CTDB Status - Clustered Samba Growing Up

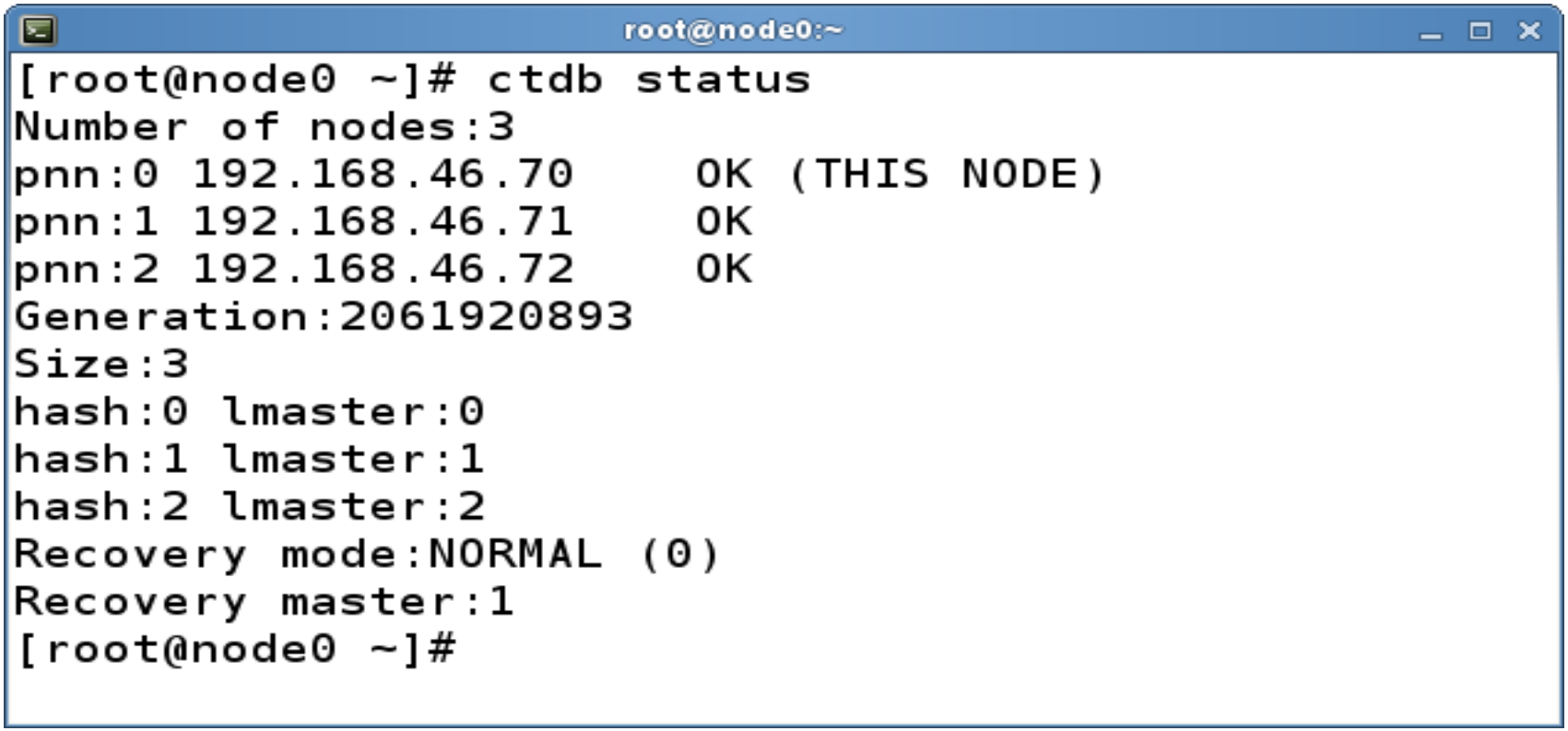
SNIA Storage Developer Conference 2011

Michael Adam

obnox@samba.org

Samba Team / SerNet

2011-09-19


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 ~]#

Thank you very much!

Introduction and History


Clusters
Mandel-Nuss
ENERGIE
FÜR
DEN TAG*

Clustering Samba - Challenges

- Prerequisite: cluster file system
- all-active $\Rightarrow$ all nodes act as **one** CIFS server
- IPC: messaging
- IPC: sessions, connections, open files, locks, ... (TDB databases)
- Persistent data: secrets, registry, id-map, ... (TDB databases)
- TDB: small, fast, key-value database with record locks and memory mapping

Clustering Samba - Challenges

- Prerequisite: cluster file system
- all-active $\Rightarrow$ all nodes act as **one** CIFS server
- IPC: messaging
- IPC: sessions, connections, open files, locks, ... (TDB databases)
- Persistent data: secrets, registry, id-map, ... (TDB databases)
- TDB: small, fast, key-value database with record locks and memory mapping

Clustering Samba - Challenges

- Prerequisite: cluster file system
- all-active $\Rightarrow$ all nodes act as **one** CIFS server
- IPC: messaging
- IPC: sessions, connections, open files, locks, ... (TDB databases)
- Persistent data: secrets, registry, id-map, ... (TDB databases)
- TDB: small, fast, key-value database with record locks and memory mapping

