Asterisk-Cluster based on Trixbox 1.2.3

Dr. Daniel Ali Aman – Kraehenbuehl

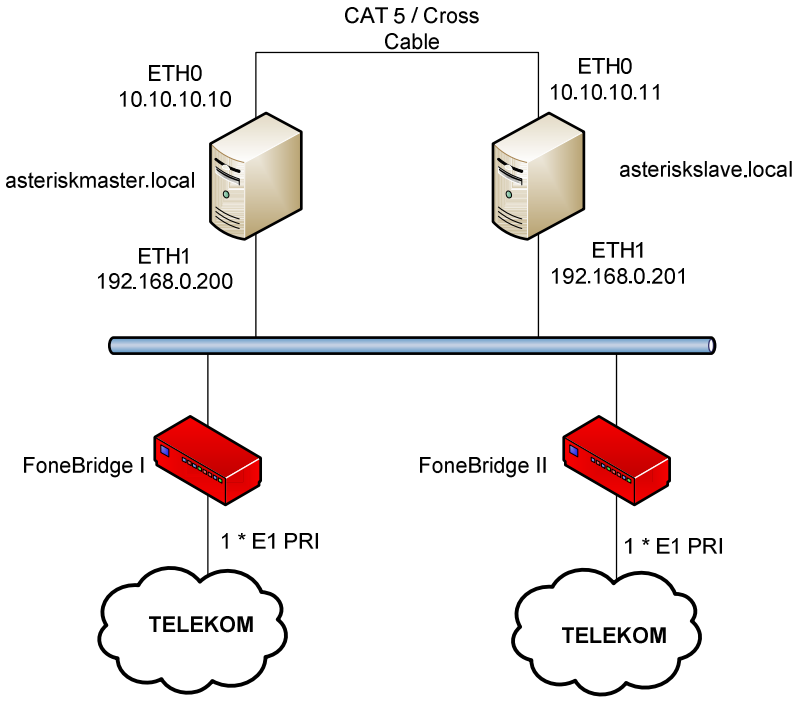
dkraehenbuehl@hotmail.com **(you can contact me anytime :-))**

Document Version: 1.0
Date: 30. May 2007

1. Scenario

This tutorial will help you to build a cluster with two Asterisk server, two Redfone FoneBridges with one PRI (E1) each. The idea behind this cluster is if one of the server's and one of the FoneBridge's fails the user is still able to conduct phone calls.

The fail-over will happen automatically within a couple of seconds. The Asterisk servers are in an active-passive cluster relationship.



For this tutorial we will name the two servers:

- Server 1 = asteriskmaster.local
- Server 2 = asteriskslave.local

Everything written on the Master or Slave server will be immediately replicated to the other machine. Therefore if you cerate a extension on the master the extension will be automatically "created" on the slave, and so on…

For this synchronization we will use DRBD, which will help us to build a network RAID 1.

To monitor our servers if there are up and running we will use Heartbeat, a Linux tool to build high availability systems.

In our scenario each of the Asterisk servers has two network cards (NIC's). The second network card (ETH1) on both servers is connected to a switch. This NIC's will handle the normal traffic between IP phones and the Asterisk server.

You can give them the following IP-Addresses:

- asteriskmaster.local: 192.168.0.200
- asteriskslave.local: 192.168.0.201

We will need a third IP address in the 192.168.0.201.xxx range this will be our floating or virtual IP-Address. Our floating IP-Address is:

- 192.168.0.201.203
- The currently active Asterisk server will automatically bind the floating IP-Address to it's ETH1. All IP-phones will point to this floating IP-Address, or in a real setup your DNS-Server need's to resolve to this IP. Let's say your phones pointing to talk.mycompany.com as register server, then talk.mycompany.com needs to resolve to 192.168.0.203

The two other NIC's (ETH0) on both Asterisk servers are connected via crossover cable. Please be aware that if you use 100Mbit NIC's you effective shared drive speed will not be more than 10 MBit/s. So if you like to synchronize a large RAID 1 it can initially take a long while to do that. I recommend that you use GBit NIC's for the DRBD synchronization.

