

Scalable NAS Cluster With Samba And CTDB

Source Talk Tage 2010

Michael Adam

`obnox@samba.org`

SerNet / Samba Team

2010-08-31

Outline

- 1 Introduction
- 2 Cluster Challenges
 - Introduction
 - Challenges For Samba
- 3 CTDB
 - The CTDB Project
 - CTDB Design
 - Setting Up CTDB
- 4 Clustered Samba
 - Getting Sources and Binaries
 - Clustered File Systems
 - Samba Configuration
 - CTDB manages...
 - Registry Configuration

Outline

- 1 Introduction
- 2 Cluster Challenges
 - Introduction
 - Challenges For Samba
- 3 CTDB
 - The CTDB Project
 - CTDB Design
 - Setting Up CTDB
- 4 Clustered Samba
 - Getting Sources and Binaries
 - Clustered File Systems
 - Samba Configuration
 - CTDB manages...
 - Registry Configuration

about me

- Mathematics (Göttingen, Bonn, ...)
- working for SerNet since 2002
- Linux / Open Source since early 90ies
- development and consulting of Samba and CTDB
- (clustering, registry, ID-Mapping, ACLs, ...)
- Co-author of german Samba book

Samba

- **Windows/Unix interoperability Software for Unix/Linux**
- SMB/CIFS file and print server
- domain controller / logon server (old Windows NT style)
- started 1992 by Andrew Tridgell
- international development team
(<http://www.samba.org/samba/team/>)
- used in production in small – big business
- small NAS boxes – some big NAS appliances
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- sambaXP: annual spring conference in Göttingen
- <http://www.samba.org/>

Samba

- Windows/Unix interoperability Software for Unix/Linux
- **SMB/CIFS file and print server**
- domain controller / logon server (old Windows NT style)
- started 1992 by Andrew Tridgell
- international development team
(<http://www.samba.org/samba/team/>)
- used in production in small – big business
- small NAS boxes – some big NAS appliances
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- sambaXP: annual spring conference in Göttingen
- <http://www.samba.org/>

Samba

- Windows/Unix interoperability Software for Unix/Linux
- SMB/CIFS file and print server
- domain controller / logon server (old Windows NT style)
- started 1992 by Andrew Tridgell
- international development team
(<http://www.samba.org/samba/team/>)
- used in production in small – big business
- small NAS boxes – some big NAS appliances
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- sambaXP: annual spring conference in Göttingen
- <http://www.samba.org/>

Samba

- Windows/Unix interoperability Software for Unix/Linux
- SMB/CIFS file and print server
- domain controller / logon server (old Windows NT style)
- **started 1992 by Andrew Tridgell**
- international development team
(<http://www.samba.org/samba/team/>)
- used in production in small – big business
- small NAS boxes – some big NAS appliances
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- sambaXP: annual spring conference in Göttingen
- <http://www.samba.org/>

Samba

- Windows/Unix interoperability Software for Unix/Linux
- SMB/CIFS file and print server
- domain controller / logon server (old Windows NT style)
- started 1992 by Andrew Tridgell
- international development team
(<http://www.samba.org/samba/team/>)
- used in production in small – big business
- small NAS boxes – some big NAS appliances
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- sambaXP: annual spring conference in Göttingen
- <http://www.samba.org/>

Samba

- Windows/Unix interoperability Software for Unix/Linux
- SMB/CIFS file and print server
- domain controller / logon server (old Windows NT style)
- started 1992 by Andrew Tridgell
- international development team
(<http://www.samba.org/samba/team/>)
- **used in production in small – big business**
- small NAS boxes – some big NAS appliances
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- sambaXP: annual spring conference in Göttingen
- <http://www.samba.org/>

Samba

- Windows/Unix interoperability Software for Unix/Linux
- SMB/CIFS file and print server
- domain controller / logon server (old Windows NT style)
- started 1992 by Andrew Tridgell
- international development team
(<http://www.samba.org/samba/team/>)
- used in production in small – big business
- **small NAS boxes – some big NAS appliances**
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- sambaXP: annual spring conference in Göttingen
- <http://www.samba.org/>

Samba

- Windows/Unix interoperability Software for Unix/Linux
- SMB/CIFS file and print server
- domain controller / logon server (old Windows NT style)
- started 1992 by Andrew Tridgell
- international development team
(<http://www.samba.org/samba/team/>)
- used in production in small – big business
- small NAS boxes – some big NAS appliances
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- sambaXP: annual spring conference in Göttingen
- <http://www.samba.org/>

Samba

- Windows/Unix interoperability Software for Unix/Linux
- SMB/CIFS file and print server
- domain controller / logon server (old Windows NT style)
- started 1992 by Andrew Tridgell
- international development team
(<http://www.samba.org/samba/team/>)
- used in production in small – big business
- small NAS boxes – some big NAS appliances
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- **sambaXP: annual spring conference in Göttingen**
- <http://www.samba.org/>

Samba

- Windows/Unix interoperability Software for Unix/Linux
- SMB/CIFS file and print server
- domain controller / logon server (old Windows NT style)
- started 1992 by Andrew Tridgell
- international development team
(<http://www.samba.org/samba/team/>)
- used in production in small – big business
- small NAS boxes – some big NAS appliances
- “Samba4” - Active Directory server (alpha!, release Spring 2011?)
- sambaXP: annual spring conference in Göttingen
- <http://www.samba.org/>

SerNet GmbH

- Linux / Open Source support company, founded 1996
- Göttingen, Berlin, Nürnberg
- 5 samba core team members
- including SerNet co-founder Volker Lendecke and Samba release manager Karolin Seeger
- Samba: support, consulting, development
- apart from that: firewalls, mail, proxy, webserver, vpn, . . . , certs-and-audits
- <http://www.sernet.de/>

SerNet GmbH

- Linux / Open Source support company, founded 1996
- Göttingen, Berlin, Nürnberg
- 5 samba core team members
- including SerNet co-founder Volker Lendecke and Samba release manager Karolin Seeger
- Samba: support, consulting, development
- apart from that: firewalls, mail, proxy, webserver, vpn, . . . , certs-and-audits
- <http://www.sernet.de/>

SerNet GmbH

- Linux / Open Source support company, founded 1996
- Göttingen, Berlin, Nürnberg
- 5 samba core team members
- including SerNet co-founder Volker Lendecke and Samba release manager Karolin Seeger
- Samba: support, consulting, development
- apart from that: firewalls, mail, proxy, webserver, vpn, . . . , certs-and-audits
- <http://www.sernet.de/>

SerNet GmbH

- Linux / Open Source support company, founded 1996
- Göttingen, Berlin, Nürnberg
- 5 samba core team members
- including SerNet co-founder Volker Lendecke and Samba release manager Karolin Seeger
- Samba: support, consulting, development
- apart from that: firewalls, mail, proxy, webserver, vpn, . . . , certs-and-audits
- <http://www.sernet.de/>

SerNet GmbH

- Linux / Open Source support company, founded 1996
- Göttingen, Berlin, Nürnberg
- 5 samba core team members
- including SerNet co-founder Volker Lendecke and Samba release manager Karolin Seeger
- **Samba: support, consulting, development**
- apart from that: firewalls, mail, proxy, webserver, vpn, . . . , certs-and-audits
- `http://www.sernet.de/`

SerNet GmbH

- Linux / Open Source support company, founded 1996
- Göttingen, Berlin, Nürnberg
- 5 samba core team members
- including SerNet co-founder Volker Lendecke and Samba release manager Karolin Seeger
- Samba: support, consulting, development
- apart from that: firewalls, mail, proxy, webserver, vpn, . . . , certs-and-audits
- <http://www.sernet.de/>

SerNet GmbH

- Linux / Open Source support company, founded 1996
- Göttingen, Berlin, Nürnberg
- 5 samba core team members
- including SerNet co-founder Volker Lendecke and Samba release manager Karolin Seeger
- Samba: support, consulting, development
- apart from that: firewalls, mail, proxy, webserver, vpn, . . . , certs-and-audits
- <http://www.sernet.de/>

Outline

- 1 Introduction
- 2 Cluster Challenges
 - Introduction
 - Challenges For Samba
- 3 CTDB
 - The CTDB Project
 - CTDB Design
 - Setting Up CTDB
- 4 Clustered Samba
 - Getting Sources and Binaries
 - Clustered File Systems
 - Samba Configuration
 - CTDB manages...
 - Registry Configuration