Clustering Samba - Challenges

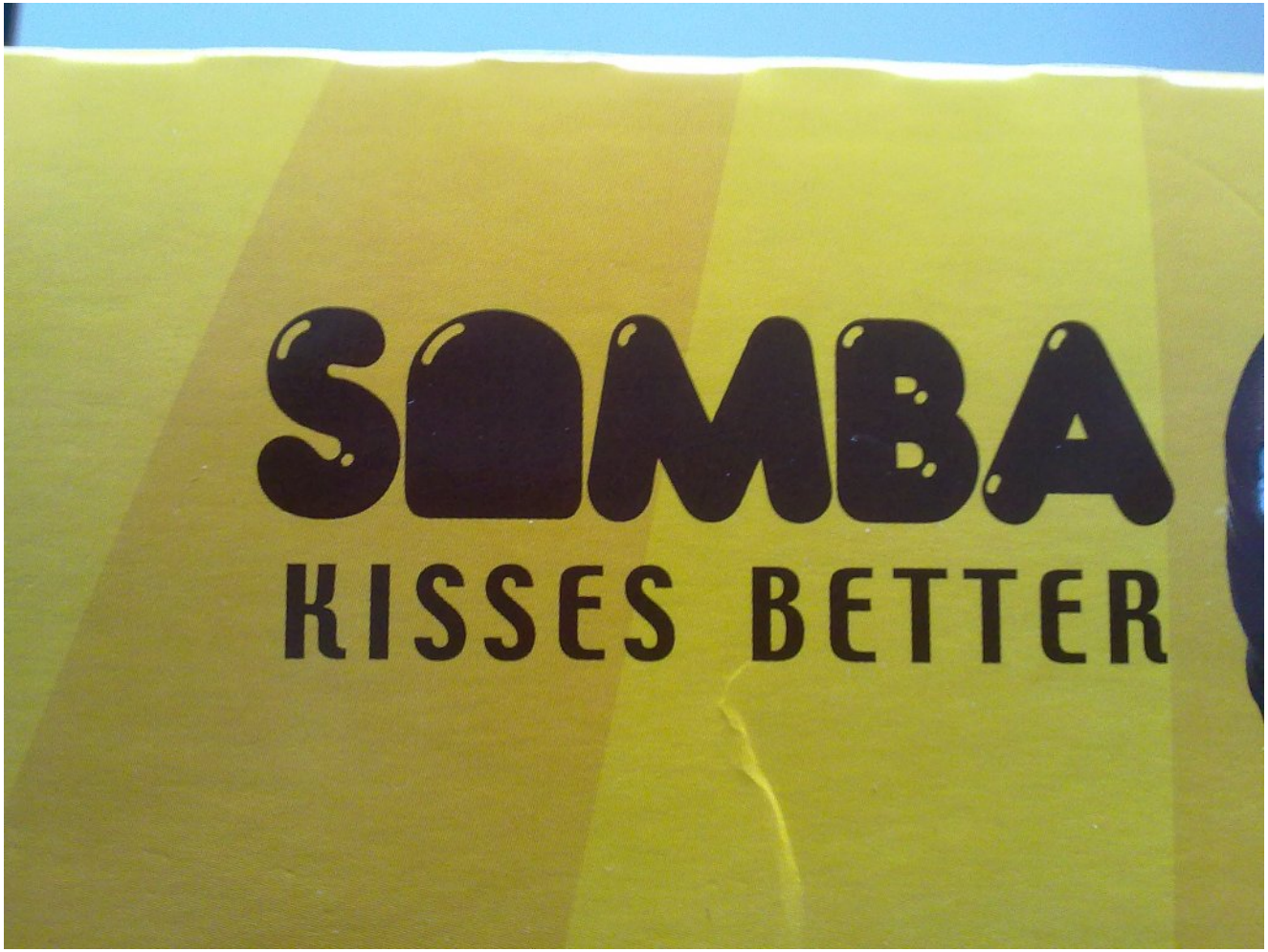
- Prerequisite: cluster file system
- all-active $\Rightarrow$ all nodes act as **one** CIFS server
- IPC: messaging
- IPC: sessions, connections, open files, locks, ... (TDB databases)
- Persistent data: secrets, registry, id-map, ... (TDB databases)
- TDB: small, fast, key-value database with record locks and memory mapping

