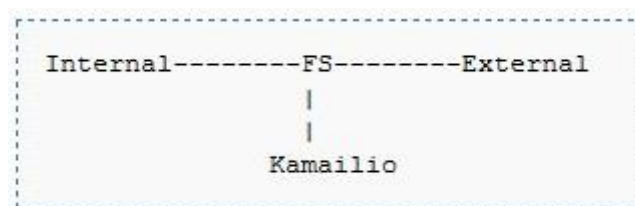


Abaixo você encontrará um passo a passo da configuração para instalar FreeSWITCH (FS) como SBC. O LCR o motor é fornecida pelo Kamailio e seu módulo `carrierroute`. Open Source Kamailio é um SIP Proxy (não um B2BUA). Com esta configuração, o plano de marcação (Dial Plan) é detalhado apenas para chamadas recebidas para tráfego de saída, mas ela pode ser facilmente estendida para saída de tráfego de entrada (ou NÃO). O projeto é o seguinte: FS é configurado com uma política interna e um perfil externo, cada perfil ouvindo uma interface de rede diferente. Kamailio está escutando a interface de `loopback`, e não é usado por nenhum outro processo do FS.



Quando FS recebe um `INVITE`, FS dispara este evento no contexto padrão (default context), onde ela sempre corresponde à extensão chamada `LOOKUP_ROUTE`. Nesta extensão, FS envia o `INVITE` para Kamailio, que responde com um redirecionamento "`302 Redirect SIP`" com a mensagem que contém a rota que o FS tem que usar para alcançar o número marcado.

Em seguida, a chamada é transferida no contexto `ROUTING` onde o FS vai coincidir com a associação do Gateway (nesta configuração é um código associado) que vai rotear a chamada.

Se Kamailio não tem rota para o número marcado, ele responde com um "**604 unable to route the call**" que FS vai retransmitir.

O Kamailio é usado apenas para pesquisa de rota pelo FS e o FS é o único equipamento que é visto na transação SIP (SIP Transaction).

Este projeto foi escolhido porque eu precisava de uma maneira de armazenar grandes tabelas de LCR (agora é mais de 450 000 reais as rotas no servidor de produção) e o módulo LCR não estava disponível quando comecei a trabalhar com FS (versão 1.0.1).

Eu tenho extensivamente testado esta configuração em um Intel Quad Core server (Q9550 funcionando com uma CPU de 2,83Ghz, com 8GB de RAM e um HD de 250 GB. Eu acredito que 4GB de RAM deve de longe ser suficientemente).

Realizei alguns testes utilizando SIPP com 750 chamadas simultâneas e entre 50 a 100 chamadas por segundo (CPS) por mais de 10 dias com 25% - 30% de CPU livre. FS neste teste foi a versão 1.2.

Vantagens

Estável com bastante grande roteamento de tabelas: Kamailio usou menos de 3% da CPU quando estava fazendo os testes de SIPP com 160 000 entradas no LCR.

Escalável: algumas pessoas estão utilizando módulo **carrieroute** com tabela LCR com até 1 milhão entradas. Conforme chamada por segundo a taxa: É de até 100 cps.

Desvantagens

Cada **INVITE** recebido vai ser enviado do FS para o Kamailio, isto com um novo **INVITE**, e uma vez que o FS recebe a resposta do Kamailio, FS vai fazer a ponte entre a chamada de entrada para o **peer** que vai processar a chamada. Estou querendo lhe dizer que, se você quiser processar mais chamada por segundos (mais de 100 CPS), o fato de a biblioteca **Sofia** ser **monothreaded** e ter que lidar com mais uma sessão (**FS** para o **Kamailio**) antes de processar a chamada para o **peer** faz com que o FS possa alcançar uma maior CPS. (Para toda as pessoas que tenha conhecimento interno de FS melhor que eu, sintam-se à vontade para corrigir essa hipótese).

