

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**Implementando uma Solução VoIP Baseada em
Asterisk com Alta-disponibilidade e
Balanceamento de Carga**

FÁBIO DANIELESKI GROSS

FLORIANÓPOLIS – SC

2009 / 1

Fábio Danieleski Gross

**Implementando uma Solução VoIP Baseada em Asterisk
com Alta-Disponibilidade e Balanceamento de Carga**

Trabalho de conclusão de curso submetido à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Roberto Willrich, Dr.
Universidade Federal de Santa Catarina
willrich@inf.ufsc.br

Banca examinadora

Prof. Vitório Bruno Mazzola, Dr.
Universidade Federal de Santa Catarina
mazzola@inf.ufsc.br

Prof. Rosvelter Coelho da Costa, Dr.
Universidade Federal de Santa Catarina
coelho@inf.ufsc.br

Sumário

1. Introdução.....	9
2. VoIP.....	10
2.1 Protocolos VoIP:.....	12
2.1.1 SIP(Session Initiation Protocol):.....	12
2.1.2 RTP (Real-time Transport Protocol):.....	13
2.1.3 IAX (Inter-Asterisk Exchange Protocol):.....	13
2.1.4 H.323:	13
2.2 CODECs:.....	14
3. Asterisk.....	16
3.1 Arquitetura do Asterisk	16
3.1.1 As APIs	17
3.1.2 O núcleo do sistema	18
3.2 Modos de operação do Asterisk.....	19
3.2.1 O Asterisk operando como SIP Proxy.....	19
3.2.2 O Asterisk operando como B2BUA	21
4. Alta-Disponibilidade de Servidores	23
5. Balanceamento de Carga	27
6. VoIP com alta disponibilidade e balanceamento de carga.....	29
7. Teste de carga em um servidor Asterisk.....	31
7.1 Teste 1 – Cem chamadas geradas	35
7.2 Teste 2 – Cento e cinqüenta chamadas geradas	36
7.3 Teste 3 – Duzentas chamadas geradas	37
7.4 Teste 4 – Duzentas e cinqüenta chamadas geradas	38
7.5 Teste 5 – Trezentas chamadas geradas	39
7.6 Teste 6 – Trezentas e cinqüenta chamadas geradas.....	40
7.7 Teste 7 – Trezentas e setenta e cinco chamadas geradas	40
7.8 Análise dos testes	42

8. Soluções de alta-disponibilidade e balanceamento de carga para VoIP	44
8.1 DUNDi	46
8.2 Ambiente ideal.....	48
8.2.1 Camada BD.....	49
8.2.2 Camada Asterisk.....	50
8.2.3 Camada DNS	51
8.2.4 Camada Cliente	51
8.3 Ambiente com localização de extensões via DUNDi.....	51
8.4 Ambiente experimental.....	53
9. Testes realizados no Ambiente Experimental.....	56
9.1 Testes de alta-disponibilidade	56
9.2 Testes de balanceamento de carga	57
9.3 Conclusão sobre os testes	57
10. Conclusão e Trabalhos Futuros.....	58
10.1 Conclusão	58
10.2 Trabalhos Futuros	58
Anexos	60
Anexo 1 – Implementando uma Solução VoIP Baseada em Asterisk com Alta-Disponibilidade e Balanceamento de Carga	Error! Bookmark not defined.
Anexo 2 – Configuração do plano de discagem para busca de ramais em um Banco de Dados.....	74
Referências Bibliográficas	75

Lista de Figuras

Figura 1: Funcionamento de um SIP Proxy.....	20
Figura 2: Funcionamento do Asterisk em modo B2BUA utilizando protocolo SIP.	21
Figura 3: Exemplo clássico de uma estrutura de alta-disponibilidade.	24
Figura 4: Estrutura física do ambiente de testes.	31
Figura 5: Estrutura lógica do ambiente de testes.	32
Figura 6: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 1.	35
Figura 7: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 2.	36
Figura 8: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 3.	37
Figura 9: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 4.	38
Figura 10: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 5.....	39
Figura 11: Gráfico de chamadas ativas e porcentagem de CPU livre ao longo do tempo como resultado obtido no teste 6.	40
Figura 12: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 7.....	41
Figura 13: Gráfico de média de chamadas simultâneas e uso da CPU nos testes aplicados	42
Figura 16: Ambiente ideal.....	48
Figura 17: Ambiente com DUNDi	52
Figura 18: Ambiente experimental.....	54