Assumptions:

- You have installed the Trixbox 1.2.3 ISO CD on both machines.
- You configured the network interfaces accordingly

2. Re-Partitioning of current Asterisk installation.

Pitifully, Trixbox 1.2.3 doesn't allow user to setup partition during the installation process. The default installation will create the following partitions:

```
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14         947     7502355   83  Linux
/dev/sda3             948        1044      779152+  82  Linux swap
```

For our synchronization we need a additional dedicated partition, therefore we need to crate a new one and re-partition our current systems (AsteriskMaster and AsteriskSlave).

- Boot from the Asterisk installation CD (Trixbox 1.2.3)

- Enter linux rescue and press Enter. Choose the charset, answer "No" when asked about network and SKIP when asked about previous installations.

After that, you should get a shell prompt.

A little remark: commands may differ depending on type of HDD. If you have SATA or SCSI drive installed, then do as said. For IDE drive, substitute "hda"<->"sda" everywhere.

- Run e2fsck –f /dev/sda2 , to test the partition (you should not get any error). Don't proceed if you get errors, resolve the first!

- Now we resize the current sda2 partition. Run resize2fs –f –p /dev/sda2 3000M The "3000M" part here is saying that you want to resize file system to the size of 3Gb. You should change that, depending on your hard drive size. When deciding, you should remember:

    - You don't need swap bigger than 1Gb, and less than 512Mb.
    - If you create too big DRBD partition it will take lots of time for full synchronization between nodes. But if you create it too small, you may run out of disk space. Minimum size defined by DRBD is 4Gb. You may want to leave it so. Maybe twice as big. It's up to you to decide. I would recommend you anything between 4 and 16 Gb.

So, the math is easy. For example, you have a 8.5Gb hard drive. You decide to create 512Mb of swap, 4Gb DRBD and rest of the drive for main partition. So, main partition should be 4Gb. **But due to bug(or feature) in resize2fs(or fdisk?), multiply main partition size by 0.75** , and that will be our file system size. For our example, you need to resize fs to 3000M.

- When done, you should see something like that:

```
-/bin/sh-3.00# e2fsck -f /dev/sda2
e2fsck 1.35 (28-Feb-2004)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/: 54715/939136 files (0.6% non-contiguous), 306552/1875588 blocks
-/bin/sh-3.00# resize2fs -f -p /dev/sda2 3000M
resize2fs 1.35 (28-Feb-2004)
Resizing the filesystem on /dev/sda2 to 768000 (4k) blocks.
Begin pass 2 (max = 173923)
Relocating blocks             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Begin pass 3 (max = 58)
Scanning inode table          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Begin pass 4 (max = 6015)
Updating inode references     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXThe filesys
tem on /dev/sda2 is now 768000 blocks long.
```

- Now we run run fdisk /dev/sda. Press “p” and enter. You should see the following output. Pls. write down on which cylinder sda2 starts! In this case it’s cylinder **14**.

```
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14         947     7502355   83  Linux
/dev/sda3             948        1044      779152+  82  Linux swap
```

  - First we delete the swap partition.
    - Press “d”
    - Enter
    - Press “3”
    - Enter
  - Second we delete the sda2 partition
    - Press “d”
    - Enter
    - Press “2”
    - Enter
  - Now we create a new sda2 partition
    - Press “n”
    - Enter
    - Press “p”
  - Enter
  - Press “2”
  - Enter
  - Press “14” or the number you wrote down (sda2 cylinder start).
  - Enter
  - Press “+4000M” or whatever the size of your main sda2 partition should be
  - Enter
  - Press “w”
  - Enter

- Run fsck –f /dev/sda2 It should not report any error. If it reports error about wrong block count, run fdisk again, and recreate partition (a little bit bigger, remember there is a bug in resize2fs).

- Run resize2fs –f –p /dev/sda2 (without size, that will enlarge the file system size to maximum allowed on that partition).

- Again run fsck –f /dev/sda2 , to make sure everything went ok.