Outras maneiras de alcançar o mesmo desempenho (não testado)

Tente o **mod_lcr** que foi introduzido no FS 1.0.3, tem relatos sobre isto na lista e na Wiki do projeto. Use o módulo **XML CURL** para processar o roteamento de chamadas usando sua linguagem de script preferida para trabalhar no seu servidor HTTP.

Preliminares

Instalar **Debian estável netinstall nofree** funcionando plenamente (se você for usar a versão 64 bits, tome cuidado se o seu processador é 64 bits).

Upgrade da distribuição Debian

```
# apt-get update
# apt-get dist-upgrade
```

Instalar Servidor SSH para administração remota

```
# apt-get install openssh-server
```

Instale pacote Vlan

```
# apt-get install vlan
```

Instalar pacotes Bridge (Ponte)

```
# apt-get install bridge-utils
```

Instalar pacotes Captura de pacotes

```
# apt-get install ngrep
# apt-get install tshark
# apt-get install tcpdump
```

Instalar editor de texto

```
# apt-get install emacs22-nox
# apt-get install vim
```

Instalar o Screen (para compartilhamento de tela)

```
# apt-get install screen
```

Desative serviços inúteis

```
# /etc/init.d/exim4 stop
# update-rc.d -f exim4 remove

# /etc/init.d/portmap stop
# update-rc.d -f portmap remove

# /etc/init.d/nfs-common stop
# update-rc.d -f nfs-common remove
```

Reinicie o servidor

```
# reboot
```

Verifique se os serviços TCP ou UDP estão sendo executados

```
# netstat -tnlp
```

Apenas o SSH (porta 22) deve estar sendo escutado

```
# netstat -unlp
```

Apenas o dhclient3 (porta 68) deve estar sendo escutado (eu preciso dhcp para meus testes e para obter um IP para a administração do servidor de teste)

Instalar utilitários para compilação

```
# apt-get install build-essential
```

Instalando o FreeSWITCH

Instalando as dependências

```
# apt-get install subversion subversion-tools automake1.9 gcc-4.1  
autoconf make wget libtool g++ libncurses5 libncurses5-dev
```

```
# apt-get install debhelper automake1.9 autoconf libtool unixodbc-  
dev libasound2-dev libcurl4-openssl-dev libssl-dev libogg-dev  
libvorbis-dev libperl-dev libgdbm-dev libdb-dev libgnutls-dev  
libspandsp-dev libtiff4-dev
```

Baixe o FreeSWITCH por GIT para pegar a versão mais recente, para a pasta **/opt/sources/**

```
# cd /opt/sources/  
# git clone git://git.freeswitch.org/freeswitch.git freeswitch-git  
# cd freeswitch-git  
# ./bootstrap.sh
```

Edite o arquivo **modules.conf** de acordo com as suas necessidades.

```
# ./configure -prefix=/opt/freeswitch-git --enable-core-odbc-support  
# make  
# make install  
# make sounds-install  
# make moh-install
```

Para atualizar para a última versão via Git

```
# cd /opt/sources/freeswitch-git  
# make current
```

É claro que tem que obter a última versão via GIT para a pasta **/opt/sources/freeswitch-git**

Preparar interface de rede (estou usando um switch com vlans, útil para isolamento de rede)

```
* Load linux vlan module
```

```
# modprobe 8021q
```

```
* Adicionar 2 vlans (um para perfil interno ex: vlan 100, outro para perfil externo ex: vlan 200).
```

```
# vconfig add eth0 100
```

```
# vconfig add eth0 200
```

```
* Ativar estas interfaces
```

```
# ip link set up dev eth0.100
```

```
# ip link set up dev eth0.200
```

```
* Ativar IP valido para estas interfaces
```

```
# ip addr add 10.10.10.254/24 dev eth0.100
```

```
# ip addr add 10.10.20.254/24 dev eth0.200
```

Editar **sip_profiles** para usar esses IPs

```
# cd /opt/freeswitch/conf/sip_profiles
```

```
* Edite o arquivo internal.xml e substitua ${local_IP_v4} pelo IP interno (10.10.10.254)
```

```
* Edite o arquivo external.xml e substitua ${local_IP_v4} pelo IP externo (10.10.20.254)
```

