Clustered Samba
Not just a hack any more

Andrew Tridgell &
Ronnie Sahlberg
Samba Team

At LCA last year ....

- We described 'CTDB', a new lightweight clustered database
  - We gave a hacked up demo
  - It sort of worked
- This year ....
  - It is now deployed in production at multiple locations
  - It is ready for YOU to use!

Start with a demo

- Demo cluster
    - Set of 4 Linux blades
    - GPFS filesystem, 12 raid arrays
- What we will demo
    - Fast IP failover
    - Snapshot exposure
    - Offline file handling
    - Online software upgrade
    - Crash resiliance

Scaling NAS

- What if?
    - you have 30,000 NAS users
    - you have 100 NAS servers
    - every day you run out of space on one of them
- What can you do?
    - Get a really big NAS box

What clustered Samba does ...

- Clustered Samba with CTDB provides
  - Highly available 'all-active' file serving
  - Very scalable performance
  - Support for snapshots
  - Support for offline files
  - Hooks for managing other cluster services

All-active NAS

- Active-passive?
  - the common solution for robust NAS in the past
  - a hot spare waits for a server to fail
  - on failure, STOMITH and take over role
  - admins pray that hot spare actually works
- All-active
  - All nodes in the cluster serve entire namespace all the time
  - when a node fails, all other nodes are already serving the same files
  - less reliance on divine intervention :-)

CTDB features

- Database
    - simple database API
    - automatic recovery on cluster changes
- IP failover
    - handles public IP assignment, gratuituous ARP
    - tickle-ACKs for fast failover
- Protocol hooks
    - CTDB offers 'event scripts' for protocol exensions
    - handles NFS lock recovery

Scaling Results

- smbtorture NBENCH test
  - 32 clients
  - 1 to 4 nodes

```
OLD (pre-CTDB) approach
1 node     95.0 Mbytes/sec
2 nodes     2.1 MBytes/sec
3 nodes     1.8 MBytes/sec
4 nodes     1.8 MBytes/sec

NEW (CTDB) approach
1 node      109 Mbytes/sec
2 nodes     210 MBytes/sec
3 nodes     278 MBytes/sec
4 nodes     308 MBytes/sec
```

- Streaming IO
  - We have seen 1.7 Gbyte/s sustained read for one share on one IP. Fastest CIFS server?

So you want to try it?

- What you need
  - A Linux cluster
  - Lots of fast disk (usually on a SAN)
  - A cluster filesystem (GPFS, GFS, GFS2 or Lustre)
  - ctdb and Samba from http://ctdb.samba.org/
- Getting help
  - Wiki and docs at ctdb.samba.org
  - #ctdb IRC channel on irc.freenode.net
- Supported version
  - IBM offers a supported, productised version called 'SOFS'
  - Maybe some other people would like to start supporting it?

Simple Clustered Samba Config

- Minimal Samba config:
  - clustering = yes
  - idmap backend = tdb2
- For ctdb
  - /etc/ctdb/nodes
  - /etc/ctdb/public_addresses
  - /etc/sysconfig/ctdb
- Filesystem specific options
  - fileid:mapping
- Winbindd options
  - idmap:backend = tdb2

Using CTDB

```
Usage: ctdb [options] <control>
Options:
   -n <node>          choose node number, or 'all' (defaults to local node)
   -Y                 generate machinereadable output
   -t <timelimit>     set timelimit for control in seconds (default 3)
Controls:
  status                                       show node status
  ping                                         ping all nodes
  getvar          <name>                       get a tunable variable
  setvar          <name> <value>               set a tunable variable
  listvars                                     list tunable variables
  statistics                                   show statistics
  statisticsreset                              reset statistics
  ip                                           show which public ip's that ctdb manages
  process-exists  <pid>                        check if a process exists on a node
  getdbmap                                     show the database map
  catdb           <dbname>                     dump a database
  getmonmode                                   show monitoring mode
  setmonmode      <0|1>                        set monitoring mode
  setdebug        <debuglevel>                 set debug level
  getdebug                                     get debug level
  attach          <dbname>                     attach to a database
  dumpmemory                                   dump memory map to logs
  getpid                                       get ctdbd process ID
  disable                                      disable a nodes public IP
  enable                                       enable a nodes public IP
  ban             <bantime|0>                  ban a node from the cluster
  unban                                        unban a node from the cluster
  shutdown                                     shutdown ctdbd
  recover                                      force recovery
  freeze                                       freeze all databases
  thaw                                         thaw all databases
  isnotrecmaster                               check if the local node is recmaster or not
  killtcp         <srcip:port> <dstip:port>    kill a tcp connection.
  gratiousarp     <ip> <interface>             send a gratious arp
  tickle          <srcip:port> <dstip:port>    send a tcp tickle ack
  gettickles      <ip>                         get the list of tickles registered for this ip
  regsrvid        <pnn> <type> <id>            register a server id
  unregsrvid      <pnn> <type> <id>            unregister a server id
  chksrvid        <pnn> <type> <id>            check if a server id exists
  getsrvids                                    get a list of all server ids
```