In this step we will create the swap and DRB partition.

- Run fdisk /dev/sda again. Now, we'll recreate swap partition and create one for DRBD.

    - Press "n"
    - Enter
    - Press "p"
    - Enter
    - Press "3"
    - Enter
    - Enter
    - Press "+768M" The size of the swap partition you want to create
    - Enter
    - Press "t'
    - Enter
    - Press "3"
    - Enter
    - Press "82"
    - Enter
    - Enter "n"
    - Enter
    - Enter "p"
    - Enter
    - Enter
    - Enter
    - Press "w"
    - Enter

- Run mkswap –L SWAP-sda3 /dev/sda3 Result should look like:

```
[root@asterisk1 ~]# mkswap -L SWAP-sda3 /dev/sda3
Setting up swapspace version 1, size = 263204 kB
[root@asterisk1 ~]# _
```

You can reboot your machines now, and boot back into "normal" mode.

3. Installation and configuration of DRBD.

The software package provided from my blog with this tutorial is preconfigured. You need to adjust some of the configuration files for your environment.

Download the ha.tar.gz2 file from my blog.

If you use a single CPU machine download this file
http://www.danielaliaman.com/blog/files/phonecube/cluster/onecpu/ha.tar.bz2

If you use a multi CPU machine download this file
http://www.danielaliaman.com/blog/files/phonecube/cluster/multicpu/ha.tar.bz2

Create a new directory in your root directory.

mkdir ha

Download, and Unpack the downloaded files into this new ha directory.
cd ha

wget http://www.danielaliaman.com/blog/files/phonecube/cluster/onecpu/ha.tar.bz2

tar –xvfj http://www.danielaliaman.com/blog/files/phonecube/cluster/onecpu/ha.tar.bz2

Attention: you must create the ha directory in the root directory of your servers, if not the installation scripts will not work!
The scripts will only work with a Trixbox 1.2.3 installation. If you install on a later Trixbox or plain Asterisk installation, you need to modify the scripts.

Before we can install these packages you need to do some “pre-flight” checks.

• ETH1 on each server should have personal IP addresses. It doesn’t really matter exactly what addresses, but so you can access servers from outer network.

• ETH0 should be preconfigured too. Primary server – 10.10.10.10, secondary server 10.10.10.11
If you want other addresses, substitute them in drbd.conf file.

• Both ethernet interfaces should start on boot.

• Hostname of Primary server is assumed to be asteriskmaster.local , on Secondary – asteriskslave.local . These should be returned by « uname -n » command. If you want to change them, you need to edit drbd.conf, ha.d/ha.cf, ha.d/haresources

Attention: You need a "." In your hostname. If you use just something like asteriskmaster sendmail will start very very slowly, therefore your failover will take minutes instead of seconds!

• «Floating» IP address is assumed to be 192.168.0.203 with netmask 255.255.255.0 and gateway 192.168.0.1. You want to change that, so edit ha.d/haresources .

• Make sure, that servers are accessible by each other. I.e. « ping 10.10.10.11 » from primary node, and « ping 10.10.10.10 » from secondary node. And there should be no firewall on ETH0 interface. It's not a security treat, because ETH0 network is closed.

• If you have Gbit NICs installed as ETH0, change drbd.conf « rate » parameter to «120M».

• If you have sata or scsi drive – change nothing, if your hdd is IDE – replace «sda» with «hda» in drbd.conf .(Currently, I've changed the drbd.conf , so the Primary servers hdd is /dev/sda, and secondary - /dev/hda).

• Run /root/ha/drbd script. On both servers. Script should not return any errors (If you don't do that simultaneously, DRBD may report that it's waiting for peer node to appear. If timeout passes or you type «yes» - it's ok. But after running drbd.primary you will need to make sure that nodes synchronize).