Otimizações antes de lançar FS

```
ulimit -c unlimited
ulimit -d unlimited
ulimit -f unlimited
ulimit -i unlimited
ulimit -n 999999
ulimit -q unlimited
ulimit -u unlimited
ulimit -v unlimited
ulimit -x unlimited
ulimit -s 240
ulimit -l unlimited
ulimit -a
```

Iniciando FreeSWITCH (FS)

```
# /opt/freeswitch/bin/freeswitch -nc -hp
```

```
-nc representa sem console
```

```
-hp representa prioridade alta (high priority)
```

Conectando o FreeSWITCH (FS) via `fs_cli`

```
/opt/freeswitch/bin/fs_cli
```

* Com esta ferramenta, você será capaz de obter uma conexão de console para FS no nível de log, não foi alterado para que pudesse ter detalhes no log. Para desativar faça login, digite `/nolog` na console.

Verifique se FS está funcionando

```
# netstat -unlp
```

* Você deve ter esta resposta

```
udp  0  0  10.10.10.254:5060  0.0.0.0:*  9240/freeswitch
udp  0  0  10.10.20.254:5060  0.0.0.0:*  9240/freeswitch
```

Instalando o Kamailio

Instalando o Bison, um gerador parser

```
# apt-get install bison
```

Instalando o Flex, um analisador muito rápido para gerador léxico

```
# apt-get install flex
```

Instalando o MySQL e as livrarias para desenvolvimento com MySQL

```
# apt-get install mysql-server-5.0
```

Você vai ser solicitado a definir a senha de seu banco de dados, vamos usar a senha "mypass" ou o que você quiser.

```
# apt-get install libmysql++-dev
```

Instalando o libconfuse, Necessários para compilação do módulo `carrieroute`

```
# apt-get install libconfuse-dev
```

Obtenha as fontes do Kamailio e coloque-as na pasta `/opt/sources`

```
# cd /opt/sources
```

```
# wget http://www.kamailio.org/pub/kamailio/1.5.0/src/kamailio-1.5.0-notls\_src.tar.gz
```

```
# tar -xvzf kamailio-1.5.0-notls_src.tar.gz
```

```
# cd kamailio-1.5.0-notls
```

Se você precisa de uma tabela LCR de rota pronta, você vai encontrar no mercado paralelo por aproximadamente 50.000, e caso tenha este capital, ignore este passo. Para quem quer carregar uma tabela LCR de rota em `carrieroute` (para as minhas necessidades: 160.000 rotas) siga os passos a seguir. Ainda não encontrei uma tabela LCR de rotas

para o Brasil, por isto estou usando da américa latina (160.000 rotas)

```
# emacs /opt/sources/kamailio-1.5.0-notls/config.h
```

```
#define PKG_MEM_POOL_SIZE 32*1024*1024 /*!< Used only if PKG_MALLOC is defined*/
#define SHM_MEM_SIZE 4*32 /*!< Used if SH_MEM is defined*/
```

Nota: Estes valores foram fixados por meios empíricos, são talvez demasiado elevados, mesmo para 160.000 rotas.

Compile o Kamailio em /opt/sources/kamailio-x.x.x

```
# make prefix=/opt/kamailio-x.x.x
# make prefix=/opt/kamailio-x.x.x modules=modules/db_mysql modules
# make prefix=/opt/kamailio-x.x.x modules=modules/carrierroute
modules
# make prefix=/opt/kamailio-x.x.x install
# make prefix=/opt/kamailio-x.x.x modules=modules/carrierroute
install
# make prefix=/opt/kamailio-x.x.x modules=modules/db_mysql install
```

Adicione um link simbólico para /opt/sources/kamailio-x.x.x

```
# ln -sf /opt/sources/kamailio-x.x.x/ /opt/kamailio
```