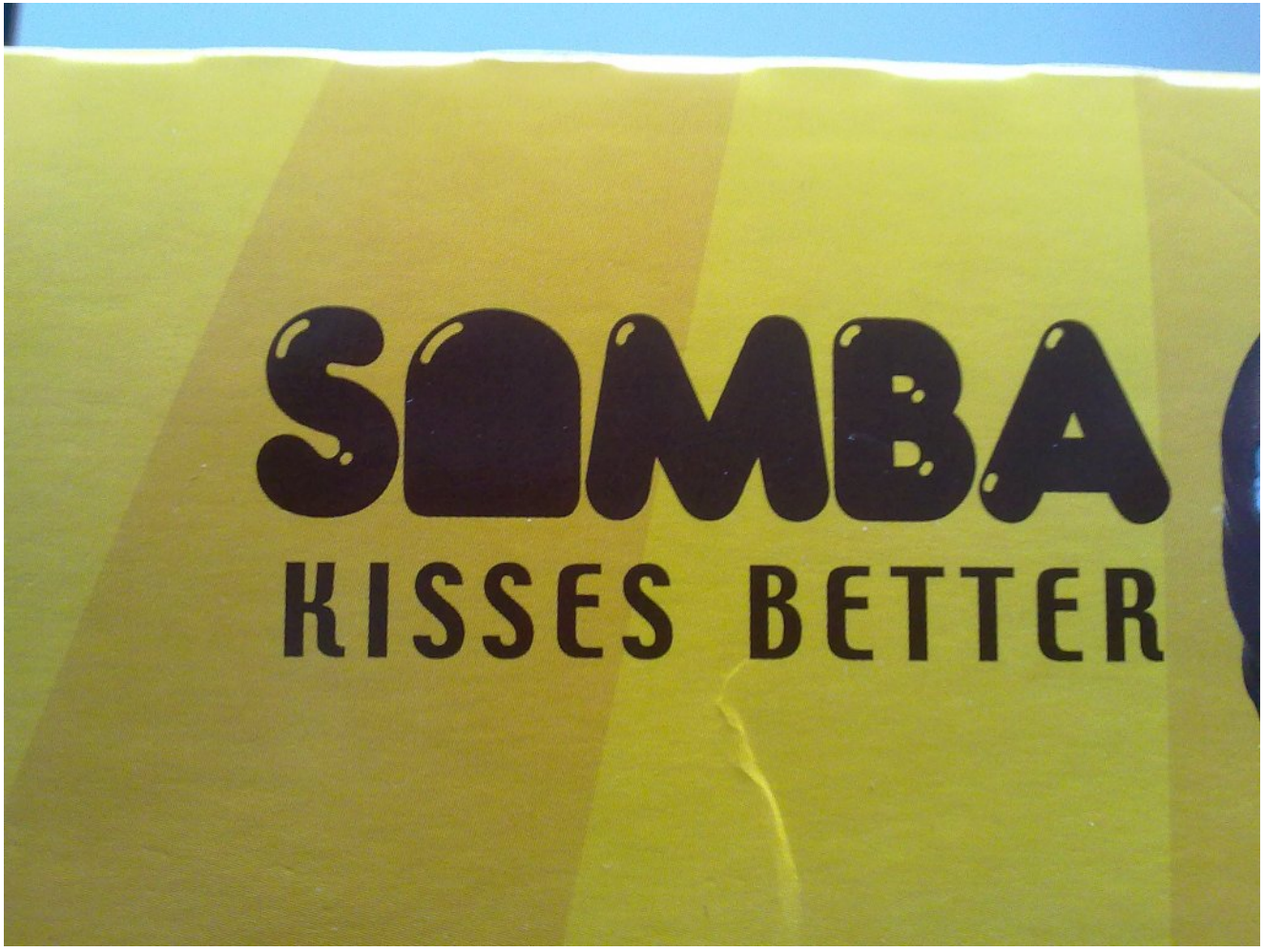
Clustering Samba - Challenges

- Prerequisite: cluster file system
- all-active $\Rightarrow$ all nodes act as **one** CIFS server
- IPC: messaging
- IPC: sessions, connections, open files, locks, ... (TDB databases)
- Persistent data: secrets, registry, id-map, ... (TDB databases)
- TDB: small, fast, key-value database with record locks and memory mapping

Clustering Samba - Challenges

- Prerequisite: cluster file system
- all-active $\Rightarrow$ all nodes act as **one** CIFS server
- IPC: messaging
- IPC: sessions, connections, open files, locks, ... (TDB databases)
- Persistent data: secrets, registry, id-map, ... (TDB databases)
- TDB: small, fast, key-value database with record locks and memory mapping


SAMBA
KISSES BETTER



... with **CTDB** ☺

CTDB ...

- is a very special clustered database implementation (may lose data...)
- is a samba-inter-node-IPC implementation
- is a simple cluster service management software
- makes Samba on a file system cluster appear as a single CIFS/SMB/SMB2 server
- does not require any client changes to access the Samba cluster

CTDB ...

- is a very special clustered database implementation (may lose data...)
- is a samba-inter-node-IPC implementation
- is a simple cluster service management software
- makes Samba on a file system cluster appear as a single CIFS/SMB/SMB2 server
- does not require any client changes to access the Samba cluster

CTDB ...

- is a very special clustered database implementation (may lose data...)
- is a samba-inter-node-IPC implementation
- is a simple cluster service management software
- makes Samba on a file system cluster appear as a single CIFS/SMB/SMB2 server
- does not require any client changes to access the Samba cluster

CTDB ...

- is a very special clustered database implementation (may lose data...)
- is a samba-inter-node-IPC implementation
- is a simple cluster service management software
- makes Samba on a file system cluster appear as a single CIFS/SMB/SMB2 server
- does not require any client changes to access the Samba cluster

CTDB ...