• You can check DRBD now by running cat /proc/drbd . Both nodes are in secondary/inconsistent state now. It's ok.

```
version: 0.7.17 (api:77/proto:74)
SVN Revision: 2093 build by buildcentos@build-i386, 2006-05-26 06:15:21
 0: cs:WFConnection st:Secondary/Secondary ld:Inconsistent
    ns:0 nr:0 dw:17632 dr:22133 al:7 bm:134 lo:0 pe:0 ua:0 ap:0
[root@asterisk2 ~]# _
```

• On the primary node run /root/ha/drbd.primary script now. It will take some time for it to finish. But no errors should be reported either. (In my case, it reports lots of warnings about "timestamp in future", but it's safe to ignore them).

• Wait. DRBD needs to synchronize disks, it may take hours depending on the size of DRBD partition and speed of the network(on Gbit NICs should be pretty fast). You can check the process state by executing cat /proc/drbd command.

```
[root@asterisk1 etc]# cat /proc/drbd
version: 0.7.17 (api:77/proto:74)
SVN Revision: 2093 build by buildcentos@build-i386, 2006-05-26 06:15:21
 0: cs:SyncSource st:Primary/Secondary ld:Consistent
    ns:442444 nr:0 dw:0 dr:442676 al:0 bm:239 lo:0 pe:50 ua:58 ap:0
        [==>.................] sync'ed: 12.9% (3041304/3483552)K
        finish: 0:12:04 speed: 4,168 (3,068) K/sec
[root@asterisk1 etc]# _
```

7) When synchronization is finished, you'll see the following on the primary server:

```
version: 0.7.17 (api:77/proto:74)
SVN Revision: 2093 build by buildcentos@build-i386, 2006-05-26 06:15:21
 0: cs:WFConnection st:Primary/Secondary ld:Consistent
    ns:0 nr:0 dw:19360 dr:22141 al:9 bm:136 lo:0 pe:0 ua:0 ap:0
[root@asterisk2 ~]# _
```

DRBD is up and working!

Let's install HeartBeat.

• Run /root/ha/ha script from the package. On both servers. It will stop following services: mysql, sendmail, asterisk, httpd, munin-node, ircd, xinetd, xplhub, vsftpd.

• After some time, heartbeat should bring services up on primary node. Check it by executing ps ax|grep asterisk command. It should give output like:

```
[root@asterisk2 ~]# ps ax|grep asterisk
 3428 ?        S      0:00 /bin/sh /usr/sbin/safe_asterisk
 3429 ?        Sl     0:02 /usr/sbin/asterisk -v -d -g -p -U asterisk -G asteris
k
 5678 tty1     S+     0:00 grep asterisk
[root@asterisk2 ~]# _
```

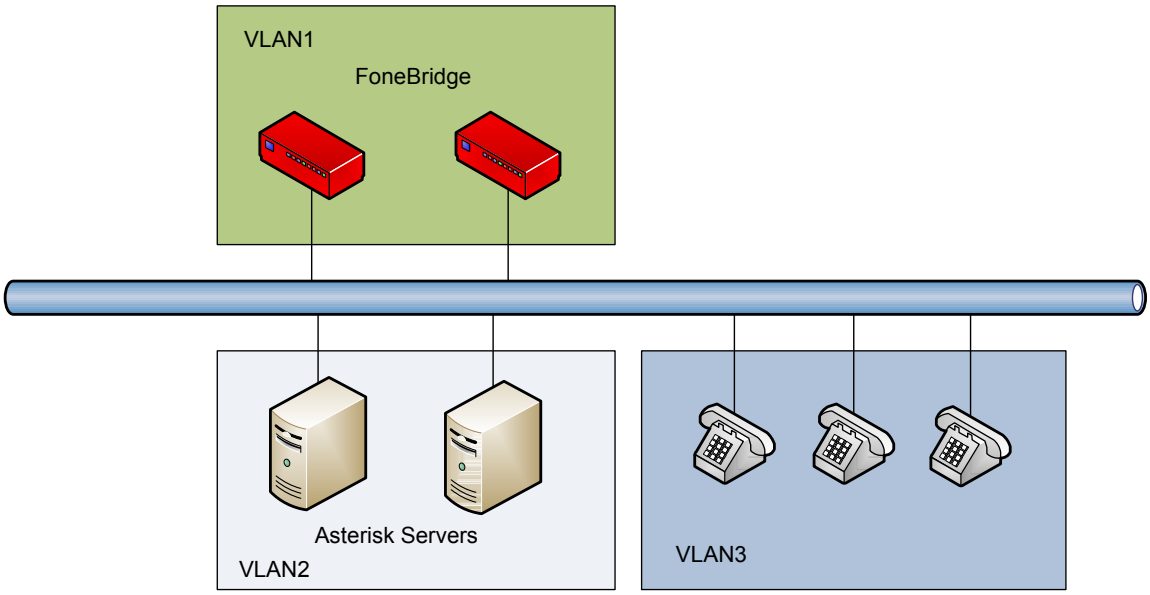
If you run a ifconfig eth1:0 you should see the floating IP-address 192.168.0.203 bounf to the eth1 of the primary server.