Ideas

- idea: share a cluster file system as a network service (NFS/CIFS)
- i.e. turn your SAN into a *clustered* NAS
- \Rightarrow we want to cluster Samba in an *all-active* fashion (load balancing cluster, not ha-cluster)
- with CTDB, we *can* cluster Samba (and nfs, and ...)
- note 1: active-passive clustering is available since long
- note 2: Microsoft currently does not active-active clustering of CIFS

Ideas

- idea: share a cluster file system as a network service (NFS/CIFS)
- i.e. turn your SAN into a *clustered* NAS
- ⇒ we want to cluster Samba in an *all-active* fashion (load balancing cluster, not ha-cluster)
- with CTDB, we *can* cluster Samba (and nfs, and ...)
- note 1: active-passive clustering is available since long
- note 2: Microsoft currently does not active-active clustering of CIFS

Ideas

- idea: share a cluster file system as a network service (NFS/CIFS)
- i.e. turn your SAN into a *clustered* NAS
- ⇒ we want to cluster Samba in an *all-active* fashion (load balancing cluster, not ha-cluster)
- with CTDB, we *can* cluster Samba (and nfs, and ...)
- note 1: active-passive clustering is available since long
- note 2: Microsoft currently does not active-active clustering of CIFS

Ideas

- idea: share a cluster file system as a network service (NFS/CIFS)
- i.e. turn your SAN into a *clustered* NAS
- \Rightarrow we want to cluster Samba in an *all-active* fashion (load balancing cluster, not ha-cluster)
- with CTDB, we *can* cluster Samba (and nfs, and ...)
- note 1: active-passive clustering is available since long
- note 2: Microsoft currently does not active-active clustering of CIFS

Ideas

- idea: share a cluster file system as a network service (NFS/CIFS)
- i.e. turn your SAN into a *clustered* NAS
- \Rightarrow we want to cluster Samba in an *all-active* fashion (load balancing cluster, not ha-cluster)
- with CTDB, we *can* cluster Samba (and nfs, and ...)
- **note 1: active-passive clustering is available since long**
- note 2: Microsoft currently does not active-active clustering of CIFS

Ideas

- idea: share a cluster file system as a network service (NFS/CIFS)
- i.e. turn your SAN into a *clustered* NAS
- \Rightarrow we want to cluster Samba in an *all-active* fashion (load balancing cluster, not ha-cluster)
- with CTDB, we *can* cluster Samba (and nfs, and ...)
- note 1: active-passive clustering is available since long
- note 2: Microsoft currently does not active-active clustering of CIFS

Starting Points

- Samba daemons on cluster nodes need to act as *one* CIFS server
- hence we need IPC of Samba daemons between nodes
- furthermode share some persistent data

Starting Points

- Samba daemons on cluster nodes need to act as *one* CIFS server
- hence we need IPC of Samba daemons between nodes
- furthermore share some persistent data

Starting Points

- Samba daemons on cluster nodes need to act as *one* CIFS server
- hence we need IPC of Samba daemons between nodes
- furthermode share some persistent data

Challenges For Samba

- Samba uses internal tdb data bases:
 - IPC: messaging (`messages.tdb` and `signals`)
 - IPC: share volatile session data:
 - `session.tdb` (per session)
 - `connections.tdb` (per connection)
 - `locks.tdb` (locking)
 - `locks.tdb` (per file)
 - share certain persistent data:
 - `passwd.tdb` (passwords)
 - `admins.tdb` (admins)
 - `groups.tdb` (groups)
 - `users.tdb` (users)

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - locks (`locks.tdb`)
 - file share locks (`filelocks.tdb`)
- share certain persistent data:
 - users database (`passwd.tdb`)
 - domain information (`domain.tdb`)
 - local SAM database (`local_sam.tdb`)
 - local security database (`local_security.tdb`)

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- share certain persistent data:
 - user database (`passwd.tdb`)
 - domain controller information (`dcinfo.tdb`)
 - local user information (`local_auth.tdb`)
 - local group information (`local_group.tdb`)
 - local share information (`local_shares.tdb`)

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- share certain persistent data:

• `sharemodes.tdb` (`messages.tdb`)

• `sharemodes.tdb` (`connections.tdb`)

• `sharemodes.tdb` (`locking.tdb`)

• `sharemodes.tdb` (`brlock.tdb`)

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - **share connections (`connections.tdb`)**
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- share certain persistent data:

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- share certain persistent data:

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - **byte range locks (`brlock.tdb`)**
- share certain persistent data:
 - user database (`passwd.tdb`)

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- share certain persistent data:
 - user database (`passdb.tdb`)
 - domain join information (`secrets.tdb`)
 - id mapping tables (`winbindd_idmap.tdb`)
 - registry (`registry.tdb`)

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- share certain persistent data:
 - **user database (`passdb.tdb`)**
 - domain join information (`secrets.tdb`)
 - id mapping tables (`winbindd_idmap.tdb`)
 - registry (`registry.tdb`)

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- share certain persistent data:
 - user database (`passdb.tdb`)
 - domain join information (`secrets.tdb`)
 - id mapping tables (`winbindd_idmap.tdb`)
 - registry (`registry.tdb`)

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- share certain persistent data:
 - user database (`passdb.tdb`)
 - domain join information (`secrets.tdb`)
 - id mapping tables (`winbindd_idmap.tdb`)
 - registry (`registry.tdb`)

Challenges For Samba

- Samba uses internal tdb data bases:
- IPC: messaging (`messages.tdb` and `signals`)
- IPC: share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- share certain persistent data:
 - user database (`passdb.tdb`)
 - domain join information (`secrets.tdb`)
 - id mapping tables (`winbindd_idmap.tdb`)
 - registry (`registry.tdb`)

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - must all reside on separate servers (no shared storage) at each node
 - TDBs like `locks.tdb`
 - I/O performance critical for users! However performance is especially important for the Windows locks
- persistent TDBs:
 - read frequently
 - written infrequently
 - data consistency very important

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - must be available on all nodes (for consistency, not for performance)
 - must be able to be replicated
 - especially important for the Windows locks
- persistent TDBs:

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- **volatile (“normal”) TDBs:**
 - read and written very frequently
 - not all data must be known to every node (or `smbd` process) at each point in time
 - R/W performance critical for overall fileserver performance
 - especially important for the Windows locks
- persistent TDBs:

• `lock:lock` (very important)

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - not all data must be known to every node (or smbd process) at each point in time
 - R/W performance critical for overall fileserver performance
 - especially important for the Windows locks
- persistent TDBs:

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - not all data must be known to every node (or smbd process) at each point in time
 - R/W performance critical for overall fileserver performance
 - especially important for the Windows locks
- persistent TDBs:

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - not all data must be known to every node (or `smbd` process) at each point in time
 - **R/W performance critical for overall fileserver performance**
 - especially important for the Windows locks
- persistent TDBs:

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - not all data must be known to every node (or `smbd` process) at each point in time
 - R/W performance critical for overall fileserver performance
 - **especially important for the Windows locks**
- persistent TDBs:
 - read frequently

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - not all data must be known to every node (or `smbd` process) at each point in time
 - R/W performance critical for overall fileserver performance
 - especially important for the Windows locks
- persistent TDBs:
 - read frequently
 - written rather rarely
 - data consistency very important

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - not all data must be known to every node (or `smbd` process) at each point in time
 - R/W performance critical for overall fileserver performance
 - especially important for the Windows locks
- persistent TDBs:
 - read frequently
 - written rather rarely
 - data consistency very important

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - not all data must be known to every node (or `smbd` process) at each point in time
 - R/W performance critical for overall fileserver performance
 - especially important for the Windows locks
- persistent TDBs:
 - read frequently
 - **written rather rarely**
 - data consistency very important

TDBs

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- volatile (“normal”) TDBs:
 - read and written very frequently
 - not all data must be known to every node (or `smbd` process) at each point in time
 - R/W performance critical for overall fileserver performance
 - especially important for the Windows locks
- persistent TDBs:
 - read frequently
 - written rather rarely
 - data consistency very important

TDBs And Clustering

- TDB R/W performance critical for Samba performance
- TDB R/W operations: excessive use of POSIX `fcntl` byte range locks
- `fcntl` locks are usually slow on cluster file systems
- the more nodes, the slower...
- ⇒ naive approach of putting TDBs on cluster storage works in principle but scales *very badly*
- Usual clustered data bases are also too slow.
- A more specialized approach is needed.