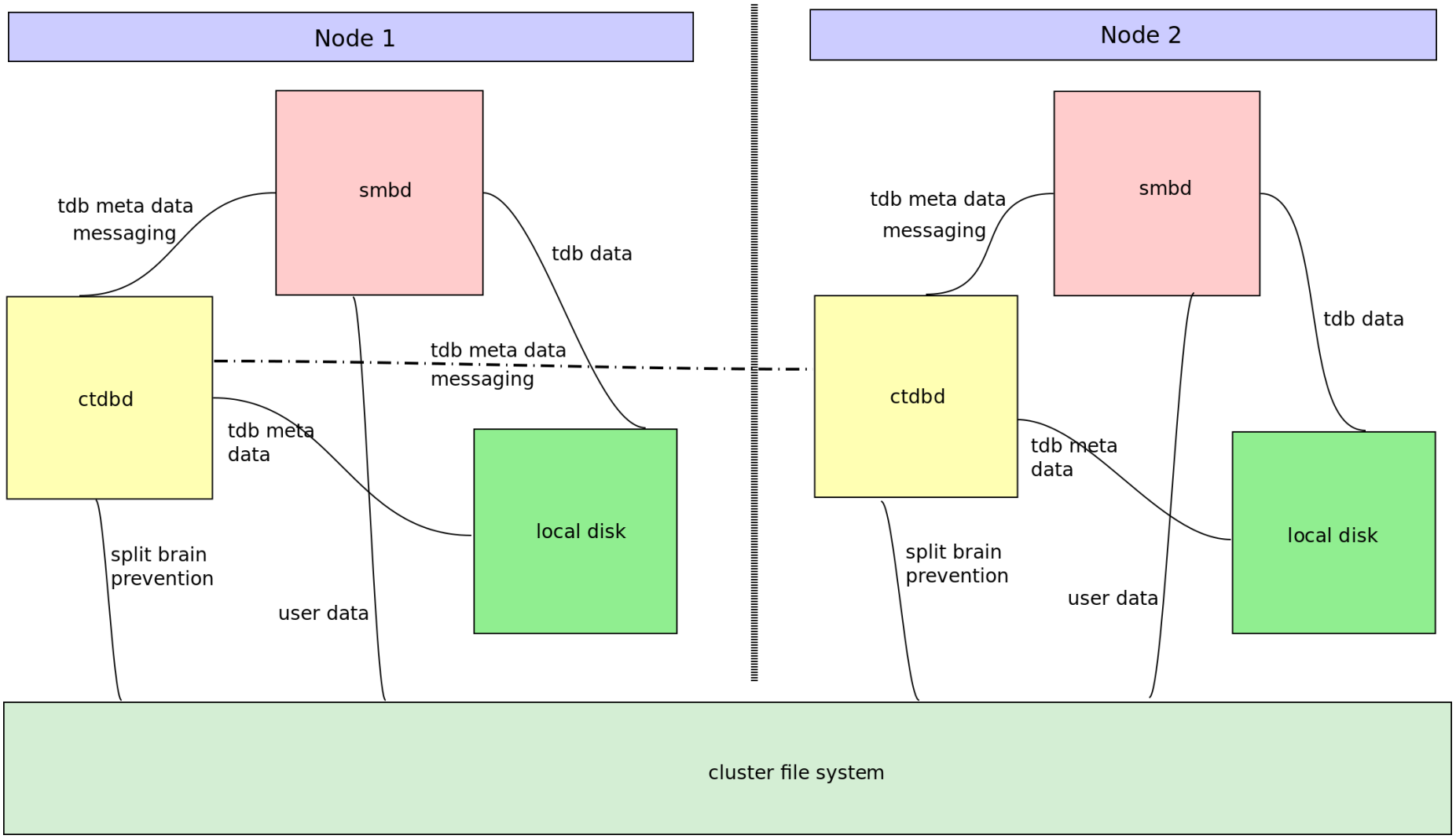
- is a very special clustered database implementation (may lose data...)
- is a samba-inter-node-IPC implementation
- is a simple cluster service management software
- makes Samba on a file system cluster appear as a single CIFS/SMB/SMB2 server
- does not require any client changes to access the Samba cluster

CTDB ...

- is a very special clustered database implementation (may lose data...)
- is a samba-inter-node-IPC implementation
- is a simple cluster service management software
- makes Samba on a file system cluster appear as a single CIFS/SMB/SMB2 server
- does not require any client changes to access the Samba cluster

How CTDB Works

How CTDB Works



CTDB - History

- first prototypes: 2006 (Volker Lendecke, Andrew Tridgell)
- first usable version: 2007 (non-persistent DBs only)
- today: Ronnie Sahlberg maintainer
- support for persistent DBs added 2008/2009

- `http://ctdb.samba.org/`
- code: `git://git.samba.org/ctdb.git`
- branches: `master`, `1.2`, `1.0.114`, ...

CTDB - History

- first prototypes: 2006 (Volker Lendecke, Andrew Tridgell)
- first usable version: 2007 (non-persistent DBs only)
- today: Ronnie Sahlberg maintainer
- support for persistent DBs added 2008/2009

- `http://ctdb.samba.org/`
- code: `git://git.samba.org/ctdb.git`
- branches: `master`, `1.2`, `1.0.114`, ...

CTDB - History

- first prototypes: 2006 (Volker Lendecke, Andrew Tridgell)
- first usable version: 2007 (non-persistent DBs only)
- today: Ronnie Sahlberg maintainer
- support for persistent DBs added 2008/2009

- `http://ctdb.samba.org/`
- code: `git://git.samba.org/ctdb.git`
- branches: `master`, `1.2`, `1.0.114`, ...

CTDB - History

- first prototypes: 2006 (Volker Lendecke, Andrew Tridgell)
- first usable version: 2007 (non-persistent DBs only)
- today: Ronnie Sahlberg maintainer
- support for persistent DBs added 2008/2009

- `http://ctdb.samba.org/`
- code: `git://git.samba.org/ctdb.git`
- branches: `master`, `1.2`, `1.0.114`, ...