• Now, let's test the cluster! Just turn power off the Primary node. In a couple of seconds all the services and ethernet interface alias on Secondary node should be up. Now start Primary node. After it boots, all services on Secondary node should go down, and come up on Primary node.

The ha-cluster is ready for work.

3. Configuring the Redfone FoneBridge

Make sure that your FoneBridge's are isolated in a VLAN (at least for the production environment).



Rules: VLAN1 can communicate with VLAN2 and vice-versa
VLAN1 can not communicate / broadcast into VLAN3
VLAN2 and VLAN3 can communicate with each other

The main goal is to isolate the FoneBridge TDMoE-Traffic from the rest of the network.

3.1 Re-Install and compile Zaptel

To run TDMoE with the Redfone FoneBridge's we need to patch a small bug in the kernel-source code.

To that you need to install the kernel source first.

If you use a single CPU machine download this file:
http://www.danielaliaman.com/blog/files/phonecube/cluster/onecpu/kernel-devel-2.6.9-34.0.2.EL.i686.rpm

If you use a multi CPU machine download this file:
http://www.danielaliaman.com/blog/files/phonecube/cluster/multicpu/kernel-smp-devel-2.6.9-34.0.2.EL.i686.rpm
Copy this file on both servers into the /tmp directory.

Install the file with the RMP command:

rmp –ihv [filename]

In case of a single CPU machine it would be like

rpm –ihv kernel-devel-2.6.9-34.0.2.EL.i686.rpm

No we need to patch the kernel-source.

cd /usr/src/kernels/2.6.9-34.0.2.EL-i686/include/linux
(you need to adjust the path if you use multi cpu)

mv spinlock.h spinlock.h.orig

sed "s/rw_lock_t/rwlock_t/" < spinlock.h.orig > spinlock.h

Now we can re-install and re-compile the Zaptel driver.

Download the Zaptel-1.2.12 driver for FoneBridge from:

http://www.danielaliaman.com/blog/files/phonecube/cluster/zaptel-1.2.12.tar.gz

Copy this file into the tmp directory and untar it.

tar –xzvf zaptel-1.2.12

cd /usr/src/zaptel-1.2.12

make clean

make linux26

make install

modprobe Zaptel

Do this step on both machines, when done reboot the machines.

 # shutdown –r now

3.1 Configure the FoneBridge's and Zapata and Zaptel

To make the FoneBrigdes boxes work, you need to do a couple of installation steps.

In a first step you need to download, compile and install the Fonulator software. This software will basically send the configuration information (from the redfone.conf files) to the Redfone box. As soon the Redfone box receives it's configuration file it will start fonulating back to server.

The communication is based on TDMoE (Time Division Multiplexing over Ethernet) via MAC-addresses.

