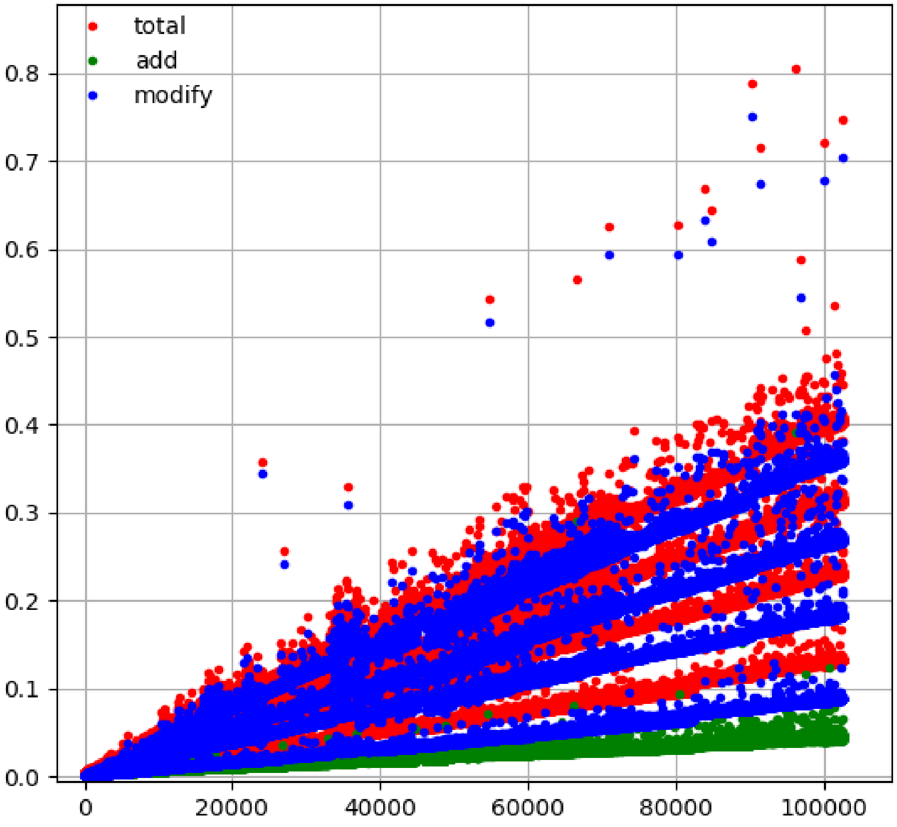
Adding users and users into groups on my workstation

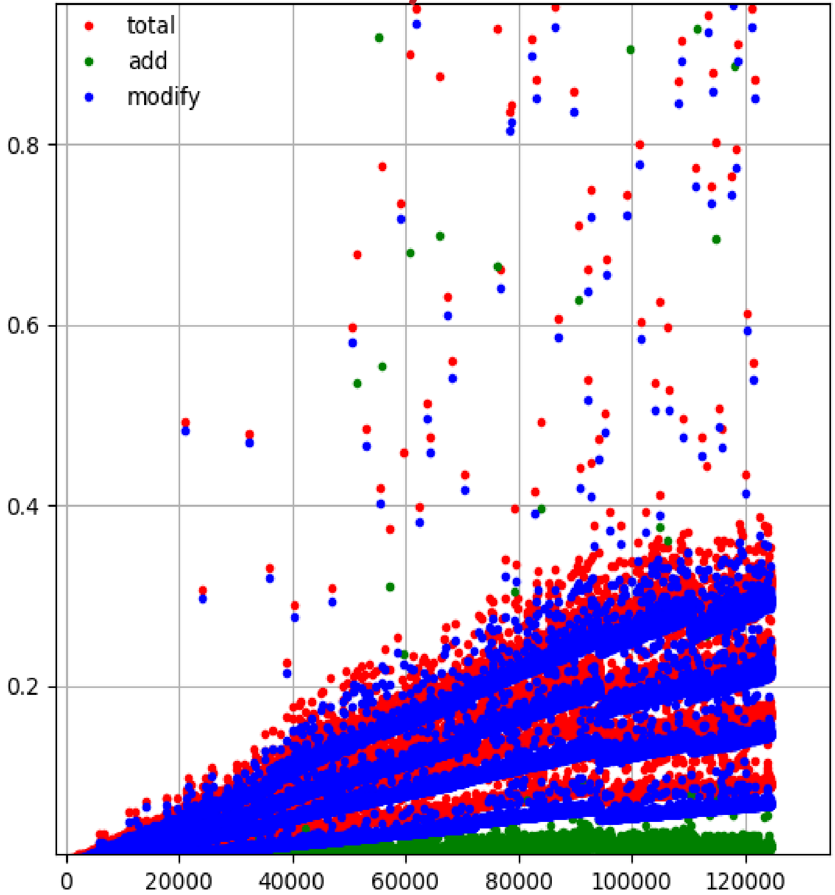
- In a four-hour benchmark adding users and adding into one to four groups (in rotation):
  - Samba 4.4: 26,000 users
  - Samba 4.5: 48,000 users
  - Samba 4.6: 55,000 users
  - Samba 4.7: 85,000 users
  - Samba 4.8: 100,000 users
  - Samba 4.9: 100,000 users (TDB)
  - Samba 4.9: 45,000 users (LMDB)

Ouch. What went wrong!