CTDB - History

- first prototypes: 2006 (Volker Lendecke, Andrew Tridgell)
- first usable version: 2007 (non-persistent DBs only)
- today: Ronnie Sahlberg maintainer
- support for persistent DBs added 2008/2009

- `http://ctdb.samba.org/`
- code: `git://git.samba.org/ctdb.git`
- branches: `master`, `1.2`, `1.0.114`, ...

CTDB - History

- first prototypes: 2006 (Volker Lendecke, Andrew Tridgell)
- first usable version: 2007 (non-persistent DBs only)
- today: Ronnie Sahlberg maintainer
- support for persistent DBs added 2008/2009

- `http://ctdb.samba.org/`
- code: `git://git.samba.org/ctdb.git`
- branches: `master`, `1.2`, `1.0.114`, ...

CTDB - History

- first prototypes: 2006 (Volker Lendecke, Andrew Tridgell)
- first usable version: 2007 (non-persistent DBs only)
- today: Ronnie Sahlberg maintainer
- support for persistent DBs added 2008/2009

- `http://ctdb.samba.org/`
- code: `git://git.samba.org/ctdb.git`
- branches: `master`, `1.2`, `1.0.114`, ...

CTDB - History

- first prototypes: 2006 (Volker Lendecke, Andrew Tridgell)
- first usable version: 2007 (non-persistent DBs only)
- today: Ronnie Sahlberg maintainer
- support for persistent DBs added 2008/2009

- `http://ctdb.samba.org/`
- code: `git://git.samba.org/ctdb.git`
- branches: `master`, `1.2`, `1.0.114`, ...

Recent and Current Improvements

- vacuuming
- persistent transactions
- samba persistent db performance tuning
- tools
- read-only record copies
- further projects

Vacuuming

Vacuuming

- purpose: garbage collection of deleted records
- was not working well at all:
- under certain workloads, databases grew despite vacuuming
- vacuuming child even crashed under certain conditions
- $\Rightarrow$ whole clusters went on strike

Vacuuming

- purpose: garbage collection of deleted records
- was not working well at all:
- under certain workloads, databases grew despite vacuuming
- vacuuming child even crashed under certain conditions
- $\Rightarrow$ whole clusters went on strike

Vacuuming

- purpose: garbage collection of deleted records
- was not working well at all:
- under certain workloads, databases grew despite vacuuming
- vacuuming child even crashed under certain conditions
- $\Rightarrow$ whole clusters went on strike

Vacuuming

- purpose: garbage collection of deleted records
- was not working well at all:
- under certain workloads, databases grew despite vacuuming
- vacuuming child even crashed under certain conditions
- $\Rightarrow$ whole clusters went on strike

Vacuuming

- purpose: garbage collection of deleted records
- was not working well at all:
- under certain workloads, databases grew despite vacuuming
- vacuuming child even crashed under certain conditions
- $\Rightarrow$ whole clusters went on strike

Vacuuming ☹

Vacuuming Fixes ☺

- Rewritten by Stefan Metzmacher, Michael Adam (2010/2011)
- available since versions `1.0.114.2` and `1.2.25`
- in-memory lists of records scheduled for deletion
- samba enhancement to schedule upon delete operation
- db traverse vacuuming as fallback

Vacuuming Fixes ☺

- Rewritten by Stefan Metzmacher, Michael Adam (2010/2011)
- available since versions `1.0.114.2` and `1.2.25`
- in-memory lists of records scheduled for deletion
- samba enhancement to schedule upon delete operation
- db traverse vacuuming as fallback

Vacuuming Fixes ☺

- Rewritten by Stefan Metzmacher, Michael Adam (2010/2011)
- available since versions `1.0.114.2` and `1.2.25`
- in-memory lists of records scheduled for deletion
- samba enhancement to schedule upon delete operation
- db traverse vacuuming as fallback

Vacuuming Fixes ☺

- Rewritten by Stefan Metzmacher, Michael Adam (2010/2011)
- available since versions `1.0.114.2` and `1.2.25`
- in-memory lists of records scheduled for deletion
- samba enhancement to schedule upon delete operation
- db traverse vacuuming as fallback

Vacuuming Fixes ☺

- Rewritten by Stefan Metzmacher, Michael Adam (2010/2011)
- available since versions `1.0.114.2` and `1.2.25`
- in-memory lists of records scheduled for deletion
- samba enhancement to schedule upon delete operation
- db traverse vacuuming as fallback

Vacuuming Fixes ☺

- Rewritten by Stefan Metzmacher, Michael Adam (2010/2011)
- available since versions `1.0.114.2` and `1.2.25`
- in-memory lists of records scheduled for deletion
- samba enhancement to schedule upon delete operation
- db traverse vacuuming as fallback

Vacuuming TODOs

- improve notion of active vs. deleted record
- several internal polishing tasks

Vacuuming TODOs

- improve notion of active vs. deleted record
- several internal polishing tasks

Vacuuming TODOs

- improve notion of active vs. deleted record
- several internal polishing tasks

Persistent Transactions

Persistent Transactions

- race conditions
- lack of global state
- $\Rightarrow$ data corruption

Persistent Transactions

- race conditions
- lack of global state
- $\Rightarrow$ data corruption

Persistent Transactions

- race conditions
- lack of global state
- $\Rightarrow$ data corruption

Persistent Transactions

- race conditions
- lack of global state
- $\Rightarrow$ data corruption

Persistent Transactions ☹

Persistent Transaction Fixes ☺

- Volker Lendecke, Stefan Metzmacher, Michael Adam
- 2009/2010:
  - created global state with global lock (g_lock.tdb)
  - rewritten transaction code (track all changes in memory until commit)
- 2011:
  - further fixes of race conditions with CTDB recoveries