Edite as configurações do DB no Kamailio

```
# cd /opt/kamailio/etc/kamailio/
# emacs kamctlrc
```

```
# this parameter...
DBENGINE=MYSQL
## database host
DBHOST=localhost
## database name (for ORACLE this is TNS name)
DBNAME=openser
# database path used by dbtext or db_berkeley
# DB_PATH="/usr/local/etc/kamailio/dbtext"
## database read/write user
DBRWUSER=openser
## password for database read/write user
DBRWPW="1234"
## database read only user
DBROUSER=openserro
## password for database read only user
DBROPW="1234"
## database super user (for ORACLE this is 'scheme-creator' user)
DBROOTUSER="root"
# user name column
USERCOL="username"
```


Em seguida, execute o Kamailio utils para criar DB

```
# /opt/kamailio/sbin/kamdbctl create
```

Você será solicitado para digitar a senha de root do mysql (lembre-se que usamos mypass ou ...) e não se esqueça de definir **yes** (sim) para criação da tabela **carrierroute**

Preencher o DB do **carrierroute** com algumas rotas

Conecte ao MySQL

```
# mysql -u root -p
```

Entre com sua senha

No MySQL console:

```
use opener;
INSERT INTO carrier_name (carrier) VALUES ("default");
INSERT INTO domain_name (domain) VALUES ("default");
INSERT INTO carrierroute
(carrier, domain, scan_prefix, flags, mask, prob, strip, rewrite_host, rewrite_prefix, rewrite_suffix, description) VALUES ("1", "1", "1000", "0", "0", "1", "0", "PEER-01", "", "", "BRASIL");
exit
```

Aqui neste exemplo você inseriu o seu primeiro percurso, que vai combinar com prefixo 1000 e encaminhar este para PEER_01 com uma descrição, A BRASIL. PEER_01 será definida posteriormente no plano de marcação (dial plan) de FS. O parâmetro de descrição poderia ser exportado, para fins de faturação por exemplo. Nesta configuração, eu vou exportar a designação de uma área variável para o FS.

Configurando o Kamailio

```
# cd /opt/kamailio/etc/kamailio
# mv kamailio.cfg kamailio.cfg.default
# emacs kamailio.cfg
```

E coloque a configuração abaixo:

```
# KAMAILIO basic configuration script for use with carrierroute and FS

##### Global Parameters #####

debug=1
log_stderr=no
log_facility=LOG_LOCAL0

fork=yes
children=8

/* uncomment the next line to disable TCP (default on) */
disable_tcp=yes

/* uncomment the next line to disable the auto discovery of local aliases
based on revers DNS on IPs (default on) */
```



```
auto_aliases=no

/* uncomment and configure the following line if you want opensips to
   bind on a specific interface/port/proto (default bind on all available) */
listen=udp:127.0.0.1:5062

##### Modules Section #####

#set module path
mpath="/opt/kamailio/lib64/kamailio/modules/"

/* uncomment next line for MySQL DB support */
loadmodule "db_mysql.so"
loadmodule "sl.so"
loadmodule "tm.so"
loadmodule "rr.so"
loadmodule "maxfwd.so"
loadmodule "textops.so"
loadmodule "mi_fifo.so"
loadmodule "xlog.so"
loadmodule "carrierroute.so"
loadmodule "pv.so"
loadmodule "avpops.so"

#----- setting module-specific parameters -----

# ----- mi_fifo params -----
modparam("mi_fifo", "fifo_name", "/tmp/kamailio_fifo")

# ----- Carrierroute -----
modparam("carrierroute", "config_source", "db")
modparam("carrierroute", "db_url", "mysql://openser:1234@localhost/openser")
modparam("carrierroute", "carrierroute_table", "carrierroute")
modparam("carrierroute", "default_tree", "default")
modparam("carrierroute", "fetch_rows", 2000)

##### Routing Logic #####

# main request routing logic

route{

    if (!mf_process_maxfwd_header("10")) {
        sl_send_reply("483","Too Many Hops");
        exit;
    }

    t_check_trans();

    if ($rU==NULL) {
        # request with no Username in RURI
        sl_send_reply("484","Address Incomplete");
        exit;
    }

    # LOOKUP ROUTE TABLE WHEN ASKED BY HEADER: X-ROUTE:LOOKUP
    if (is_method("INVITE") && $hdr(X-ROUTE)=="LOOKUP"){
```

```

        if(!cr_route("default", "default", "$rU", "$rU",
"call_id", "$avp(s:route_desc)")){
            #xlog("ROUTING FAILED: no route found for $rU");
            sl_send_reply("604", "Unable to route this call");
            exit;
        } else {
            #xlog("LOOKUP FOUND: $rd $avp(s:route_desc)");
            avp_push("ru/username", "$avp(s:route_desc)");
            sl_send_reply("302", "$rd");
            exit;
        }
    }
}