CTDB Tunables

- Lots of tunables
  - rarely need to be modified

```
[root@fscc-hs21-12 ~]# ctdb listvars
MaxRedirectCount    = 3
SeqnumFrequency     = 1
ControlTimeout      = 60
TraverseTimeout     = 20
KeepaliveInterval   = 2
KeepaliveLimit      = 5
MaxLACount          = 7
RecoverTimeout      = 5
RecoverInterval     = 1
ElectionTimeout     = 3
TakeoverTimeout     = 5
MonitorInterval     = 15
MonitorRetry        = 5
TickleUpdateInterval = 20
EventScriptTimeout  = 20
RecoveryGracePeriod = 60
RecoveryBanPeriod   = 300
DatabaseHashSize    = 10000
RerecoveryTimeout   = 10
EnableBans          = 1
DeterministicIPs    = 1
```

Status Monitoring

- 'ctdb status'
  - shows state of each node
  - most commonly used ctdb command

```
[root@fscc-hs21-12 ~]# ctdb status
Number of nodes:4
pnn:0 9.155.61.96      OK (THIS NODE)
pnn:1 9.155.61.97      OK
pnn:2 9.155.61.98      BANNED
pnn:3 9.155.61.99      OK
Generation:159484266
Size:4
hash:0 lmaster:0
hash:1 lmaster:1
hash:2 lmaster:2
hash:3 lmaster:3
Recovery mode:NORMAL (0)
Recovery master:1
```

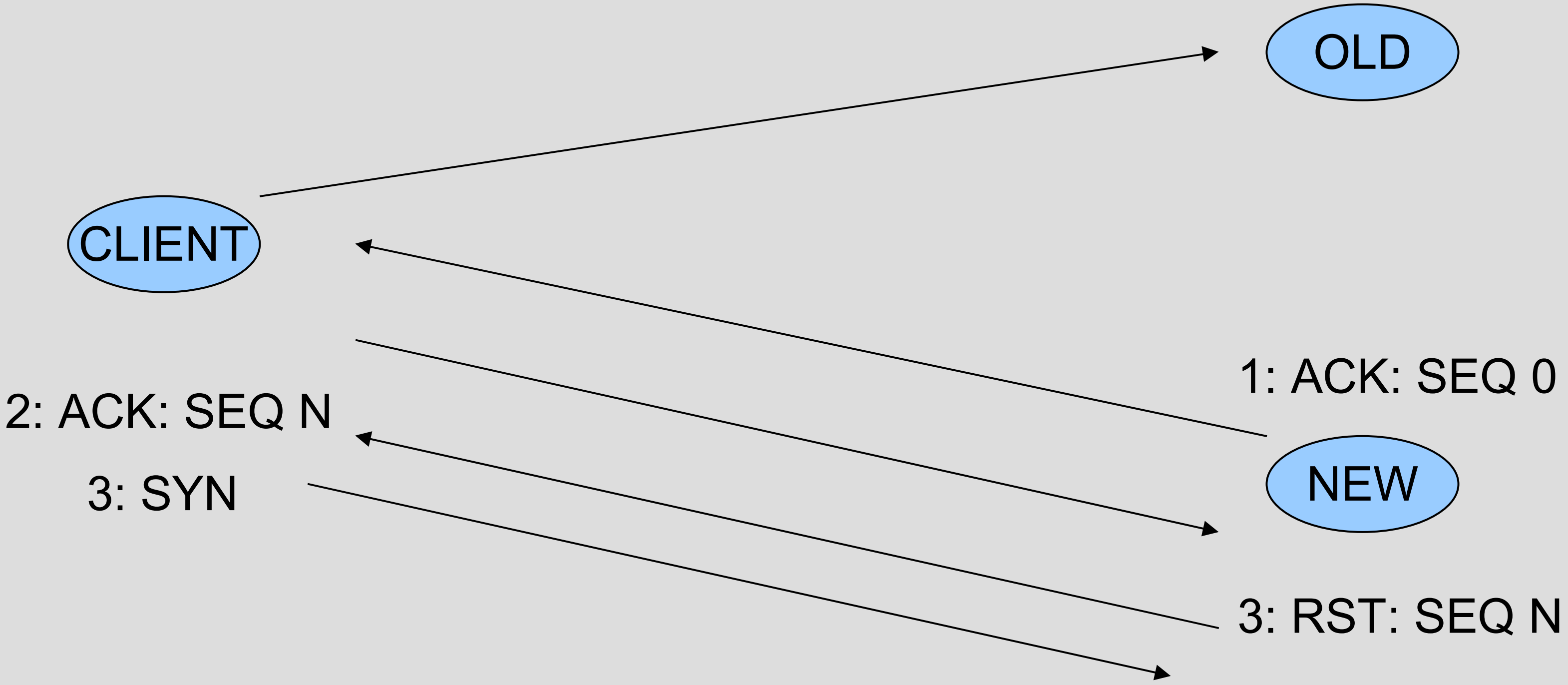
Public IPs

- IP Failover
  - 'HEALTHY' nodes get public IPs
  - these IPs are setup in rr-DNS
  - Alternatively, you can configure as a single IP for all nodes, using LVS

```
[root@fscc-hs21-12 ~]# ctdb ip
Public IPs on node 0
10.13.26.1 0
10.13.26.2 1
10.13.26.3 2
10.13.26.4 3
10.13.26.5 0
10.13.26.6 1
```

A nice TCP hack ....

- TCP tickle-ACK
  - new node constructs raw ACK, sequence 0
  - client sends ACK reply, correct sequence
  - new node sends RST
  - client re-establishes transport



Show your managers!

- Some flash movies available
  - http://samba.org/~tridge/ctdb_movies

Questions?

- For more information on CTDB see

  http://ctdb.samba.org/