Persistent Transaction Fixes ☺

- Volker Lendecke, Stefan Metzmacher, Michael Adam
- 2009/2010:
  - created global state with global lock (g_lock)
  - rewritten transaction code (track all changes in memory until commit)
- 2011:
  - further fixes of race conditions with CTDB recoveries

Persistent Transaction Fixes ☺

- Volker Lendecke, Stefan Metzmacher, Michael Adam
- 2009/2010:
  - created global state with global lock (`g_lock`)
  - rewritten transaction code (track all changes in memory until commit)
- 2011:
  - further fixes of race conditions with CTDB recoveries

Persistent Transaction Fixes ☺

- Volker Lendecke, Stefan Metzmacher, Michael Adam
- 2009/2010:
  - created global state with global lock (`g_lock`)
  - rewritten transaction code (track all changes in memory until commit)
- 2011:
  - further fixes of race conditions with CTDB recoveries

Persistent Transaction Fixes ☺

- Volker Lendecke, Stefan Metzmacher, Michael Adam
- 2009/2010:
  - created global state with global lock (`g_lock`)
  - rewritten transaction code (track all changes in memory until commit)
- 2011:
  - further fixes of race conditions with CTDB recoveries

Persistent Transaction Fixes ☺

- Volker Lendecke, Stefan Metzmacher, Michael Adam
- 2009/2010:
  - created global state with global lock (`g_lock`)
  - rewritten transaction code (track all changes in memory until commit)
- 2011:
  - further fixes of race conditions with CTDB recoveries

Persistent Transaction Fixes ☺

- Volker Lendecke, Stefan Metzmacher, Michael Adam
- 2009/2010:
  - created global state with global lock (`g_lock`)
  - rewritten transaction code (track all changes in memory until commit)
- 2011:
  - further fixes of race conditions with CTDB recoveries

Persistent Databases: TODOs

- implement transactions inside CTDB (currently much of the logic is in samba client code)
- correctly handle delete record operations
- implement recoveries for persistent databases differently

Samba DB Tuning

Samba Persistent DB Performance Tuning

- more frequently used persistent DBs too slow in a cluster
- e.g. idmap and registry
- all write operations to persistent database are protected by transactions
- especially expensive in a cluster

- id-mapping code was rewritten to make creation of ID mappings atomic (2010/2011, Michael Adam)

- registry improvements are currently WIP (Gregor Beck, Michael Adam)

Samba Persistent DB Performance Tuning

- more frequently used persistent DBs too slow in a cluster
- e.g. idmap and registry
- all write operations to persistent database are protected by transactions
- especially expensive in a cluster
- id-mapping code was rewritten to make creation of ID mappings atomic (2010/2011, Michael Adam)
- registry improvements are currently WIP (Gregor Beck, Michael Adam)

Samba Persistent DB Performance Tuning

- more frequently used persistent DBs too slow in a cluster
- e.g. idmap and registry
- all write operations to persistent database are protected by transactions
- especially expensive in a cluster
- id-mapping code was rewritten to make creation of ID mappings atomic (2010/2011, Michael Adam)
- registry improvements are currently WIP (Gregor Beck, Michael Adam)

Samba Persistent DB Performance Tuning

- more frequently used persistent DBs too slow in a cluster
- e.g. idmap and registry
- all write operations to persistent database are protected by transactions
- especially expensive in a cluster
- id-mapping code was rewritten to make creation of ID mappings atomic (2010/2011, Michael Adam)
- registry improvements are currently WIP (Gregor Beck, Michael Adam)

Samba Persistent DB Performance Tuning

- more frequently used persistent DBs too slow in a cluster
- e.g. idmap and registry
- all write operations to persistent database are protected by transactions
- especially expensive in a cluster
- id-mapping code was rewritten to make creation of ID mappings atomic (2010/2011, Michael Adam)
- registry improvements are currently WIP (Gregor Beck, Michael Adam)

Samba Persistent DB Performance Tuning

- more frequently used persistent DBs too slow in a cluster
- e.g. idmap and registry
- all write operations to persistent database are protected by transactions
- especially expensive in a cluster
- id-mapping code was rewritten to make creation of ID mappings atomic (2010/2011, Michael Adam)
- registry improvements are currently WIP (Gregor Beck, Michael Adam)

Samba Persistent DB Performance Tuning

- more frequently used persistent DBs too slow in a cluster
- e.g. idmap and registry
- all write operations to persistent database are protected by transactions
- especially expensive in a cluster
- id-mapping code was rewritten to make creation of ID mappings atomic (2010/2011, Michael Adam)
- registry improvements are currently WIP (Gregor Beck, Michael Adam)

Tools

Repair / Check / Convert Tools

- ltdbtool (dump, convert)
- net idmap check (samba)
- net registry check (samba)
- mostly done by Gregor Beck

Repair / Check / Convert Tools

- ltdbtool (dump, convert)
- net idmap check (samba)
- net registry check (samba)
- mostly done by Gregor Beck

Repair / Check / Convert Tools

- ltdbtool (dump, convert)
- net idmap check (samba)
- net registry check (samba)
- mostly done by Gregor Beck

