So Samba 4.0 is out ... 
and what’s next?

sambaXP 2013

Michael Adam (obnoxsamba.org)
Stefan Metzmacher (metz@samba.org)

Samba Team / SerNet

2013-05-15
On December 11, 2012 ...
… around 6pm CET …
... something unexpected happened! ...
Samba 4.0.0 was released!
History (Part 1)

- 2003-06-07: Samba 3.0.0 beta1
- 2003-08-13: First public commit of Samba 4 code (Tridge)
  - 773 files changed, 352638 insertions(+)
  - focus: Full protocol testing for SMB (there is no documentation yet)
  - focus: NTVFS - rewrite of SMB server
  - focus: make SMB clusterable?
- 2003-08-16: Samba 3.0.0 rc1
- 2003-10-24: Samba 3.0.0 released
- ... code repositories diverge ...
- 2004-03-31: LDB is introduced into Samba4
- 2004-04-25: PIDL is introduced into Samba4
- 2006-01-31: Release of Samba4WINS based on Samba 4
2006: S4 focus changed: Implementation of AD/DC
   - The Samba Team worked out AD replication (without documentation)

... code repositories diverge ...

2007-12-19: Samba Team receives documentation from Microsoft

2007-2008: S3 is made cluster aware with CTDB

2008-05-08: Franky-idea is born

2008-07-01: Samba 3.2.0 released
   - GPLv3+
   - including PIDL from S4
   - experimental cluster support

2008-09-14: Merged branch for Samba3 and Samba4
   - v3-devel:source/ → master:source3/
   - v4-0-test:source/ → master:source4/
   - common/merged build
History (Part 3)

▶ ... reconcile commonly used components into top level ...
  ▶ talloc, tdb, tevent, ldb
  ▶ lib/util, libcli/smb, librpc/
▶ 2010-03-01: Samba 3.5.0 is released
  ▶ experimental support for SMB 2.0
▶ 2011-08-09: Samba 3.6.0 is released
  ▶ official support for SMB 2.0 (except for durable handles)
▶ ...

▶ sambaXP 2012: Samba 4 was not really covered! 😞
Chronicle since sambaXP 2012

- 2012-06-05: 4.0.0 beta1
- 2012-08-31: 4.0.0 beta8
- 2012-09-13: 4.0.0 rc1
- 2012-12-04: 4.0.0 rc6
- 2012-12-11: 4.0.0 (2865 commits since beta1)
- 2013-01-14: 4.0.1
- 2013-01-29: 4.0.2
- 2013-02-05: 4.0.3
- 2013-03-19: 4.0.4
- 2013-04-09: 4.0.5 (473 commits since 4.0.0)
What does Samba 4.0 look like?
What does Samba 4.0 look like?

- **NEW:**
  - Active Directory Compatible Server (AD/DC)
    - daemon "samba"
    - integrated LDAP server
    - integrated Kerberos server (heimdal)
    - integrated DNS server (or external bind)
    - SMB server: smbd (started automagically)
    - very simple to set up and run!

- **classical:**
  - Standalone and domain member file server as known from Samba 3
    - daemons smbd, nmbd, winbindd
    - improved file server
      - SMB 2.0 now complete with durable handles
      - partial SMB 2.1 support with Multi-Credit
      - basic SMB 3.0 support
How Can I Use It?

- Compile from sources
  - requires python (waf)
  - https://wiki.samba.org/index.php/BuildsystemUseAndWhy

- Packages from SerNet: (Commercial-Alert!)
  - available since 2013-05-14
  - http://www.enterprisesamba.com/samba/
  - for Debian, Ubuntu, RHEL, CentOS, SLES, openSUSE
"officially" supported in 4.0:
- forests: 1, domains: 1, domain controllers: 1*

trusts:
- Samba can be trusted
- Samba can not trust (yet)

replication:
- directory replication works
- sysvol replication not implemented yet
- *multiple Samba DCs possible (sysvol replicated externally)
And what’s next?
AD/DC TODOs

- subdomains
- trusts in general
- winbindd/idmap todos
- sysvol-replication (file system replication)
  - need async dcerpc server infrastructure
  - may require a better MS-FSA like abstraction in the file server backend
- ... ⇒ Matthieu Patou’s talk
SMB server TODOs

- SMB 2.1:
  - leases
- SMB 3.0:
  - directory leases
  - multi channel
  - RDMA
  - cluster concepts
    (scale-out/continuous availability)
  - persistent handles
  - witness

SMB: Leases and Directory Leases

- Leases ⇒ oplocks done right (content caching)
- Directory Leases ⇒ change notify done right (metadata caching)
- extend FSA oplocks to cope with SMB oplocks and leases
- remove the 1:1 relation between open and oplock (locking.tdb)
- add support for oplock keys (empty for SMB 1)
- cleanup / preparation work was already started by Volker
- lease keys and client guid need to be maintained at the SMB layer
SMB: Multi Channel

- bind multiple transport connections to one SMB session
- interface discovery:
  - new fsctl (FSCTL_QUERY_NETWORK_INTERFACE_INFO)
  - client just connect to one cluster node

- extend current 1:1 relation smbd ↔ TCP connection
- transfer TCP-socket to smbd serving connections with the same ClientGUID in negprot (fd-passing)
- ⇒ session bind automatically on correct smbd
- ⇒ only one process has the file open for multi-channel sessions
- ⇒ we only need to do book-keeping on the SMB level (replay/retry counters, channel sequence numbers, ....)
- ⇒ the posix/file system level won’t notice multi channel
SMB: RDMA

- RDMA uses (infiniband, iWarp or RoCE hardware, or sofware emulation)
- transport abstraction needed (TCP/NBT vs. RDMA)
- buffer abstraction needed in order to do zero-copy transfers
  - SMB_VFS_READ_BUFFER_SEND/RECV
  - SMB_VFS_WRITE_BUFFER_SEND/RECV
- there’s already a hacked client implementation

- Problems with the current libibverbs/librdmacm libraries
  - it’s not fork() safe, which is currently required by smbd
  - ”FD-passing” is not supported, would be needed for the current planned multi channel design
persistent handles: durable handles with strong guarantees

server application workload

need to make some DBs persistent (or by record) (⇒ changes to tdb / ctdb ...)
  smbXsrv_open_global
  locking
  brlock

need new index databases for smbXsrv_open
The witness protocol is implemented as DCERPC service.
- Using `ncacn_ip_tcp` as transport
- Heartbeat link between a SMB 3.0 client and server cluster.
- It provides faster planned or unplanned failover

Needs async DCERPC server infrastructure
- Which can be used independently from `smbd/samba`
- It needs to support a single process model
Questions?