

Clustered NAS For Everyone Clustering Samba With CTDB A Tutorial At sambaXP 2009

Michael Adam

`obnox@samba.org`

2009-04-21

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	9
3.1	Configuration Options	10
3.2	Registry Configuration	10

1 Cluster Challenges

1.1 The Ideas

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

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

- most problems are about distributing TDBs in the cluster
- TDB: small fast Berkeley-DB-style database with record locks and memory mapping
- persistent TDBs:
 - read frequently
 - written rather rarely
 - data consistency very important
- 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

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*
- A more clever 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 ...
- ... and messaging solution.

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

2 CTDB

2.1 The CTDB Project

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)

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
- management features Samba, NFS and other services

persistent TDBs

- each node always has complete copy in LTDB
- reads operations directly to LTDB
- write operations
 - lock entire DB in a transaction
 - perform R/W ops within the transaction
 - at commit distribute changes to other nodes and write to LTDB
- ⇒ data integrity and good read performance guaranteed

normal TDBs

- one node does not need to know all records all the time
- 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
- only one node has the current, authoritative copy of a record
- \Rightarrow *data master*
- R/W operation to a record:
 - check if data master
 - if not, request data master role and current copy of record over network
 - 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 Election / Recovery Lock

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

Performance Figures

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

32 client smbtorure NBENCH test

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

2.3 Clustered File Systems

Clustered File System - Requirements

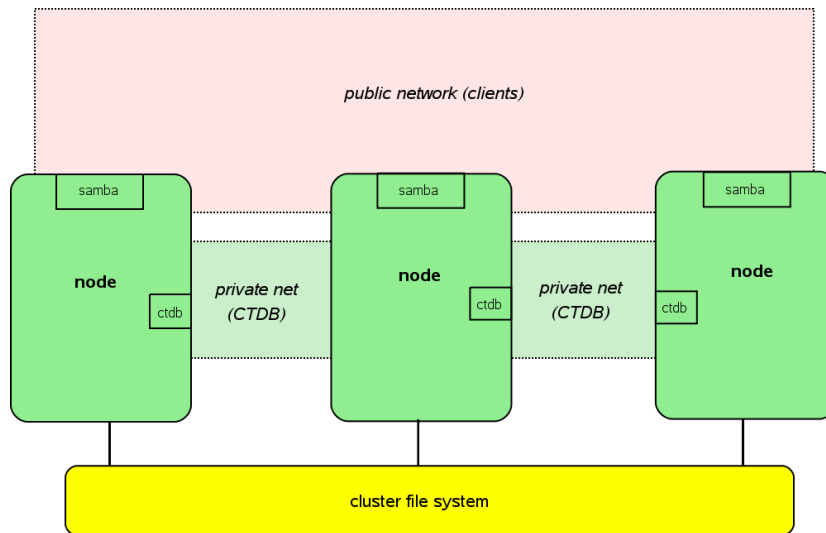
- file system : black box
- storage: fibre channel, iSCSI, drbd, ...
- simultaneous writes from all nodes
- coherent POSIX `fcntl` byte range lock support
- use `ping_pong` test to verify

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) : *not yet OK*

2.4 Setting Up CTDB

CTDB - Basic Setup



CTDB - Configuration

- central file: `/etc/sysconfig/ctdb`
- must set: `CTDB_RECOVERY_LOCK`
- fill `/etc/ctdb/notes` with internal addresses

example `/etc/ctdb/nodes`

```
10.0.0.10
10.0.0.11
10.0.0.12
10.0.0.13
```

- 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`

```
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...)

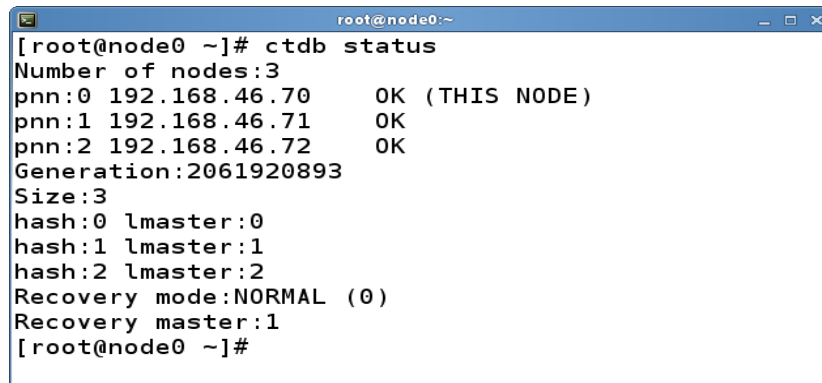
IP Failover

- 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*:
 - 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 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`
- `vfs objects = fileid`
`fileid:algorithm = fsid / fsname`
- `idmap backend = tdb2`
- 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

    fileid:algorithm = fsname

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

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
- 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.
```