Repair / Check / Convert Tools

- ltdbtool (dump, convert)
- net idmap check (samba)
- net registry check (samba)
- mostly done by Gregor Beck

Repair / Check / Convert Tools

- ltdbtool (dump, convert)
- net idmap check (samba)
- net registry check (samba)
- mostly done by Gregor Beck

Read-Only Record Copies

Read-Only Record Copies

- Problem: record ping-pong when accessing the same file from multiple nodes.
- Current code: migrates the record (e.g. brlock) each time it is accessed.
- $\Rightarrow$ very bad performance under these workloads
- Ronnie Sahlberg and Rusty Russell are currently working on the implementation

Read-Only Record Copies

- Problem: record ping-pong when accessing the same file from multiple nodes.
- Current code: migrates the record (e.g. brlock) each time it is accessed.
- $\Rightarrow$ very bad performance under these workloads
- Ronnie Sahlberg and Rusty Russell are currently working on the implementation

Read-Only Record Copies

- Problem: record ping-pong when accessing the same file from multiple nodes.
- Current code: migrates the record (e.g. brlock) each time it is accessed.
- $\Rightarrow$ very bad performance under these workloads
- Ronnie Sahlberg and Rusty Russell are currently working on the implementation

Read-Only Record Copies

- Problem: record ping-pong when accessing the same file from multiple nodes.
- Current code: migrates the record (e.g. brlock) each time it is accessed.
- $\Rightarrow$ very bad performance under these workloads
- Ronnie Sahlberg and Rusty Russell are currently working on the implementation

Read-Only Record Copies

- Problem: record ping-pong when accessing the same file from multiple nodes.
- Current code: migrates the record (e.g. brlock) each time it is accessed.
- $\Rightarrow$ very bad performance under these workloads
- Ronnie Sahlberg and Rusty Russell are currently working on the implementation

Further Projects

Further Projects

- CTDB client library `libctdb`
  (Ronnie Sahlberg, Rusty Russell, Volker Lendecke)
- exploit SMB2 durable file handles
  (Stefan Metzmacher, (Michael Adam))
- exploit new SMB 2.1 and 2.2 features
  large MTU, multi-credit, leases, re-auth, multi-channel, ...
  (Stefan Metzmacher, (Michael Adam))

Further Projects

- CTDB client library `libctdb`
  (Ronnie Sahlberg, Rusty Russell, Volker Lendecke)
- exploit SMB2 durable file handles
  (Stefan Metzmacher, (Michael Adam))
- exploit new SMB 2.1 and 2.2 features
  large MTU, multi-credit, leases, re-auth, multi-channel, ...
  (Stefan Metzmacher, (Michael Adam))

Further Projects

- CTDB client library `libctdb`
  (Ronnie Sahlberg, Rusty Russell, Volker Lendecke)
- exploit SMB2 durable file handles
  (Stefan Metzmacher, (Michael Adam))
- exploit new SMB 2.1 and 2.2 features
  large MTU, multi-credit, leases, re-auth, multi-channel, ...
  (Stefan Metzmacher, (Michael Adam))

Further Projects

- CTDB client library `libctdb`
  (Ronnie Sahlberg, Rusty Russell, Volker Lendecke)
- exploit SMB2 durable file handles
  (Stefan Metzmacher, (Michael Adam))
- exploit new SMB 2.1 and 2.2 features
  large MTU, multi-credit, leases, re-auth, multi-channel, ...
  (Stefan Metzmacher, (Michael Adam))

Management / Integration

CTDB as cluster manager

- manages services (samba/winbind/nfs/apache/...): start/stop/monitor
- pluggable extensible event script architecture (`/etc/ctdb/events.d/`)
- handles IP (re)allocation on public network: fail-over/fail-back
- tickles clients to reconnect in case of fail-over

- When this was created, Linux cluster stack did not have all-active.
- But nowadays, pacemaker is getting more popular in distributions.
- All of the above CTDB features are optional.

CTDB as cluster manager

- manages services (samba/winbind/nfs/apache/...): start/stop/monitor
- pluggable extensible event script architecture (`/etc/ctdb/events.d/`)
- handles IP (re)allocation on public network: fail-over/fail-back
- tickles clients to reconnect in case of fail-over

- When this was created, Linux cluster stack did not have all-active.
- But nowadays, pacemaker is getting more popular in distributions.
- All of the above CTDB features are optional.

CTDB as cluster manager

- manages services (samba/winbind/nfs/apache/...): start/stop/monitor
- pluggable extensible event script architecture (`/etc/ctdb/events.d/`)
- handles IP (re)allocation on public network: fail-over/fail-back
- tickles clients to reconnect in case of fail-over
- When this was created, Linux cluster stack did not have all-active.
- But nowadays, pacemaker is getting more popular in distributions.
- All of the above CTDB features are optional.

CTDB as cluster manager

- manages services (samba/winbind/nfs/apache/...): start/stop/monitor
- pluggable extensible event script architecture (`/etc/ctdb/events.d/`)
- handles IP (re)allocation on public network: fail-over/fail-back
- tickles clients to reconnect in case of fail-over

- When this was created, Linux cluster stack did not have all-active.
- But nowadays, pacemaker is getting more popular in distributions.
- All of the above CTDB features are optional.

