Enterprise deployment UltraMonkey

About

This method provides load balancing, and scalability. It should work across many protocols such as SIP, IAX, Woomera, H.323, and others. You may use this method on your FreeSWITCH™ servers regardless of operating system or platform. The Director Servers would however need to be Linux.

Your users would only need to know about one IP address (probably via DNS) and would not have to have any special setup on their end. It is likely they would never know that you have more than one machine performing the switching and routing regardless of how many servers are installed.

Scope

PRI fail-over via SS7, Media Proxies for maintaining RTP streams if a failure occurs and network redundancy are beyond the scope of this document. While these methods may optionally be added, and it is recommended for a more redundant network to add them, they will not be discussed here.

You are responsible for implementing additional measures as your topology and requirements dictate.

Ultra Monkey Design

In an Ultra Monkey environment there are two types of servers employed. We will look at each in turn later on.

Ultra Monkey utilizes 3 main components to accomplish its tasks:

- The Linux Virtual Server (LVS) for fast load balancing
- Linux HA framework is used to monitor the directors
- Idirectord which monitors the real-servers.

By utilizing these technologies, Ultra Monkey can use tested software and combine them in a way that provides a robust network.

Caveats

- This configuration assumes that you are doing only bridged calls, effectively using the real servers as media gateways and not as application servers. While some applications may not matter such as an IVR system where each real server has access to the same sound files and configuration data, others like conferencing becomes more tricky, although not impossible.

- IPVS treats UDP packets from the same source IP+port as the same connection, so will send them to the same destination. If you have multiple calls coming from the same server (e.g. providing a SIP trunking service to another PBX/softswitch) they will not be load balanced.

  - On 1st October 2010 a IPVS patch was announced on the netfilter mailing lists which added support for load balancing UDP SIP based on Call-ID through a new persistence engine framework. This will allow load balancing in the above scenario by individual call. This will be probably be available in a future version of the linux kernel.

Warranty

No warranties are expressed or implied this document is provided for informational and educational purposes only. If you use this method and your boxes crash and it doesn’t fail-over, or something else bad happens, neither the author(s) of this page nor the FreeSWITCH™ community are liable. You should thoroughly test your environment for the level of fault tolerance you desire and ensure that it does work the way you want it to, but ultimately you are responsible for implementation of your servers and network.

Total call volume

The total call volume through this system depends on various factors, not limited to network bandwidth, network topology, cpu power, and specific applications being served. It largely boils down to money though. The more money you put into this configuration, in terms of bandwidth and servers, the more concurrent calls you can process. It is not unreasonable to say that 100,000 concurrent calls can be achieved, potentially even 1,000,000 concurrent calls can be processed, although that has yet to be tested since it would require roughly $600,000 worth of servers. As a volunteer project and FreeSWITCH™ planning on becoming a 501(c)(3) entity, we do not have the resources to test a million concurrent call network.

In general if you are going to be doing transcoding, running heavy applications such as TTS or ASR you will require more servers. This will not scale in a perfect linear fashion either, however it will generally be linear in real world scenarios.

Network Map
This network diagram uses RFC1918 addresses, this does not mean that it will work in this configuration. They are used this way only for display purposes, and references later on in the configuration files will make use of them. If you want this to be internet accessible you should use your own routable IPs in their place.

You are not required to have 3 real servers, you may have no less than one but as many as you can reasonably fit into your network. 1 server by itself is a little silly since you cannot load balance or have any redundancy, 2 real servers would be the minimum. Keep in mind that as you add more servers you need to ensure that your network can handle that load.

This document does not discuss having only one director server, since high availability is part of the design. Having more than two is also not covered as that requires tweaks that would be outside the scope of this document.

**Director Servers**

The directors are the servers that route traffic selectively to each of your FreeSWITCH™ servers. These machines will process only the traffic from the end user into your network, where the return from the real server goes direct. In this way the highest level of network performance can be obtained since the director boxes do not need to process the response packets. This also means that for a given amount of load the director boxes do not have to be super fast, since they arent routing as much traffic.

Depending on network load you may be able to use a higher end Pentium III or low end Pentium 4 box. This reduces overall cost, while still providing a level of service that is as close to native as possible. You will not need a lot of memory, as the process of mapping connections takes only 16 bytes of ram per connection. You are encouraged to test the amount of load your setup can handle however and make provisioning decisions on hardware based on that.

**Real Servers**

The real servers in this example would be running FreeSWITCH™ and processing the calls. They may run on almost any FreeSWITCH™ supported operating system including at least Linux, Windows 2003, *BSD, etc. Please note that certain operating systems such as Windows CE are not suitable for this method due to some of the networking configurations that are required to avoid extra load on the directors. This, however should not be an issue as embedded systems running Windows CE are not likely to require a load balanced highly available network setup.

**Related Notes**

Asterisk behind UltraMonkey