

## Contents

- [1 Introduction](#)
- [2 Configuration Requirements](#)
  - ◆ [2.1 Track Calls](#)
  - ◆ [2.2 Failover to a Second Server](#)
- [3 Instruct FreeSWITCH to recover calls](#)
- [4 HA configuration](#)
  - ◆ [4.1 Floating IP method on Debian Squeeze](#)
    - ◇ [4.1.1 Install arping](#)
    - ◇ [4.1.2 Allow FreeSWITCH to bind to nonlocal IP](#)
    - ◇ [4.1.3 Add an ethernet alias for the floating IP](#)
    - ◇ [4.1.4 Start primary FreeSWITCH server](#)
    - ◇ [4.1.5 Start secondary FreeSWITCH server](#)
    - ◇ [4.1.6 TEST!](#)
      - [4.1.6.1 Crash the primary server](#)
      - [4.1.6.2 Recover the calls on the secondary server](#)
  - ◆ [4.2 To Do](#)
- [5 Interesting links / snips](#)

## Introduction

At ClueCon MMX (2010) the FreeSWITCH team announced the call recovery feature that's now available in FreeSWITCH.

This feature means you can restart FreeSWITCH after a failure, and keep existing calls up with only a few seconds of audio dropped.

In the case of HA Clustered FreeSWITCH instances, you can automatically fail calls from box A to box B within a few seconds of a failure being detected. (typically 3-5 seconds)

## Configuration Requirements

### Track Calls

In the SIP profiles you must have track-calls enabled.

So if your call has 2 legs, you must enable track-calls on both profiles that the call uses.

## Freeswitch\_HA

This is done simply by using this config snip in your SIP profiles.

```
<param name="track-calls" value="true"/>
```

Be aware there is a minor performance hit using "track-calls" as call state (all session vars, similar to the mod\_xml\_cdr data) is stored in the core DB every time the channel state changes.

## Failover to a Second Server

If you want to recover calls a second server, then you must have some way of sharing the call recovery data.

ODBC or PostgreSQL are possible methods, and to do so, you must specify a DSN for all involved sip profiles (like you needed track-calls on all involved profiles).

```
<param name="odbc-dsn" value="odbc://dsn:username:password"/>
```

And also in switch.conf.xml like:

```
<param name="core-recovery-db-dsn" value="odbc://dsn:username:password"/>
```

If you want to clone everything, you will also need to add DSN information to:

```
voicemail.conf.xml (also you need to rsync/drbd/NAS share the actual voicemail files)
```

The hostname on both machines needs to be the same, as the sql query to recover calls selects by hostname. To do this, simply set the following parameter in switch.conf.xml on both FS instances to the same value:

```
<param name="switchname" value="freeswitch01"/>
```

## Instruct FreeSWITCH to recover calls

To instruct FreeSWITCH to recover from a failure all you need to do is execute:

```
sofia recover
```

To crash test the feature, you may execute:

```
fsctl crash
```

To have FreeSWITCH to automatically recover calls on startup:

```
<X-PRE-PROCESS cmd="set" data="api_on_startup=sofia recover"/>
```

## HA configuration

## Floating IP method on Debian Squeeze

I've tested this on debian squeeze, it works like a charm. :)

Please note that 10.10.10.11 should be a public facing IP, it is only used as an example here.

### Install arping

```
apt-get install arping
```

### Allow FreeSWITCH to bind to nonlocal IP

Add the following line to /etc/sysctl.conf

```
echo 'net.ipv4.ip_nonlocal_bind=1' >> /etc/sysctl.conf
```

Restart networking:

```
/etc/init.d/networking restart
```

Run:

```
sysctl -p
```

You should see:

```
net.ipv4.ip_nonlocal_bind = 1
```

If you don't, you did something wrong, if you do, continue...

### Add an ethernet alias for the floating IP

Add the following to /etc/network/interfaces on both primary and secondary FreeSWITCH servers:

(substituting 10.10.10.11 and 255.255.255.0 with your floating IP config)

```
iface eth0:0 inet static
address 10.10.10.11
netmask 255.255.255.0
```

### Start primary FreeSWITCH server

Bring up the floating IP

```
ifup eth0:0
```

Make sure that the SIP profiles are set to listen on your floating IP

## Freeswitch\_HA

```
<param name="rtp-ip" value="10.10.10.11"/>
<param name="sip-ip" value="10.10.10.11"/>
<param name="presence-hosts" value="10.10.10.11"/>
<param name="ext-rtp-ip" value="10.10.10.11"/>
<param name="ext-sip-ip" value="10.10.10.11"/>
```

### Start FreeSWITCH

```
/etc/init.d/freeswitch start
```

### Start secondary FreeSWITCH server

Make sure the floating IP is not loaded.

```
ifdown eth0:0
```

Make sure that the SIP profiles are set to listen on your floating IP

```
<param name="rtp-ip" value="10.10.10.11"/>
<param name="sip-ip" value="10.10.10.11"/>
<param name="presence-hosts" value="10.10.10.11"/>
<param name="ext-rtp-ip" value="10.10.10.11"/>
<param name="ext-sip-ip" value="10.10.10.11"/>
```

### Start FreeSWITCH

```
/etc/init.d/freeswitch start
```

## TEST!

Make a call and don't hang up.

### Crash the primary server

```
fs_cli -x 'fsctl crash'
```

### Bring down the floating IP

```
ifdown eth0:0
```

### Recover the calls on the secondary server

### Bring up the floating IP

```
ifup eth0:0
```

Run arping command to update the router

```
arping -q -c 3 -A -I eth0:0 10.10.10.11
```

### Start primary FreeSWITCH server

Recover the calls.

```
fs_cli -x 'sofia recover'
```

Voilà, the call should continue with only 3-5 seconds of audio dropped!!

## To Do

- TODO: Someone please fill this in as you play with this setup.
- TODO: Heartbeat example *Check the git-contrib in ledr/ha.d/ for some possibly good stuff.*
- TODO: UCarp config examples
- TODO: Pacemaker config example

See also: [Enterprise Deployment Freeswitch HA Keepalived](#)

## Interesting links / snips

It is possible to bind FreeSWITCH to a floating IP that is not yet assigned to the box. This means FreeSWITCH can be in a ready & waiting state when the IP moves over, making the switchover much faster.

Normally, the OS will refuse to allow FreeSWITCH to listen on an IP that is not assigned. Sofia will get an error and the profile will not start.

On Linux you work around this using:

```
sysctl to set net.ipv4.ip_nonlocal_bind = 1  
or  
echo 1 > /proc/sys/net/ipv4/ip_nonlocal_bind  
ubuntu: sysctl net.ipv4.ip_nonlocal_bind=1
```

These can be set on boot, either from a rc.d script or a /etc/sysctl.d/\* file.

This means the OS will allow FreeSWITCH to use that IP even though it isn't yet assigned to the machine. It won't receive traffic or respond to ARP requests until the machine takes control of that IP, but will FS to be already started so that it'll start handling calls as soon as it fails over.

- Interesting discussion on the mailing list:  
<http://lists.freeswitch.org/pipermail/freeswitch-users/2010-July/059943.html>
- A lot of the process used here can be applied to FS HA:  
[http://wiki.freeswitch.org/wiki/Enterprise\\_deployment\\_OpenSIPS#Heartbeat](http://wiki.freeswitch.org/wiki/Enterprise_deployment_OpenSIPS#Heartbeat)
- The FreeSWITCH high availability feature is demonstrated by Anthony Minessale in this presentation starting from 36:40: <http://www.viddler.com/explore/cluecon/videos/66/>