- fsync()/fdatasync()/msync() still called
- Patches quickly written
- New numbers:
  - Samba 4.9: 100,000 users (TDB)
  - Samba 4.9: 124,000 users (LMDB, no fsync())
- Lesson:
  - Samba’s module stack is still the slowest factor

TDB vs LMDB (latency vs number of users added)





OK, so not so bad

- We addressed the customer's desire for scale
  - Currently limited to 6GB but that is compile-time constant only
- Opens up new opportunities
  - Could use sub-databases instead of multiple files
  - Use ordered walk for indexed range searches?

LMDB: Sharp Edges

- Different locking behaviour (no exclusive access)
- Files are sparse by default
  - DB operations can fill the file and partition without going via a specific resize
- Files are not extended automatically
  - The inverse to the above, when a file is full unlike TDB there is no auto-resize
  - Requires that the admin or Samba know the size up-front
    - LDB / Samba has not required this kind of planning in the past
- Need real-world experience

Still TODO

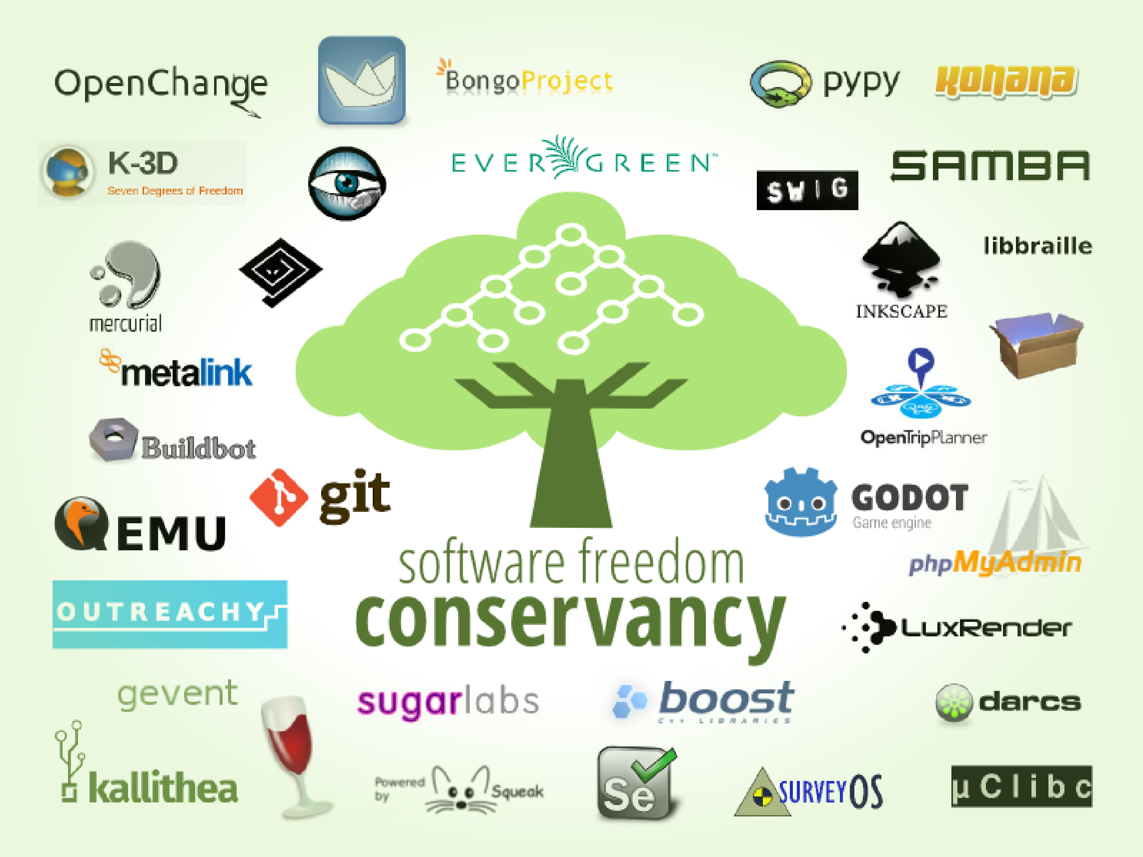
- Full support at 100,000 users is a task for Samba 4.11
  - Expected September 2019
- Improve subtree rename efficiently
  - Faster re-organisation of OUs
- New pack format
  - Avoid reading data that will not be returned
- Improved memory management
  - Avoid individual memory allocations when not required

So, are we there yet?

- Probably
- Look forward to Samba 4.10 and Samba 4.11
- Real world feedback really valuable
  - Let me know if you are using Samba AD at whatever scale

Become an OFFICIAL CONSERVANCY SUPPORTER!

GPL



Catalyst's Open Source Technologies – Questions?

Linux
debian
perl
php
Git
PostgreSQL
Lucene
Drupal
WordPress
moodle
totara
mahara
koha
kannel
openstack
CLOUD SOFTWARE
ceph
SAMBA

Want to work with my team at Catalyst to make your Samba scale? - talk to me in the hallway track!