TDBs And Clustering

- TDB R/W performance critical for Samba performance
- TDB R/W operations: excessive use of POSIX `fcntl` byte range locks
- `fcntl` locks are usually slow on cluster file systems
- the more nodes, the slower...
- ⇒ naive approach of putting TDBs on cluster storage works in principle but scales *very badly*
- Usual clustered data bases are also too slow.
- A more specialized approach is needed.

TDBs And Clustering

- TDB R/W performance critical for Samba performance
- TDB R/W operations: excessive use of POSIX `fcntl` byte range locks
- `fcntl` locks are usually slow on cluster file systems
- the more nodes, the slower...
- ⇒ naive approach of putting TDBs on cluster storage works in principle but scales *very badly*
- Usual clustered data bases are also too slow.
- A more specialized approach is needed.

TDBs And Clustering

- TDB R/W performance critical for Samba performance
- TDB R/W operations: excessive use of POSIX `fcntl` byte range locks
- `fcntl` locks are usually slow on cluster file systems
- **the more nodes, the slower...**
- ⇒ naive approach of putting TDBs on cluster storage works in principle but scales *very badly*
- Usual clustered data bases are also too slow.
- A more specialized approach is needed.

TDBs And Clustering

- TDB R/W performance critical for Samba performance
- TDB R/W operations: excessive use of POSIX `fcntl` byte range locks
- `fcntl` locks are usually slow on cluster file systems
- the more nodes, the slower...
- \Rightarrow naive approach of putting TDBs on cluster storage works in principle but scales *very badly*
- Usual clustered data bases are also too slow.
- A more specialized approach is needed.

TDBs And Clustering

- TDB R/W performance critical for Samba performance
- TDB R/W operations: excessive use of POSIX `fcntl` byte range locks
- `fcntl` locks are usually slow on cluster file systems
- the more nodes, the slower...
- \Rightarrow naive approach of putting TDBs on cluster storage works in principle but scales *very badly*
- Usual clustered data bases are also too slow.
- A more specialized approach is needed.

TDBs And Clustering

- TDB R/W performance critical for Samba performance
- TDB R/W operations: excessive use of POSIX `fcntl` byte range locks
- `fcntl` locks are usually slow on cluster file systems
- the more nodes, the slower...
- \Rightarrow naive approach of putting TDBs on cluster storage works in principle but scales *very badly*
- Usual clustered data bases are also too slow.
- **A more specialized approach is needed.**

Goals

- Cluster Samba So That:
 - One node is not slower than an unclustered Samba server.
 - $n + 1$ nodes should be faster than n nodes.
- This in requires a clustered TDB implementation ...
- ... and messaging solution.
- \Rightarrow CTDB

Goals

- Cluster Samba So That:
 - One node is not slower than an unclustered Samba server.
 - $n + 1$ nodes should be faster than n nodes.
- This in requires a clustered TDB implementation ...
- ... and messaging solution.
- \Rightarrow CTDB

Goals

- Cluster Samba So That:
 - One node is not slower than an unclustered Samba server.
 - $n + 1$ nodes should be faster than n nodes.
- This in requires a clustered TDB implementation ...
- ... and messaging solution.
- \Rightarrow CTDB

Goals

- Cluster Samba So That:
 - One node is not slower than an unclustered Samba server.
 - $n + 1$ nodes should be faster than n nodes.
- This in requires a clustered TDB implementation ...
 - ... and messaging solution.
 - ⇒ CTDB

Goals

- Cluster Samba So That:
 - One node is not slower than an unclustered Samba server.
 - $n + 1$ nodes should be faster than n nodes.
- This in requires a clustered TDB implementation ...
- ... and messaging solution.
- ⇒ CTDB

Goals

- Cluster Samba So That:
 - One node is not slower than an unclustered Samba server.
 - $n + 1$ nodes should be faster than n nodes.
- This in requires a clustered TDB implementation ...
- ... and messaging solution.
- ⇒ CTDB

Outline

- 1 Introduction
- 2 Cluster Challenges
 - Introduction
 - Challenges For Samba
- 3 CTDB
 - The CTDB Project
 - CTDB Design
 - Setting Up CTDB
- 4 Clustered Samba
 - Getting Sources and Binaries
 - Clustered File Systems
 - Samba Configuration
 - CTDB manages...
 - Registry Configuration

The CTDB Project

- started in 2006
- first prototype in v1-messaging SVN branch
- Volker Lendecke, Andrew Tridgell, ...
- first usable version of CTDB: April 2007
- meanwhile: Ronnie Sahlberg project maintainer
- `git://git.samba.org/sahlberg/ctdb.git`
- `http://ctdb.samba.org/packages/` (RPMs, Sources)

The CTDB Project

- started in 2006
- first prototype in v1-messaging SVN branch
- Volker Lendecke, Andrew Tridgell, ...
- first usable version of CTDB: April 2007
- meanwhile: Ronnie Sahlberg project maintainer
- [git://git.samba.org/sahlberg/ctdb.git](https://git.samba.org/sahlberg/ctdb.git)
- <http://ctdb.samba.org/packages/> (RPMs, Sources)

The CTDB Project

- started in 2006
- first prototype in v1-messaging SVN branch
- Volker Lendecke, Andrew Tridgell, ...
- first usable version of CTDB: April 2007
- meanwhile: Ronnie Sahlberg project maintainer
- `git://git.samba.org/sahlberg/ctdb.git`
- `http://ctdb.samba.org/packages/` (RPMs, Sources)

The CTDB Project

- started in 2006
- first prototype in v1-messaging SVN branch
- Volker Lendecke, Andrew Tridgell, ...
- **first usable version of CTDB: April 2007**
- meanwhile: Ronnie Sahlberg project maintainer
- `git://git.samba.org/sahlberg/ctdb.git`
- `http://ctdb.samba.org/packages/` (RPMs, Sources)

The CTDB Project

- started in 2006
- first prototype in v1-messaging SVN branch
- Volker Lendecke, Andrew Tridgell, ...
- first usable version of CTDB: April 2007
- **meanwhile: Ronnie Sahlberg project maintainer**
- `git://git.samba.org/sahlberg/ctdb.git`
- `http://ctdb.samba.org/packages/` (RPMs, Sources)

The CTDB Project

- started in 2006
- first prototype in v1-messaging SVN branch
- Volker Lendecke, Andrew Tridgell, ...
- first usable version of CTDB: April 2007
- meanwhile: Ronnie Sahlberg project maintainer
- [git://git.samba.org/sahlberg/ctdb.git](https://git.samba.org/sahlberg/ctdb.git)
- <http://ctdb.samba.org/packages/> (RPMs, Sources)

The CTDB Project

- started in 2006
- first prototype in v1-messaging SVN branch
- Volker Lendecke, Andrew Tridgell, ...
- first usable version of CTDB: April 2007
- meanwhile: Ronnie Sahlberg project maintainer
- `git://git.samba.org/sahlberg/ctdb.git`
- <http://ctdb.samba.org/packages/> (RPMs, Sources)

The CTDB Project - Community

- #ctdb channel on freenode
- samba-technical mailing list
- feedback and contributions by packagers
- increasing development activity, number of developers

The CTDB Project - Community

- #ctdb channel on freenode
- **samba-technical mailing list**
- feedback and contributions by packagers
- increasing development activity, number of developers

The CTDB Project - Community

- #ctdb channel on freenode
- samba-technical mailing list
- feedback and contributions by packagers
- increasing development activity, number of developers

The CTDB Project - Community

- #ctdb channel on freenode
- samba-technical mailing list
- feedback and contributions by packagers
- increasing development activity, number of developers

CTDB Design - Warning

A Word Of Warning

- Client connections are *not* spread over multiple cluster nodes.
- I.e., each single client connection (CIFS, nfs, ...) is served by one node just as a non-clustered file server would server the connection.
- Hence a single connection is not faster than on a non-clustered file server, but the sum should (possibly) be faster.
- In case of failover, connections are not migrated: clients need to reconnect.
- The CIFS protocol just can't do that.