```

Iniciando o Kamailio

```
# /opt/kamailio/sbin/kamctl start
```

Verificando se o Kamailio está rodando

```
# netstat -unlp
```

Você deve ver algo como isto

```
udp  0  0  127.0.0.1:5062  0.0.0.0:*  16269/kamailio
```

Verifique a tabela de rota

Interrompa o Kamailio, e cheque o MySQL Table

```
# /opt/kamailio/sbin/kamctl cr show
```

Você deverá ver uma rota com PEER_01

Configurando o FreeSWITCH

Configurando o plano de marcação (dial plan) do FS

```
# cd /opt/freeswitch/conf/dialplan
```

```
# emacs default.xml
```

Coloque isto

```

<?xml version="1.0" encoding="utf-8"?>
<!-- http://wiki.freeswitch.org/wiki/Dialplan_XML -->
<include>
  <context name="default">

    <extension name="LOOKUP_ROUTE">
      <condition field="destination_number" expression="(\d+)$">
        <action application="set" data="hangup_after_bridge=true"/>
        <action application="set" data="continue_on_fail=true"/>
        <action application="export" data="sip_h_X-ROUTE=LOOKUP"/>
        <action application="bridge"
data="sofia/internal/${destination_number}@127.0.0.1:5062"/>

```

```
        <action application="set"
data="ROUTE_GW=${sip_redirect_contact_host_0}"/>
        <action application="set" data="AREA=${sip_redirect_contact_user_0}"/>
        <action application="transfer" data="${destination_number} XML ROUTING"/>
    </condition>
</extension>

</context>

<context name="ROUTING">

    <extension name="PEER_01">
        <condition field="${ROUTE_GW}" expression="PEER_01">
            <action application="set" data="hangup_after_bridge=true"/>
            <action application="set"
data="continue_on_fail=NORMAL_TEMPORARY_FAILURE,TIMEOUT,NO_ROUTE_DESTINATION"/>
            <action application="set" data="PEER=1.1.1.1"/>
            <action application="bridge"
data="sofia/external/${destination_number}@1.1.1.1"/>
            <action application="set" data="PEER=2.2.2.2"/>
            <action application="bridge"
data="sofia/external/${destination_number}@2.2.2.2"/>
            <action application="set" data="PEER=3.3.3.3"/>
            <action application="bridge"
data="sofia/external/${destination_number}@3.3.3.3"/>
        </condition>
    </extension>

</context>

</include>
```

De fato PEER_01 é utilizado como um índice para uma combinação de pares. Eu posso fazer isso para ter um mecanismo de failover para um backup peer se a 1.ª peer estiver down.

Testar a conectividade entre FreeSWITCH e o Kamailio

Testar se o FS e Kamailio estão falando entre sí

Registrar um telefone IP usando credenciais em `/opt/freeswitch/conf/directory/default/1001`, por exemplo. Para descarregar o tráfego na interface de loopback para mensagens SIP entre FS e Kamailio.

```
# ngrep -d lo -nn -W byline
```

