

Clustered CIFS For Everybody Clustering Samba With CTDB

LinuxTag 2009

Michael Adam

`obnox@samba.org`

2009-06-24

Contents

1	Cluster Challenges	2
1.1	The Ideas	2
1.2	Challenges For Samba	2
2	CTDB	4
2.1	The CTDB Project	4
2.2	CTDB Design	4
2.3	Clustered File Systems	6
2.4	Setting Up CTDB	6
3	Clustered Samba	8
3.1	Configuration Options	8
3.2	Registry Configuration	9

1 Cluster Challenges

1.1 The Ideas

Basic Ideas

- storage tends to become too small
- ⇒ use a SAN and volume based file systems
- services using the storage tend to become too slow
- ⇒ cluster these services (all-active)
- this clustering makes use of a *clustered file system*
- quite common for web and database servers
- how about offering the file system itself via CIFS or NFS in a clustered fashion?
- i.e. turn your SAN in a clustered NAS...
- Windows servers don't offer this form of clustering
- Samba now does! With the help of CTDB.

1.2 Challenges For Samba

About Samba

- open source software (GPL) started in 1992
- file and print services for windows clients on unix systems
- makes unix host appear in the windows network neighborhood
- member file server in windows NT and Active Directory domains
- can act as NT-style domain controller (logon server)
- deployed widely in production environments
- Samba4 (alpha): Active Directory domain controller
- “Franky”: glue the good parts together

Challenges For Samba

- samba daemons on cluster nodes need to act as *one* CIFS server:
 - view of file ownership
 - windows file lock coherence
- samba instances need to share certain persistent data:
 - user database (`passdb.tdb`)

- join information (`secrets.tdb`)
- id mapping tables (`winbindd_idmap.tdb`)
- further share volatile session data:
 - SMB sessions (`sessionid.tdb`)
 - share connections (`connections.tdb`)
 - share modes (`locking.tdb`)
 - byte range locks (`brlock.tdb`)
- messaging

TDBs And Clustering

- most problems are about distributing TDBs in the cluster
- TDB: small, fast Berkeley-DB-style database with record locks and memory mapping
- persistent TDBs vs. "normal" (volatile) TDBs
- TDB R/W performance critical for Samba performance
- TDB R/W ops: excessive use of POSIX `fcntl` byte range locks
- `fcntl` locks are *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*
- \Rightarrow A more clever approach is needed.

Goals

- Cluster Samba so that:
 - One node is not slower than an unclustered Samba server.
 - $n + 1$ nodes are faster than n nodes.
- This requires a clustered TDB implementation ...
- ... and a clustered messaging solution.

$\Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow$ *CTDB :-)*

2 CTDB

2.1 The CTDB Project

The CTDB Project

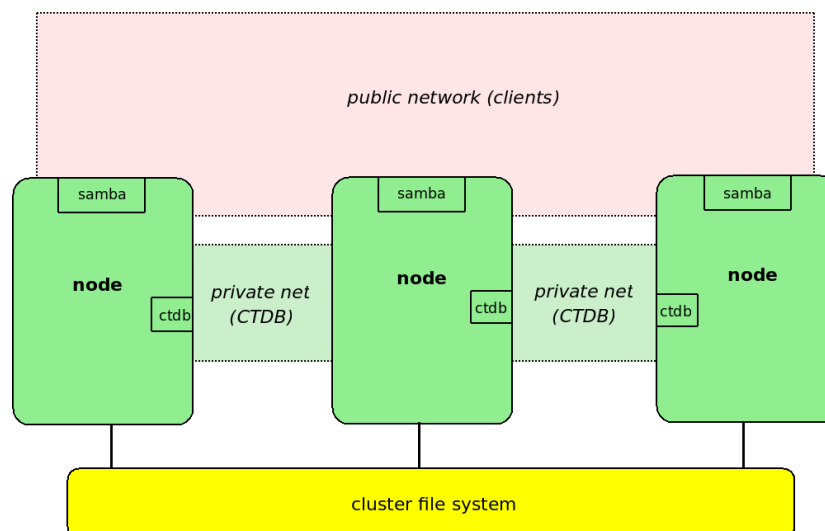
- started in 2006
- first prototypes by Volker Lendecke, Andrew Tridgell, ...
- first usable version of CTDB: April 2007
- meanwhile: Ronnie Sahlberg project maintainer
- `git://git.samba.org/sahlberg/ctdb.git`
- `http://ctdb.samba.org/packages/` (RPMs, Sources)
- Currently runs on Linux and AIX

2.2 CTDB Design

CTDB Design

- one daemon `ctdbd` on each node
- `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
- persistent and normal TDBs are handled differently
- CTDB distributes public IPs across cluster nodes
- management features: Samba, NFS and other services

CTDB - Basic Setup



persistent TDBs

- complete copy in LTDB on all nodes
- read ops directly on LTDB (fast)
- write ops automatically distributed to all nodes (slow)
- ⇒ data integrity and read performance guaranteed

normal TDBs

- some records may be lost
- keep only needed records in LTDB
- most records are only ever accessed by one node
- only one node has current copy of a record (*data master*)
- before accessing a record, switch data master role
- ⇒ good R/W performance, and sufficient data integrity