CTDB Design - Warning

A Word Of Warning

- Client connections are *not* spread over multiple cluster nodes.
- I.e., each single client connection (CIFS, nfs, ...) is served by one node just as a non-clustered file server would server the connection.
- Hence a single connection is not faster than on a non-clustered file server, but the sum should (possibly) be faster.
- In case of failover, connections are not migrated: clients need to reconnect.
- The CIFS protocol just can't do that.

CTDB Design - Warning

A Word Of Warning

- Client connections are *not* spread over multiple cluster nodes.
- I.e., each single client connection (CIFS, nfs, ...) is served by one node just as a non-clustered file server would server the connection.
- Hence a single connection is not faster than on a non-clustered file server, but the sum should (possibly) be faster.
- In case of failover, connections are not migrated: clients need to reconnect.
- The CIFS protocol just can't do that.

CTDB Design - Warning

A Word Of Warning

- Client connections are *not* spread over multiple cluster nodes.
- I.e., each single client connection (CIFS, nfs, ...) is served by one node just as a non-clustered file server would server the connection.
- Hence a single connection is not faster than on a non-clustered file server, but the sum should (possibly) be faster.
- In case of failover, connections are not migrated: clients need to reconnect.
- The CIFS protocol just can't do that.

CTDB Design - Warning

A Word Of Warning

- Client connections are *not* spread over multiple cluster nodes.
- I.e., each single client connection (CIFS, nfs, ...) is served by one node just as a non-clustered file server would server the connection.
- Hence a single connection is not faster than on a non-clustered file server, but the sum should (possibly) be faster.
- In case of failover, connections are not migrated: clients need to reconnect.
- The CIFS protocol just can't do that.

CTDB Design – General

- one daemon `ctdbd` on each node (and temporary forks)
- `smbd` talks to local `ctdbd` for messaging and TDB access
- `ctdbd` handles metadata of TDBs via the network
- `ctdbd` keeps local TDB copy (LTDB) for fast data reads/writes
- the actual record read and write ops are directly to the LTDB
- normal and persistent TDBs are handled differently
- HA and cluster management features: monitor and fail over/back IP addresses and Samba, NFS and other services

CTDB Design – General

- one daemon `ctdbd` on each node (and temporary forks)
- `smbd` talks to local `ctdbd` for messaging and TDB access
- `ctdbd` handles metadata of TDBs via the network
- `ctdbd` keeps local TDB copy (LTDB) for fast data reads/writes
- the actual record read and write ops are directly to the LTDB
- normal and persistent TDBs are handled differently
- HA and cluster management features: monitor and fail over/back IP addresses and Samba, NFS and other services

CTDB Design – General

- one daemon `ctdbd` on each node (and temporary forks)
- `smbd` talks to local `ctdbd` for messaging and TDB access
- `ctdbd` handles metadata of TDBs via the network
- `ctdbd` keeps local TDB copy (LTDB) for fast data reads/writes
- the actual record read and write ops are directly to the LTDB
- normal and persistent TDBs are handled differently
- HA and cluster management features: monitor and fail over/back IP addresses and Samba, NFS and other services

CTDB Design – General

- one daemon `ctdbd` on each node (and temporary forks)
- `smbd` talks to local `ctdbd` for messaging and TDB access
- `ctdbd` handles metadata of TDBs via the network
- `ctdbd` keeps local TDB copy (LTDB) for fast data reads/writes
- the actual record read and write ops are directly to the LTDB
- normal and persistent TDBs are handled differently
- HA and cluster management features: monitor and fail over/back IP addresses and Samba, NFS and other services

CTDB Design – General

- one daemon `ctdbd` on each node (and temporary forks)
- `smbd` talks to local `ctdbd` for messaging and TDB access
- `ctdbd` handles metadata of TDBs via the network
- `ctdbd` keeps local TDB copy (LTDB) for fast data reads/writes
- the actual record read and write ops are directly to the LTDB
- normal and persistent TDBs are handled differently
- HA and cluster management features: monitor and fail over/back IP addresses and Samba, NFS and other services

CTDB Design – General

- one daemon `ctdbd` on each node (and temporary forks)
- `smbd` talks to local `ctdbd` for messaging and TDB access
- `ctdbd` handles metadata of TDBs via the network
- `ctdbd` keeps local TDB copy (LTDB) for fast data reads/writes
- the actual record read and write ops are directly to the LTDB
- normal and persistent TDBs are handled differently
- HA and cluster management features: monitor and fail over/back IP addresses and Samba, NFS and other services

CTDB Design – General

- one daemon `ctdbd` on each node (and temporary forks)
- `smbd` talks to local `ctdbd` for messaging and TDB access
- `ctdbd` handles metadata of TDBs via the network
- `ctdbd` keeps local TDB copy (LTDB) for fast data reads/writes
- the actual record read and write ops are directly to the LTDB
- normal and persistent TDBs are handled differently
- HA and cluster management features: monitor and fail over/back IP addresses and Samba, NFS and other services

CTDB Design – normal TDBs

- one node does not need to know all records all the time:
- the records related to connections to a node are node specific
- when a node goes down:
- ⇒ we may, even *should* lose records specific to that node
- a node only has those records in its LTDB that it has already accessed

CTDB Design – normal TDBs

- one node does not need to know all records all the time:
- the records related to connections to a node are node specific
- when a node goes down:
- ⇒ we may, even *should* lose records specific to that node
- a node only has those records in its LTDB that it has already accessed

CTDB Design – normal TDBs

- one node does not need to know all records all the time:
- the records related to connections to a node are node specific
- when a node goes down:
 - ⇒ we may, even *should* lose records specific to that node
 - a node only has those records in its LTDB that it has already accessed

CTDB Design – normal TDBs

- one node does not need to know all records all the time:
- the records related to connections to a node are node specific
- when a node goes down:
 - ⇒ we may, even *should* lose records specific to that node
 - a node only has those records in its LTDB that it has already accessed

CTDB Design – normal TDBs

- one node does not need to know all records all the time:
- the records related to connections to a node are node specific
- when a node goes down:
- \Rightarrow we may, even *should* lose records specific to that node
- a node only has those records in its LTDB that it has already accessed

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - updates are written to the record by DMASTER
- LMASTER (location master):
 - knows the location of a record's DMASTER
 - can find (calculated by record hash)
 - LMASTER roles distributed across active nodes
- R/W operation to a record:
 - check if we are DMASTER
 - if not, request DMASTER role and current copy of record from nearest LMASTER
 - R/W locally

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- **DMASTER (data master):**
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):

• always the location of a record in CTDB

• can be calculated by record hash

• LMASTER role distributed across active nodes

- R/W operation to a record:

• always done by DMASTER

• if not DMASTER, node will connect to DMASTER and get a copy of record and update

• then return to local

• then return to local

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):

- R/W operation to a record:

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):

- knows the location of a record's DMASTER

- will find (and update) the DMASTER

- will find (and update) the DMASTER if the DMASTER is unreachable

- R/W operation to a record:

- always goes to DMASTER

- if DMASTER is unreachable, will find DMASTER and read the current copy of record

- will find (and update) the DMASTER

- will find (and update) the DMASTER

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):
 - knows the location of a record's DMASTER
 - is fixed (calculated by record hash)
 - LMASTER roles distributed across active nodes
- R/W operation to a record:



CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):
 - **knows the location of a record's DMASTER**
 - is fixed (calculated by record hash)
 - LMASTER roles distributed across active nodes
- R/W operation to a record:

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):
 - knows the location of a record's DMASTER
 - is fixed (calculated by record hash)
 - LMASTER roles distributed across active nodes
- R/W operation to a record:

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):
 - knows the location of a record's DMASTER
 - is fixed (calculated by record hash)
 - **LMASTER roles distributed across active nodes**
- R/W operation to a record:
 - check if we are DMASTER

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):
 - knows the location of a record's DMASTER
 - is fixed (calculated by record hash)
 - LMASTER roles distributed across active nodes
- R/W operation to a record:
 - check if we are DMASTER
 - if not, request DMASTER role and current copy of record over network (via LMASTER)
 - read/write locally