Quando o telefone estiver registrado, disque 1000, você deve ver isso no rastreamento do ngrep.

```
U 10.10.10.254:5060 -> 127.0.0.1:5062
INVITE sip:1000@127.0.0.1:5062 SIP/2.0.
Via: SIP/2.0/UDP 10.10.10.254;rport;branch=z9hG4bKZjpQ3tQ8SN7gc.
Max-Forwards: 69.
From: "1001" <sip:1001@10.10.10.254>;tag=pZ4egZ0m43Q3c.
To: <sip:1000@127.0.0.1:5062>.
Call-ID: 7015d499-6c86-122c-80be-001b21009ccd.
CSeq: 110711449 INVITE.
Contact: <sip:mod_sofia@10.10.10.254:5060>.
User-Agent: FreeSWITCH-mod_sofia/1.0.trunk-11592M.
Allow: INVITE, ACK, BYE, CANCEL, OPTIONS, PRACK, MESSAGE, SUBSCRIBE, NOTIFY,
REFER, UPDATE, REGISTER, INFO.
Supported: timer, precondition, path, replaces.
Allow-Events: talk, refer.
Min-SE: 120.
Content-Type: application/sdp.
Content-Disposition: session.
Content-Length: 333.
X-ROUTE: LOOKUP.
Remote-Party-ID: "1001" <sip:1001@10.10.10.254>;screen=yes;privacy=off.
.
v=0.
o=FreeSWITCH 3223313286554167397 4405097820346134292 IN IP4 10.10.10.254.
s=FreeSWITCH.
c=IN IP4 10.10.10.254.
t=0 0.
m=audio 31686 RTP/AVP 9 0 8 3 101 13.
a=rtpmap:9 G722/8000.
a=rtpmap:0 PCMU/8000.
a=rtpmap:8 PCMA/8000.
a=rtpmap:3 GSM/8000.
a=rtpmap:101 telephone-event/8000.
a=fmtp:101 0-16.
a=rtpmap:13 CN/8000.
a=ptime:20.

#
U 127.0.0.1:5062 -> 10.10.10.254:5060
SIP/2.0 302 PEER_01.
Via: SIP/2.0/UDP 10.10.10.254;rport=5060;branch=z9hG4bKZjpQ3tQ8SN7gc.
From: "1001" <sip:1001@10.10.10.254>;tag=pZ4egZ0m43Q3c.
To: <sip:1000@127.0.0.1:5062>;tag=458fb4012080e656b6742c09466dabcd.1740.
Call-ID: 7015d499-6c86-122c-80be-001b21009ccd.
CSeq: 110711449 INVITE.
Contact: sip:1000@PEER_01.
Server: Kamailio (1.4.3-not1s (x86_64/linux)).
```

```
Content-Length: 0.  
.  
  
#  
U 10.10.10.254:5060 -> 127.0.0.1:5062  
ACK sip:1000@127.0.0.1:5062 SIP/2.0.  
Via: SIP/2.0/UDP 10.10.10.254;rport;branch=z9hG4bKZjpQ3tQ8SN7gc.  
Max-Forwards: 69.  
From: "1001" <sip:1001@10.10.10.254>;tag=pZ4egZ0m43Q3c.  
To: <sip:1000@127.0.0.1:5062>;tag=458fb4012080e656b6742c09466dabcd.1740.  
Call-ID: 7015d499-6c86-122c-80be-001b21009ccd.  
CSeq: 110711449 ACK.  
Content-Length: 0.  
.
```

Otimizações

Pare o FreeSWITCH

```
# /opt/freeswitch/bin/freeswitch -stop
```

Baixar o nível de LOG do FreeSWITCH

```
# emacs /opt/freeswitch/conf/autoload_configs/switch.conf.xml
```

```
<!-- Default Global Log Level - value is one of  
debug,info,notice,warning,err,crit,alert -->  
  <param name="loglevel" value="err"/>
```

Configuração do Controle de admissão de chamadas (CAC)

```
# emacs /opt/freeswitch/conf/autoload_configs/switch.conf.xml
```

```
<!--Most channels to allow at once -->  
<param name="max-sessions" value="1000"/>  
<!--Most channels to create per second -->  
<param name="sessions-per-second" value="100"/>
```

Com estas definições, estamos definido um limite máximo de 500 chamadas em ponte (2 * 500 canais) e um máximo de 100 novas sessões por segundo. Você deve definir estes parâmetros de acordo com o seu hardware.