But first you download the fonulator software from here:

http://www.danielaliaman.com/blog/files/phonecube/cluster/fonulator-0.2.4d.tar.gz

Unpack the archive

#tar -xzvf fonulator-0.2.4d.tar.gz

Change to fonulator-0.2.4d directory

#cd fonulator-0.2.4d

Configure and build fonulator

```
#CFLAGS=-static ./configure
#make clean
#make
#make install
```

This last step will place the fonulator in the /usr/local/bin/ directory

Make sure that fonulator is executable

cd /usr/local/bin

chmod a+**x** fonulator

This installation needs to be done on both machines

Download the fonulator startup script

http://www.danielaliaman.com/blog/files/phonecube/cluster/fonulator

copy this file into /etc/init.d/ directory on both machines

Make sure the file can be executed

cd /etc/init.d

chmod a+**x** fonulator

Next you need to download the two redfone.conf files:

redfone1.conf
http://www.danielaliaman.com/blog/files/phonecube/cluster/redfone1.conf

redfone2.conf
http://www.danielaliaman.com/blog/files/phonecube/cluster/redfone2.conf

Copy this two files into /etc/ directory on both machines

You need to adjust this two files to fit your hardware.

redfone1.conf

```
[globals]
fb1=00:50:C2:65:D1:62
fb2=00:50:C2:65:D1:63
server1=00:1A:64:11:3B:76
card=eth1,fb1

[span1]
span=1,0,0,ccs,hdb3,crc4
server1
fb1
pri
```

Change fb1 and fb2 to the MAC address written on the back of your first FoneBridge.
Change server1 to the MAC address of your eth1 network card
Attention: the server1 parameter on the slave will have different MAC address, therefore redfone1.conf and redfone2.conf are different on the master and slave!

redfone2.conf

```
[globals]
```

```
fb1=00:50:C2:65:D1:6A
fb2=00:50:C2:65:D1:6B
server1=00:1A:64:11:3B:76
card=eth1,fb1

[span1]
span=1,0,0,ccs,hdb3,crc4
server1
fb1
pri
```

Change fb1 and fb2 to the MAC address written on the back of your second FoneBridge.
Change server1 to the MAC address of your eth1 network card
Attention: the server1 parameter on the slave will have different MAC address, therefore redfone1.conf and redfone2.conf are different on the master and slave!

The fonulator script in /etc/init.d will use this two config files to initialize the fonebridges.

Download the zaptel.conf from here:

http://www.danielaliaman.com/blog/files/phonecube/cluster/zaptel.conf

and copy it into /etc/ directory

This zaptel.conf is configured for two fonebridges with one E1 line each.

zaptel.conf

```
# 2 X E1 PRIs, on 2 separate fonebridges
#
# fonebridge No. 1
#
dynamic=eth,eth1/00:50:C2:65:D1:62/0,31,1
#
# fonebridge No. 2
#
dynamic=eth,eth1/00:50:C2:65:D1:6A/0,31,1
#
bchan=1-15,17-31
dchan=16
alaw=1-15,17-31
bchan=32-46,48-62
dchan=47
alaw=32-46,48-62
```

```
# Global data

loadzone  = us
defaultzone    = us
```

Please adjust the MAC addresses according to your FoneBridge MAC addresses. You only need to configure the MAC address of the FB1 interface of each FoneBridge.

Now you need to download the Zapata.conf file from:

http://www.danielaliaman.com/blog/files/phonecube/cluster/zapata.conf

Copy zapata.conf into the /etc/asterisk/ directory

You need to do that on both machines.

No you need to edit the haresources file on both machines.

vi /etc/ha.d/haresources

asteriskmaster.local 192.168.0.203/24/eth1 drbddisk::shared Filesystem::/dev/drbd0::/share::ext3 drbdlinks mysqld sendmail asterisk httpd munin-node vsftpd ircd xplhub amportal xinetd

change this file to

asteriskmaster.local 192.168.0.203/24/eth1 drbddisk::shared Filesystem::/dev/drbd0::/share::ext3 drbdlinks mysqld sendmail fonulator asterisk httpd munin-node vsftpd ircd xplhub amportal xinetd

This will start the fonulator script via the cluster software.

Power-Up the FoneBridge's and make sure they are connected to the switch.

The restart both machines.

If the servers are restarted login into the master server and run the following command:

zttool

You should see the two SPAN and the alarms should be cleared.