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):
 - knows the location of a record's DMASTER
 - is fixed (calculated by record hash)
 - LMASTER roles distributed across active nodes
- R/W operation to a record:
 - **check if we are DMASTER**
 - if not, request DMASTER role and current copy of record over network (via LMASTER)
 - read/write locally

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):
 - knows the location of a record's DMASTER
 - is fixed (calculated by record hash)
 - LMASTER roles distributed across active nodes
- R/W operation to a record:
 - check if we are DMASTER
 - if not, request DMASTER role and current copy of record over network (via LMASTER)
 - read/write locally

CTDB Design - Record Roles

- nodes can carry certain roles with respect to a record:
- DMASTER (data master):
 - has the current, authoritative copy of a record
 - moves around as nodes write to the record
- LMASTER (location master):
 - knows the location of a record's DMASTER
 - is fixed (calculated by record hash)
 - LMASTER roles distributed across active nodes
- R/W operation to a record:
 - check if we are DMASTER
 - if not, request DMASTER role and current copy of record over network (via LMASTER)
 - **read/write locally**

Recovery

- what happens if a node goes down?
- data master for some records will be lost
- one node – the *recovery master* – performs *recovery*
- recovery master collects most recent copy of all records from all nodes
- additional TDB header *record sequence number* determines recentness
- at the end, the recovery master is data master for all records

Recovery

- what happens if a node goes down?
- data master for some records will be lost
- one node – the *recovery master* – performs *recovery*
- recovery master collects most recent copy of all records from all nodes
- additional TDB header *record sequence number* determines recentness
- at the end, the recovery master is data master for all records

Recovery

- what happens if a node goes down?
- data master for some records will be lost
- **one node – the *recovery master* – performs *recovery***
- recovery master collects most recent copy of all records from all nodes
- additional TDB header *record sequence number* determines recentness
- at the end, the recovery master is data master for all records

Recovery

- what happens if a node goes down?
- data master for some records will be lost
- one node – the *recovery master* – performs *recovery*
- **recovery master collects most recent copy of all records from all nodes**
- additional TDB header *record sequence number* determines recentness
- at the end, the recovery master is data master for all records

Recovery

- what happens if a node goes down?
- data master for some records will be lost
- one node – the *recovery master* – performs *recovery*
- recovery master collects most recent copy of all records from all nodes
- additional TDB header *record sequence number* determines recentness
- at the end, the recovery master is data master for all records

Recovery

- what happens if a node goes down?
- data master for some records will be lost
- one node – the *recovery master* – performs *recovery*
- recovery master collects most recent copy of all records from all nodes
- additional TDB header *record sequence number* determines recentness
- at the end, the *recovery master* is data master for all records

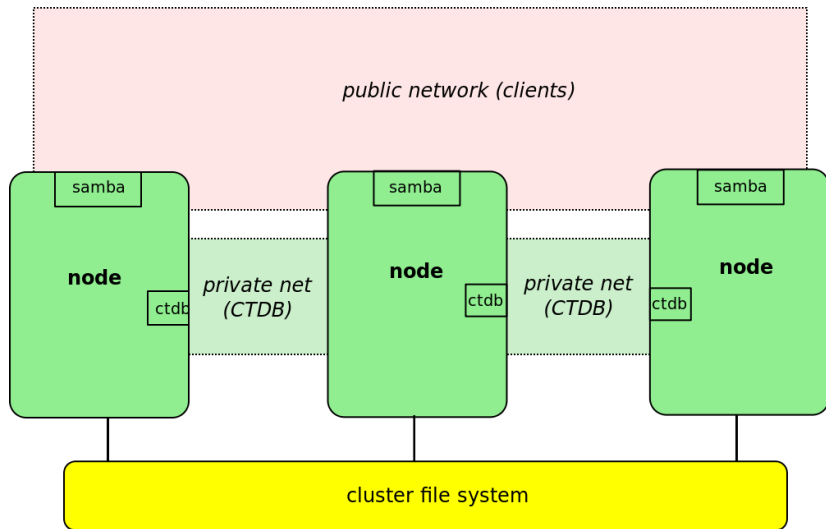
Performance Figures

By Andrew Tridgell and Ronnie Sahlberg, Linux Conf Australia 2009
GPFS file system

32 client smbtoriture NBENCH test

- 1 node: 109 MBytes/sec
- 2 nodes: 210 MBytes/sec
- 3 nodes: 278 MBytes/sec
- 4 nodes: 308 MBytes/sec

CTDB - Basic Setup



CTDB - Configuration

- central file: `/etc/sysconfig/ctdb`
- debian based: `/etc/default/ctdb`
- set `CTDB_RECOVERY_LOCK` for split brain prevention
- fill `/etc/ctdb/nodes` with internal node addresses

example `/etc/ctdb/nodes`

```
10.11.12.10  
10.11.12.11  
10.11.12.12
```

• same file on all nodes

CTDB - Configuration

- central file: `/etc/sysconfig/ctdb`
- debian based: `/etc/default/ctdb`
- set `CTDB_RECOVERY_LOCK` for split brain prevention
- fill `/etc/ctdb/nodes` with internal node addresses

example `/etc/ctdb/nodes`

```
10.11.12.10  
10.11.12.11  
10.11.12.12
```

CTDB - Configuration

- central file: `/etc/sysconfig/ctdb`
- debian based: `/etc/default/ctdb`
- set `CTDB_RECOVERY_LOCK` for split brain prevention
- fill `/etc/ctdb/nodes` with internal node addresses

example `/etc/ctdb/nodes`

```
10.11.12.10
```

```
10.11.12.11
```

```
10.11.12.12
```

CTDB - Configuration

- central file: `/etc/sysconfig/ctdb`
- debian based: `/etc/default/ctdb`
- set `CTDB_RECOVERY_LOCK` for split brain prevention
- fill `/etc/ctdb/nodes` with internal node addresses

example `/etc/ctdb/nodes`

```
10.11.12.10
```

```
10.11.12.11
```

```
10.11.12.12
```

CTDB - Configuration

- central file: `/etc/sysconfig/ctdb`
- debian based: `/etc/default/ctdb`
- set `CTDB_RECOVERY_LOCK` for split brain prevention
- fill `/etc/ctdb/nodes` with internal node addresses

example `/etc/ctdb/nodes`

```
10.11.12.10  
10.11.12.11  
10.11.12.12
```

• same file on all nodes!

CTDB - Configuration

- central file: `/etc/sysconfig/ctdb`
- debian based: `/etc/default/ctdb`
- set `CTDB_RECOVERY_LOCK` for split brain prevention
- fill `/etc/ctdb/nodes` with internal node addresses

example `/etc/ctdb/nodes`

```
10.11.12.10
```

```
10.11.12.11
```

```
10.11.12.12
```

• same file on all nodes!

CTDB - Configuration

- central file: `/etc/sysconfig/ctdb`
- debian based: `/etc/default/ctdb`
- set `CTDB_RECOVERY_LOCK` for split brain prevention
- fill `/etc/ctdb/nodes` with internal node addresses

example `/etc/ctdb/nodes`

```
10.11.12.10  
10.11.12.11  
10.11.12.12
```

- same file on all nodes!

CTDB - Public Addresses

- set `CTDB_PUBLIC_ADDRESSES` in `/etc/sysconfig/ctdb`
- typical value `/etc/ctdb/public_addresses`

example `/etc/ctdb/public_addresses`

```
172.16.17.10/24 eth2
172.16.17.11/24 eth2
172.16.17.12/24 eth2
172.16.17.13/24 eth2
172.16.17.14/24 eth2
172.16.17.15/24 eth2
```

CTDB - Public Addresses

- set CTDB_PUBLIC_ADDRESSES in /etc/sysconfig/ctdb
- typical value /etc/ctdb/public_addresses

example /etc/ctdb/public_addresses

```
172.16.17.10/24 eth2
172.16.17.11/24 eth2
172.16.17.12/24 eth2
172.16.17.13/24 eth2
172.16.17.14/24 eth2
172.16.17.15/24 eth2
```

CTDB - Public Addresses

- set CTDB_PUBLIC_ADDRESSES in /etc/sysconfig/ctdb
- typical value /etc/ctdb/public_addresses

example /etc/ctdb/public_addresses

```
172.16.17.10/24 eth2
172.16.17.11/24 eth2
172.16.17.12/24 eth2
172.16.17.13/24 eth2
172.16.17.14/24 eth2
172.16.17.15/24 eth2
```

• need *not* be the same on all nodes

• need *not* even be present on all nodes (management node)

CTDB - Public Addresses

- set CTDB_PUBLIC_ADDRESSES in /etc/sysconfig/ctdb
- typical value /etc/ctdb/public_addresses

example /etc/ctdb/public_addresses

```
172.16.17.10/24 eth2
172.16.17.11/24 eth2
172.16.17.12/24 eth2
172.16.17.13/24 eth2
172.16.17.14/24 eth2
172.16.17.15/24 eth2
```

• need not be the same on all nodes

• need not even be present on all nodes (management node...)