Criação de um ramdisk para o database do FS

```
# mv /opt/freeswitch/db /opt/freeswitch/db_old  
# mkdir /opt/freeswitch/db  
# mount -t tmpfs tmpfs /opt/freeswitch/db
```

Use Proxy Media para manter o caminho da mídia do FS na RTP (ocultação de topologia), mas sem analisar o RTP ou o DTMF

```
# emacs /opt/freeswitch/conf/sip_profiles/internal.xml
```

```
<param name="inbound-late-negotiation" value="true"/>
```

Adicionar diretiva proxy media no plano de marcação (Dial Plan).

```
<extension name="PEER_01">
  <condition field="${sip_h_X-ROUTE}" expression="PEER_01">
    <action application="set" data="proxy_media=true"/>
    <action application="set" data="hangup_after_bridge=true"/>
    <action application="set"
data="continue_on_fail=NORMAL_TEMPORARY_FAILURE,TIMEOUT,NO_ROUTE_DESTINATION"/>
    <action application="set" data="PEER=1.1.1.1"/>
    <action application="bridge"
data="sofia/external/${destination_number}@1.1.1.1"/>
    <action application="set" data="PEER=2.2.2.2"/>
    <action application="bridge"
data="sofia/external/${destination_number}@2.2.2.2"/>
    <action application="set" data="PEER=3.3.3.3"/>
    <action application="bridge"
data="sofia/external/${destination_number}@3.3.3.3"/>
  </condition>
</extension>
```

Desabilite a presença de apoio em SIP_Profiles

```
# emacs /opt/freeswitch/conf/sip_profiles/internal.xml
```

```
<param name="manage-presence" value="false"/>
```

```
# emacs /opt/freeswitch/conf/sip_profiles/external.xml
```

```
<param name="manage-presence" value="false"/>
```

Assegurar-se de que as suas Interfaces de Rede (PLACA DE REDE) não estão com compartilhamento de IRQ

```
# cat /proc/interrupts
```

O ideal é que as placas ethernet sejam colocadas em IRQ diferentes, e se você tiver mais de uma CPU, você pode forçar afinidade de uma placa com uma CPU específica (vou adicionar uma nota mais tarde sobre isso). Faça um reboot no servidor.

Verificar o uso da CPU

Instalar algumas ferramentas uteis para verificar os consumos da CPU

```
# apt-get install sysstat
```

Você pode verificar a carga de suas CPUs usando

```
# mpstat -P ALL 1
```

```
13:13:13 CPU %user %nice %sys %iowait %irq %soft %steal %idle intr/s
13:13:14 all 0,00 0,00 0,00 0,00 0,00 0,00 0,00 100,00 162,00
13:13:14 0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 100,00 162,00
```

O %idle representa a porcentagem da CPU, quanto maior melhor, pois significa que você tem mais recursos de CPU livre.

```
# vmstat 1
```

```
procs -----memory----- --swap-- --io-- --system-- --cpu---
r b swpd free buff cache si so bi bo in cs us sy id wa
1 0 80 234312 93496 1273928 0 0 1 32 55 65 2 3 95 0
0 0 80 234296 93496 1273928 0 0 0 0 106 1299 0 0 100 0
```

As últimas 2 colunas são interessantes:

Id: idle, mesmo que antes de mpstat, quanto maior melhor.

Wa: em espera E/S (I/O), quando esse número é alto, significa que a sua CPU está à espera para E/S (ex: HDD com muito acesso) e terá um impacto negativo sobre o desempenho do sistema.

```
# top
```

Eu acredito que eu não tenho que informar nada sobre esta ferramenta.

Links importante sobre Kamailio e carrieroute

- <http://www.kamailio.org/>
- <http://www.kamailio.org/docs/modules/1.4.x/carrieroute.html>

Referencia

- <http://www.it-ebooks.info/book/2405/>
- <http://www.it-ebooks.info/book/2905/>