CTDB as cluster manager

- manages services (samba/winbind/nfs/apache/...): start/stop/monitor
- pluggable extensible event script architecture (`/etc/ctdb/events.d/`)
- handles IP (re)allocation on public network: fail-over/fail-back
- tickles clients to reconnect in case of fail-over

- When this was created, Linux cluster stack did not have all-active.
- But nowadays, pacemaker is getting more popular in distributions.
- All of the above CTDB features are optional.

CTDB as cluster manager

- manages services (samba/winbind/nfs/apache/...): start/stop/monitor
- pluggable extensible event script architecture (`/etc/ctdb/events.d/`)
- handles IP (re)allocation on public network: fail-over/fail-back
- tickles clients to reconnect in case of fail-over

- When this was created, Linux cluster stack did not have all-active.
- But nowadays, pacemaker is getting more popular in distributions.
- All of the above CTDB features are optional.

CTDB as cluster manager

- manages services (samba/winbind/nfs/apache/...): start/stop/monitor
- pluggable extensible event script architecture (`/etc/ctdb/events.d/`)
- handles IP (re)allocation on public network: fail-over/fail-back
- tickles clients to reconnect in case of fail-over

- When this was created, Linux cluster stack did not have all-active.
- But nowadays, pacemaker is getting more popular in distributions.
- All of the above CTDB features are optional.

CTDB as cluster manager

- manages services (samba/winbind/nfs/apache/...): start/stop/monitor
- pluggable extensible event script architecture (`/etc/ctdb/events.d/`)
- handles IP (re)allocation on public network: fail-over/fail-back
- tickles clients to reconnect in case of fail-over

- When this was created, Linux cluster stack did not have all-active.
- But nowadays, pacemaker is getting more popular in distributions.
- All of the above CTDB features are optional.

Integrating CTDB and Samba

Independently of Linux cluster stack

- CTDB manages samba
- CTDB manages winbindd
- CTDB manges public IP addresses

As managed resources

- CTDB does not manage samba, winbind nor public IPs
- CTDB only provides clustered TDB services
- Linux cluster suite (pacemaker) manages CTDB and Samba and Winbind
- Resource dependency: Cluster FS ⇒ CTDB ⇒ winbindd ⇒ samba

Integrating CTDB and Samba

Two choices:

Independently of Linux cluster stack

- CTDB manages samba
- CTDB manages winbindd
- CTDB manges public IP addresses

As managed resources

- CTDB does not manage samba, winbind nor public IPs
- CTDB only provides clustered TDB services
- Linux cluster suite (pacemaker) manages CTDB and Samba and Winbind
- Resource dependency: Cluster FS ⇒ CTDB ⇒ winbindd ⇒ samba

Integrating CTDB and Samba

Two choices:

Independently of Linux cluster stack

- CTDB manages samba
- CTDB manages winbindd
- CTDB manges public IP addresses

As managed resources

- CTDB does not manage samba, winbind nor public IPs
- CTDB only provides clustered TDB services
- Linux cluster suite (pacemaker) manages CTDB and Samba and Winbind
- Resource dependency: Cluster FS ⇒ CTDB ⇒ winbindd ⇒ samba

Integrating CTDB and Samba

Two choices:

Independently of Linux cluster stack

- CTDB manages samba
- CTDB manages winbindd
- CTDB manges public IP addresses

As managed resources

- CTDB does not manage samba, winbind nor public IPs
- CTDB only provides clustered TDB services
- Linux cluster suite (pacemaker) manages CTDB and Samba and Winbind
- Resource dependency: Cluster FS ⇒ CTDB ⇒ winbindd ⇒ samba

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd, too
  - but there is a mode for CTDB to run as clustered TDB only (!)
  - matching resource agents for samba and winbindd still missing

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd too
  - but there is a mode for CTDB to run as clustered TDB only
  - matching resource agents for samba and winbindd still needed

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd, too
  - but there is a mode for CTDB to run as clustered NDB only (!)
  - matching resource agents for samba and winbindd still needed

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd, too
  - but there is a mode for CTDB to run as clustered NDB only
  - matching resource agents for samba and winbindd still missing

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd, too
  - but there is a mode for CTDB to run as clustered IDMAP only
  - matching resource agents for samba and winbindd will exist

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd ☹
  - but there is a mode for CTDB to run as clustered TDB only ☺
  - matching resource agents for samba and winbindd still needed ...

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd ☹
  - but there is a mode for CTDB to run as clustered TDB only ☺
  - matching resource agents for samba and winbindd still needed ...

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd ☹
  - but there is a mode for CTDB to run as clustered TDB only ☺
  - matching resource agents for samba and winbindd still needed ...

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd ☹
  - but there is a mode for CTDB to run as clustered TDB only ☺
  - matching resource agents for samba and winbindd still needed ...

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd ☹
  - but there is a mode for CTDB to run as clustered TDB only ☺
  - matching resource agents for samba and winbindd still needed ...

Integration: Status Quo

- Red Hat
  - starts using pacemaker
  - currently (RHEL 6) CTDB is run as service managing samba, not cluster resource
  - Samba/CTDB/GFS howto
- SuSE
  - pacemaker in use
  - CTDB run as cluster resource
  - currently CTDB manages samba and winbindd ☹
  - but there is a mode for CTDB to run as clustered TDB only ☺
  - matching resource agents for samba and winbindd still needed ...

Thank you very much!

(Now really... ☺)