CTDB - Public Addresses

- set CTDB_PUBLIC_ADDRESSES in /etc/sysconfig/ctdb
- typical value /etc/ctdb/public_addresses

example /etc/ctdb/public_addresses

```
172.16.17.10/24 eth2
172.16.17.11/24 eth2
172.16.17.12/24 eth2
172.16.17.13/24 eth2
172.16.17.14/24 eth2
172.16.17.15/24 eth2
```

- need *not* be the same on all nodes
- need not even be present on all nodes (management node...)

CTDB - Public Addresses

- set CTDB_PUBLIC_ADDRESSES in /etc/sysconfig/ctdb
- typical value /etc/ctdb/public_addresses

example /etc/ctdb/public_addresses

```
172.16.17.10/24 eth2
172.16.17.11/24 eth2
172.16.17.12/24 eth2
172.16.17.13/24 eth2
172.16.17.14/24 eth2
172.16.17.15/24 eth2
```

- need *not* be the same on all nodes
- need not even be present on all nodes (management node...)

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:

● if a client does not yet know the IP has moved

● the server does not have a valid TCP connection to the client

● the node sends the TCP ACK packet to the client (ignoring if

the client has backlogged ACKs) and to the new node

● the new node sends back a RST packet to the client

● the client establishes connection to the new node

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:

• if a server goes down, the IP has moved

• if a client does not have a valid TCP connection to the server

• it will send a TCP ACK packet to the client, indicating it

• is still alive and ready to accept connections

• now the client sends a new packet to the server

• and the connection is re-established

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:

→ client does not yet know the IP has moved

→ client sends a request and fails to get a response → TCP connection timeout

→ client sends a request and fails to get a response → TCP connection timeout

→ client sends a request and fails to get a response → TCP connection timeout

→ client sends a request and fails to get a response → TCP connection timeout

→ client sends a request and fails to get a response → TCP connection timeout

→ client sends a request and fails to get a response → TCP connection timeout

→ client sends a request and fails to get a response → TCP connection timeout

→ client sends a request and fails to get a response → TCP connection timeout

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:
 - client does not yet know the IP has moved
 - new node does not have a valid TCP connection to client
 - new node sends illegal TCP ACK packet to the client (seqnum 0)
 - client sends back correct ACK packet to the *new* node
 - new node sends back a RST packet to the client
 - client re-establishes connection to the new node

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:
 - **client does not yet know the IP has moved**
 - new node does not have a valid TCP connection to client
 - new node sends illegal TCP ACK packet to the client (seqnum 0)
 - client sends back correct ACK packet to the *new* node
 - new node sends back a RST packet to the client
 - client re-establishes connection to the new node

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:
 - client does not yet know the IP has moved
 - **new node does not have a valid TCP connection to client**
 - new node sends illegal TCP ACK packet to the client (seqnum 0)
 - client sends back correct ACK packet to the *new* node
 - new node sends back a RST packet to the client
 - client re-establishes connection to the new node

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:
 - client does not yet know the IP has moved
 - new node does not have a valid TCP connection to client
 - **new node sends illegal TCP ACK packet to the client (seqnum 0)**
 - client sends back correct ACK packet to the *new* node
 - new node sends back a RST packet to the client
 - client re-establishes connection to the new node

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:
 - client does not yet know the IP has moved
 - new node does not have a valid TCP connection to client
 - new node sends illegal TCP ACK packet to the client (seqnum 0)
 - **client sends back correct ACK packet to the new node**
 - new node sends back a RST packet to the client
 - client re-establishes connection to the new node

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:
 - client does not yet know the IP has moved
 - new node does not have a valid TCP connection to client
 - new node sends illegal TCP ACK packet to the client (seqnum 0)
 - client sends back correct ACK packet to the *new* node
 - **new node sends back a RST packet to the client**
 - client re-establishes connection to the new node

IP Failover

- healthy nodes get IP addresses from their public pool
- when a node goes down: public IPs are moved to other nodes
- CTDB distributes the public IPs equally among healthy nodes
- with round robin DNS \Rightarrow HA and load balancing
- speed up client reconnects with *tickle ACKs*:
 - client does not yet know the IP has moved
 - new node does not have a valid TCP connection to client
 - new node sends illegal TCP ACK packet to the client (seqnum 0)
 - client sends back correct ACK packet to the *new* node
 - new node sends back a RST packet to the client
 - **client re-establishes connection to the new node**

CTDB Toolbox

- `ctdb` – control `ctdbd`
- `onnode` – execute programs on selected nodes

CTDB Toolbox

- `ctdb` – control `ctdbd`
- `onnode` – execute programs on selected nodes

ctdb status

```
root@node0:~  
[root@node0 ~]# ctdb status  
Number of nodes:3  
pnn:0 192.168.46.70    OK (THIS NODE)  
pnn:1 192.168.46.71    OK  
pnn:2 192.168.46.72    OK  
Generation:2061920893  
Size:3  
hash:0 lmaster:0  
hash:1 lmaster:1  
hash:2 lmaster:2  
Recovery mode:NORMAL (0)  
Recovery master:1  
[root@node0 ~]#
```

ctdb ip

```
root@node0:~  
[root@node0 ~]# ctdb ip  
Public IPs on node 0  
192.168.45.70 0  
192.168.45.71 1  
192.168.45.72 2  
192.168.45.73 0  
192.168.45.74 1  
192.168.45.75 2  
[root@node0 ~]# █
```

Outline

- 1 Introduction
- 2 Cluster Challenges
 - Introduction
 - Challenges For Samba
- 3 CTDB
 - The CTDB Project
 - CTDB Design
 - Setting Up CTDB
- 4 Clustered Samba
 - Getting Sources and Binaries
 - Clustered File Systems
 - Samba Configuration
 - CTDB manages...
 - Registry Configuration

Getting A Clustered Samba

- in vanilla Samba code since Samba 3.3 (January 2009)
- transaction rewrite in 3.5.2 (March 2010)
- precompiled packages from <http://www.enterprisesamba.org/>
- clustered Samba repository:
`git://git.samba.org/obnox/samba-ctdb.git`
branches: `v3-4-ctdb` and `v3-2-ctdb`
- configure `--with-cluster-support`
- add `idmap_tdb2` to `--with-shared-modules`
- verify that `gpfs.so` is built for GPFS usage

Getting A Clustered Samba

- in vanilla Samba code since Samba 3.3 (January 2009)
- **transaction rewrite in 3.5.2 (March 2010)**
- precompiled packages from <http://www.enterprisesamba.org/>
- clustered Samba repository:
`git://git.samba.org/obnox/samba-ctdb.git`
branches: `v3-4-ctdb` and `v3-2-ctdb`
- configure `--with-cluster-support`
- add `idmap_tdb2` to `--with-shared-modules`
- verify that `gpfs.so` is built for GPFS usage

Getting A Clustered Samba

- in vanilla Samba code since Samba 3.3 (January 2009)
- transaction rewrite in 3.5.2 (March 2010)
- precompiled packages from <http://www.enterprisesamba.org/>
- clustered Samba repository:
`git://git.samba.org/obnox/samba-ctdb.git`
branches: v3-4-ctdb and v3-2-ctdb
- configure `--with-cluster-support`
- add `idmap_tdb2` to `--with-shared-modules`
- verify that `gpfs.so` is built for GPFS usage