Performance Figures

By Andrew Tridgell and Ronnie Sahlberg, Linux Conf Australia 2008

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

Recovery

- what happens when a node goes down?
- data master for some records will be lost
- one node (*recovery master*) performs *recovery*
- recovery master collects most recent copy of all records from all nodes
- at the end, the recovery master is data master for all records

Recovery Election / Recovery Lock

- recovery master is determined by an election process
- election involves *recovery lock* file on shared storage
 - nodes compete with POSIX `fcntl` byte range locks
- finally, the new recovery master holds lock on the recovery lock file
- ⇒ CTDB requires POSIX `fcntl` lock support in the cluster FS
- ⇒ CTDB has no split brain (other than the file system)

2.3 Clustered File Systems

Clustered File System - Requirements

- file system : black box
- storage: fibre channel, iSCSI, drbd, ...
- simultaneous writes from all nodes
- 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 (new!)

2.4 Setting Up CTDB

CTDB - Configuration

- central file: `/etc/sysconfig/ctdb`
- must set: `CTDB_RECOVERY_LOCK`
- fill `/etc/ctdb/nodes` with internal addresses
same file on all nodes!

example `/etc/ctdb/nodes`

```
10.0.0.10
10.0.0.11
10.0.0.12
10.0.0.13
```

CTDB - Public Addresses

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

example `/etc/ctdb/public_addresses`

```
192.168.111.10/24 eth0
192.168.111.11/24 eth0
192.168.111.12/24 eth0
192.168.111.13/24 eth0
```

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

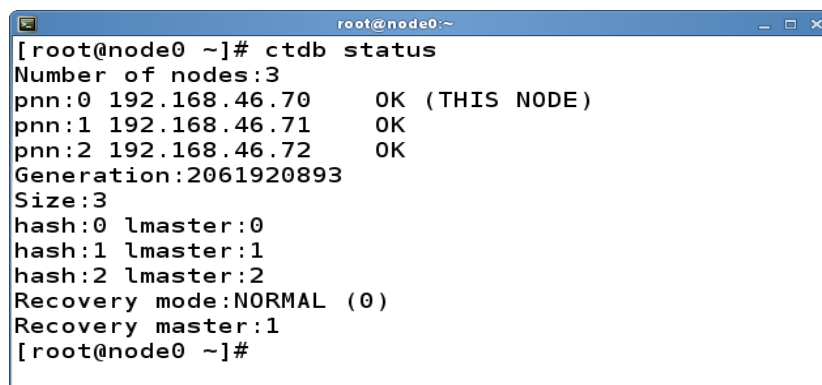
IP Failover

- HEALTHY nodes get IP addresses from their public pool
- when a node goes down: public IPs 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*!

CTDB Toolbox

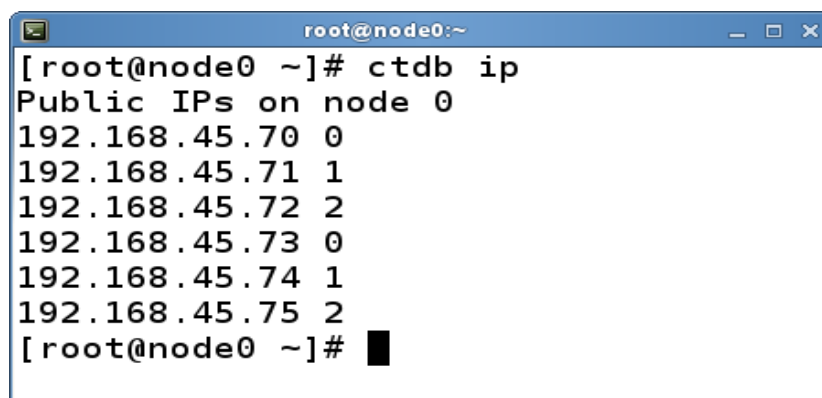
- `ctdb` – control `ctdbd`
- `onnode` – execute programs on selected nodes

`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 ~]# █
```

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

CTDB manages ...

- `CTDB_MANAGES_SAMBA`
- `CTDB_MANAGES_WINBIND`
- `CTDB_MANAGES_NFS`
- `CTDB_MANAGES_VSFTPD`
- `CTDB_MANAGES_HTTPD`

3 Clustered Samba

Getting A Clustered Samba

- in vanilla Samba code since Samba 3.3 (January 2009)
- precompiled packages from <http://www.enterprisesamba.org/>
- configure `--with-cluster-support`
- add `idmap_tdb2` to `--with-shared-modules`
- verify that `gpfs.so` is built for GPFS usage

3.1 Configuration Options

Samba Configuration

identical configuration on all nodes

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

- if CTDB_MANAGES_SAMBA, do *not* set interfaces or bind interfaces only

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

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

3.2 Registry Configuration

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
- \Rightarrow 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
```

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.
```