Getting A Clustered Samba

- in vanilla Samba code since Samba 3.3 (January 2009)
- transaction rewrite in 3.5.2 (March 2010)
- precompiled packages from <http://www.enterprisesamba.org/>
- clustered Samba repository:
`git://git.samba.org/obnox/samba-ctdb.git`
branches: `v3-4-ctdb` and `v3-2-ctdb`
- `configure --with-cluster-support`
- `add idmap_tdb2 to --with-shared-modules`
- `verify that gpfs.so is built for GPFS usage`

Getting A Clustered Samba

- in vanilla Samba code since Samba 3.3 (January 2009)
- transaction rewrite in 3.5.2 (March 2010)
- precompiled packages from <http://www.enterprisesamba.org/>
- clustered Samba repository:
git://git.samba.org/obnox/samba-ctdb.git
branches: v3-4-ctdb and v3-2-ctdb
- **configure --with-cluster-support**
- add `idmap_tdb2` to `--with-shared-modules`
- verify that `gpfs.so` is built for GPFS usage

Getting A Clustered Samba

- in vanilla Samba code since Samba 3.3 (January 2009)
- transaction rewrite in 3.5.2 (March 2010)
- precompiled packages from <http://www.enterprisesamba.org/>
- clustered Samba repository:
`git://git.samba.org/obnox/samba-ctdb.git`
branches: `v3-4-ctdb` and `v3-2-ctdb`
- configure `--with-cluster-support`
- **add `idmap_tdb2` to `--with-shared-modules`**
- verify that `gpfs.so` is built for GPFS usage

Getting A Clustered Samba

- in vanilla Samba code since Samba 3.3 (January 2009)
- transaction rewrite in 3.5.2 (March 2010)
- precompiled packages from <http://www.enterprisesamba.org/>
- clustered Samba repository:
`git://git.samba.org/obnox/samba-ctdb.git`
branches: `v3-4-ctdb` and `v3-2-ctdb`
- configure `--with-cluster-support`
- add `idmap_tdb2` to `--with-shared-modules`
- **verify that `gpfs.so` is built for GPFS usage**

Clustered File System - Requirements

- file system: black box
- storage: fibre channel, iSCSI, drbd, ...
- simultaneous writes from all nodes
- good to have: coherent POSIX `fcntl` byte range lock support
use ping_pong test to verify

Clustered File System - Requirements

- file system: black box
- storage: fibre channel, iSCSI, drbd, ...
- simultaneous writes from all nodes
- good to have: coherent POSIX `fcntl` byte range lock support
use ping_pong test to verify

Clustered File System - Requirements

- file system: black box
- storage: fibre channel, iSCSI, drbd, ...
- **simulatneous writes from all nodes**
- good to have: coherent POSIX `fcntl` byte range lock support
use ping_pong test to verify

Clustered File System - Requirements

- file system: black box
- storage: fibre channel, iSCSI, drbd, ...
- simultaneous writes from all nodes
- good to have: coherent POSIX `fcntl` byte range lock support
use ping_pong test to verify

Special File Systems

- General Parallel File System GPFS (IBM): OK
- Global File System GFS(2) (Red Hat): OK
- GNU Cluster File System GlusterFS: OK
- Lustre (Sun): OK
- Oracle Cluster File System OCFS(2): OK
- Ceph: ?

Samba Configuration

identical configuration on all nodes

- `clustering = yes`
- `passdb backend = tdbsam`
- `groupdb:backend = tdb`
- `vfs objects = fileid`
`fileid:algorithm = fsid / fsname`
- `idmap backend = tdb2`
- no need to change `private dir`

Samba Configuration

identical configuration on all nodes

- `clustering = yes`
- `passdb backend = tdbsam`
- `groupdb:backend = tdb`
- `vfs objects = fileid`
`fileid:algorithm = fsid / fsname`
- `idmap backend = tdb2`
- `no need to change private dir`

Samba Configuration

identical configuration on all nodes

- `clustering = yes`
- `passdb backend = tdbsam`
- `groupdb:backend = tdb`
- `vfs objects = fileid`
`fileid:algorithm = fsid / fsname`
- `idmap backend = tdb2`
- `no need to change private dir`

Samba Configuration

identical configuration on all nodes

- `clustering = yes`
- `passdb backend = tdbsam`
- `groupdb:backend = tdb`
- `vfs objects = fileid`
`fileid:algorithm = fsid / fsname`
- `idmap backend = tdb2`
- no need to change `private dir`

Samba Configuration

identical configuration on all nodes

- `clustering = yes`
- `passdb backend = tdbsam`
- `groupdb:backend = tdb`
- `vfs objects = fileid`
`fileid:algorithm = fsid / fsname`
- `idmap backend = tdb2`
- `no need to change private dir`

Samba Configuration

identical configuration on all nodes

- `clustering = yes`
- `passdb backend = tdbsam`
- `groupdb:backend = tdb`
- `vfs objects = fileid`
`fileid:algorithm = fsid / fsname`
- `idmap backend = tdb2`
- **no need to change private dir**

example smb.conf

```
[global]
    clustering = yes
    netbios name = smbcluster
    workgroup = mydomain
    security = ads
    passdb backend = tdbsam

    groupdb:backend = tdb

    idmap backend = tdb2
    idmap uid = 1000000-2000000
    idmap gid = 1000000-2000000

    fileid:algorithm = fsname

[share]
    path = /cluster_storage/share
    writeable = yes
    vfs objects = fileid
```

Let's configure Samba on our cluster!

CTDB manages ...

- CTDB can manage several services
- i.e. start, stop, monitor them
- controlled by sysconfig variables `CTDB_MANAGES_SERVICE`
- management performed by scripts in `/etc/ctdb/events.d`
- managed services should be removed from the runlevels
- NOTE: if `CTDB_MANAGES_SAMBA`, do *not* set `interfaces` or `bind interfaces only`

CTDB manages ...

- CTDB can manage several services
- i.e. start, stop, monitor them
- controlled by sysconfig variables `CTDB_MANAGES_SERVICE`
- management performed by scripts in `/etc/ctdb/events.d`
- managed services should be removed from the runlevels
- NOTE: if `CTDB_MANAGES_SAMBA`, do *not* set `interfaces` or `bind interfaces only`

CTDB manages ...

- CTDB can manage several services
- i.e. start, stop, monitor them
- controlled by sysconfig variables `CTDB_MANAGES_SERVICE`
- management performed by scripts in `/etc/ctdb/events.d`
- managed services should be removed from the runlevels
- NOTE: if `CTDB_MANAGES_SAMBA`, do *not* set `interfaces` or `bind interfaces only`

CTDB manages ...

- CTDB can manage several services
- i.e. start, stop, monitor them
- controlled by sysconfig variables `CTDB_MANAGES_SERVICE`
- management performed by scripts in `/etc/ctdb/events.d`
- managed services should be removed from the runlevels
- NOTE: if `CTDB_MANAGES_SAMBA`, do *not* set `interfaces` or `bind interfaces only`

CTDB manages ...

- CTDB can manage several services
- i.e. start, stop, monitor them
- controlled by sysconfig variables `CTDB_MANAGES_SERVICE`
- management performed by scripts in `/etc/ctdb/events.d`
- managed services should be removed from the runlevels
- NOTE: if `CTDB_MANAGES_SAMBA`, do *not* set `interfaces` or `bind interfaces` only

CTDB manages ...

- CTDB can manage several services
- i.e. start, stop, monitor them
- controlled by sysconfig variables `CTDB_MANAGES_SERVICE`
- management performed by scripts in `/etc/ctdb/events.d`
- managed services should be removed from the runlevels
- **NOTE: if `CTDB_MANAGES_SAMBA`, do *not* set interfaces or bind interfaces only**

Registry Configuration

- store config in Samba's registry
- HKLM\Software\Samba\smbconf
- subkey \leftrightarrow section
- value \leftrightarrow parameter
- stored in `registry.tdb` \Rightarrow distributed across cluster by CTDB
- means of easily managing the whole Samba cluster

Registry Configuration

- store config in Samba's registry
- `HKLM\Software\Samba\smbconf`
- subkey \leftrightarrow section
- value \leftrightarrow parameter
- stored in `registry.tdb` \Rightarrow distributed across cluster by CTDB
- means of easily managing the whole Samba cluster

Registry Configuration

- store config in Samba's registry
- HKLM\Software\Samba\smbconf
- **subkey** ↔ **section**
- value ↔ parameter
- stored in `registry.tdb` ⇒ distributed across cluster by CTDB
- means of easily managing the whole Samba cluster

Registry Configuration

- store config in Samba's registry
- HKLM\Software\Samba\smbconf
- subkey \Leftrightarrow section
- value \Leftrightarrow parameter
- stored in `registry.tdb` \Rightarrow distributed across cluster by CTDB
- means of easily managing the whole Samba cluster

Registry Configuration

- store config in Samba's registry
- HKLM\Software\Samba\smbconf
- subkey \Leftrightarrow section
- value \Leftrightarrow parameter
- stored in `registry.tdb` \Rightarrow distributed across cluster by CTDB
- means of easily managing the whole Samba cluster

Registry Configuration

- store config in Samba's registry
- `HKLM\Software\Samba\smbconf`
- subkey \Leftrightarrow section
- value \Leftrightarrow parameter
- stored in `registry.tdb` \Rightarrow distributed across cluster by CTDB
- means of easily managing the whole Samba cluster

Activation of Registry Configuration

- `registry shares = yes`
- `include = registry`
- `config backend = registry`

`smb.conf` for cluster usage

```
[global]
    clustering = yes
    include = registry
```

Activation of Registry Configuration

- registry shares = yes
- include = registry
- config backend = registry

smb.conf for cluster usage

```
[global]
    clustering = yes
    include = registry
```

Activation of Registry Configuration

- registry shares = yes
- include = registry
- config backend = registry

smb.conf for cluster usage

```
[global]
    clustering = yes
    include = registry
```

Activation of Registry Configuration

- registry shares = yes
- include = registry
- config backend = registry

smb.conf for cluster usage

```
[global]
    clustering = yes
    include = registry
```

Activation of Registry Configuration

- registry shares = yes
- include = registry
- config backend = registry

smb.conf for cluster usage

```
[global]
    clustering = yes
    include = registry
```

net conf

manage the whole Samba cluster with one command

```
net conf list           Dump the complete configuration in smb.conf format.
net conf listshares    List the share names.
net conf import        Import configuration from file in smb.conf format.
net conf drop          Delete the complete configuration.
net conf showshare     Show the definition of a share.
net conf addshare      Create a new share.
net conf delshare      Delete a share.
net conf setparm       Store a parameter.
net conf getparm       Retrieve the value of a parameter.
net conf delparm       Delete a parameter.
net conf getincludes   Show the includes of a share definition.
net conf setincludes   Set includes for a share.
net conf delincludes   Delete includes from a share definition.
```

SerNet

SAMBA

net conf

manage the whole Samba cluster with one command

```
net conf list           Dump the complete configuration in smb.conf format.
net conf listshares    List the share names.
net conf import        Import configuration from file in smb.conf format.
net conf drop          Delete the complete configuration.
net conf showshare     Show the definition of a share.
net conf addshare      Create a new share.
net conf delshare      Delete a share.
net conf setparm       Store a parameter.
net conf getparm       Retrieve the value of a parameter.
net conf delparm       Delete a parameter.
net conf getincludes   Show the includes of a share definition.
net conf setincludes   Set includes for a share.
net conf delincludes   Delete includes from a share definition.
```

SerNet

SAMBA

net conf

manage the whole Samba cluster with one command

```
net conf list           Dump the complete configuration in smb.conf format.
net conf listshares    List the share names.
net conf import        Import configuration from file in smb.conf format.
net conf drop          Delete the complete configuration.
net conf showshare     Show the definition of a share.
net conf addshare      Create a new share.
net conf delshare      Delete a share.
net conf setparm       Store a parameter.
net conf getparm       Retrieve the value of a parameter.
net conf delparm       Delete a parameter.
net conf getincludes   Show the includes of a share definition.
net conf setincludes   Set includes for a share.
net conf delincludes   Delete includes from a share definition.
```

net conf

manage the whole Samba cluster with one command

```
net conf list          Dump the complete configuration in smb.conf format.
net conf listshares   List the share names.
net conf import       Import configuration from file in smb.conf format.
net conf drop         Delete the complete configuration.
net conf showshare    Show the definition of a share.
net conf addshare     Create a new share.
net conf delshare     Delete a share.
net conf setparm      Store a parameter.
net conf getparm      Retrieve the value of a parameter.
net conf delparm      Delete a parameter.
net conf getincludes  Show the includes of a share definition.
net conf setincludes  Set includes for a share.
net conf delincludes  Delete includes from a share definition.
```

net conf

manage the whole Samba cluster with one command

```
net conf list          Dump the complete configuration in smb.conf format.
net conf listshares   List the share names.
net conf import       Import configuration from file in smb.conf format.
net conf drop         Delete the complete configuration.
net conf showshare    Show the definition of a share.
net conf addshare     Create a new share.
net conf delshare     Delete a share.
net conf setparm      Store a parameter.
net conf getparm      Retrieve the value of a parameter.
net conf delparm      Delete a parameter.
net conf getincludes  Show the includes of a share definition.
net conf setincludes  Set includes for a share.
net conf delincludes  Delete includes from a share definition.
```

SerNet

SAMBA

net conf

manage the whole Samba cluster with one command

```
net conf list          Dump the complete configuration in smb.conf format.
net conf listshares   List the share names.
net conf import       Import configuration from file in smb.conf format.
net conf drop         Delete the complete configuration.
net conf showshare    Show the definition of a share.
net conf addshare     Create a new share.
net conf delshare     Delete a share.
net conf setparm      Store a parameter.
net conf getparm      Retrieve the value of a parameter.
net conf delparm      Delete a parameter.
net conf getincludes  Show the includes of a share definition.
net conf setincludes  Set includes for a share.
net conf delincludes  Delete includes from a share definition.
```

net conf

manage the whole Samba cluster with one command

```
net conf list           Dump the complete configuration in smb.conf format.
net conf listshares    List the share names.
net conf import        Import configuration from file in smb.conf format.
net conf drop          Delete the complete configuration.
net conf showshare     Show the definition of a share.
net conf addshare      Create a new share.
net conf delshare      Delete a share.
net conf setparm       Store a parameter.
net conf getparm       Retrieve the value of a parameter.
net conf delparm       Delete a parameter.
net conf getincludes   Show the includes of a share definition.
net conf setincludes   Set includes for a share.
net conf delincludes   Delete includes from a share definition.
```

SerNet

SAMBA

net conf

manage the whole Samba cluster with one command

```
net conf list           Dump the complete configuration in smb.conf format.
net conf listshares    List the share names.
net conf import        Import configuration from file in smb.conf format.
net conf drop          Delete the complete configuration.
net conf showshare     Show the definition of a share.
net conf addshare      Create a new share.
net conf delshare      Delete a share.
net conf setparm       Store a parameter.
net conf getparm       Retrieve the value of a parameter.
net conf delparm       Delete a parameter.
net conf getincludes   Show the includes of a share definition.
net conf setincludes   Set includes for a share.
net conf delincludes   Delete includes from a share definition.
```

SerNet

SAMBA

net conf

manage the whole Samba cluster with one command

```
net conf list          Dump the complete configuration in smb.conf format.
net conf listshares   List the share names.
net conf import        Import configuration from file in smb.conf format.
net conf drop          Delete the complete configuration.
net conf showshare     Show the definition of a share.
net conf addshare      Create a new share.
net conf delshare      Delete a share.
net conf setparm       Store a parameter.
net conf getparm       Retrieve the value of a parameter.
net conf delparm       Delete a parameter.
net conf getincludes   Show the includes of a share definition.
net conf setincludes   Set includes for a share.
net conf delincludes   Delete includes from a share definition.
```

SerNet

SAMBA

net conf

manage the whole Samba cluster with one command

```
net conf list          Dump the complete configuration in smb.conf format.
net conf listshares   List the share names.
net conf import       Import configuration from file in smb.conf format.
net conf drop         Delete the complete configuration.
net conf showshare    Show the definition of a share.
net conf addshare     Create a new share.
net conf delshare     Delete a share.
net conf setparm      Store a parameter.
net conf getparm      Retrieve the value of a parameter.
net conf delparm      Delete a parameter.
net conf getincludes  Show the includes of a share definition.
net conf setincludes  Set includes for a share.
net conf delincludes  Delete includes from a share definition.
```

SerNet

SAMBA

Thank you very much!