Lista de Tabelas

Tabela 1: CODECs e banda consumida durante a chamada.....	15
Tabela 2: Disponibilidade, downtime e exemplos de ambientes computacionais .	25
Tabela 3: Exemplo de tabela DNS SRV	28
Tabela 4: Resultado geral dos testes	34

Resumo

Esse trabalho tem como principal objetivo formar um documento que sirva como fonte de referência para responsáveis por projetos de implantação de VoIP em ambientes onde alta-disponibilidade e balanceamento de carga sejam necessários.

É abrangido desde uma análise de quantas chamadas são suportadas por servidores comerciais, ou seja, sem necessidade de hardware específico até a sugestão de algumas possibilidades de implantação do ambiente em diferentes formas, assim como as vantagens e desvantagens de cada um.

Foram utilizadas somente ferramentas de software livre para que não houvesse limitações nas implementações por licenças e por essas se apresentarem totalmente satisfatórias para realizar as tarefas que foram demandadas na concepção do trabalho.

1. Introdução

O mercado abraçou fortemente o conceito de VoIP e hoje vem investindo muito nessa idéia. As corporações estão migrando boa parte de seus sistemas para a internet e agora se observa que é possível, e mais, é muito vantajoso migrar também o serviço de telefonia para trafegar sobre a rede de computadores que já abrange a maior parte das empresas e uma boa parte das residências no mundo. Deixa-se de ter dois serviços totalmente independentes aumentando a necessidade de dois tipos totalmente distintos de estrutura, mão de obra, etc. para ter uma única solução toda baseada na tecnologia IP que vem se difundindo em larga escala.

Com o advento crescente do VoIP surgiu o Asterisk com a finalidade de substituir e expandir as necessidades que se tem com um PABX tradicional. O software Asterisk é hoje o PBX IP mais utilizado no mundo “com mais de três milhões de downloads, apenas em 2007 [...]” [ASTERISK EXPERTS, 2008]. Por ser uma aplicação tão difundida seu reconhecimento vem aumentando bastante e tende a aumentar ainda mais, seguindo o crescimento do mercado de VoIP.

Tal expansão se deve a várias vantagens que se consegue utilizando soluções de voz sobre IP. A principal vantagem que conquista a atenção para a maioria dos projetos é a redução dos custos mensais com telefonia, já que as tarifas de operadoras VoIP são, em geral, mais reduzidas em relação às tarifas cobradas pelas operadoras de telefonia tradicional. A redução de custos é ainda mais drástica quando se usa o VoIP para fazer a comunicação matriz-filial, onde o custo será somente o do link entre os sites sem serem contabilizados custos por pulsos ou minutos. A convergência dos serviços, conforme comentado acima, também é uma vantagem bastante interessante.

Em comparação com o serviço de telefonia tradicional o VoIP ainda não tem um número de usuários tão expressivo, visto que no Brasil é bem mais barato comprar um telefone e contratar uma linha telefônica comparado ao custo de comprar um computador ou dispositivo IP e contratar um link de internet. Então as soluções de voz sobre IP ainda estão mais focadas nas corporações.

A qualidade das chamadas e a disponibilidade são dois pontos-chave que os administradores dos sistemas VoIP estão sempre atentos, já que esse sistema de comunicação é bem mais sensível em relação à telefonia tradicional. A qualidade da rede, dos dispositivos, dos servidores, etc. interfere diretamente na qualidade do serviço.

Esse trabalho visa abordar uma solução para dar mais robustez aos sistemas VoIP, trazendo opções de alta-disponibilidade e balanceamento de carga entre servidores utilizando Asterisk e alguns softwares que darão suporte à implantação da arquitetura.

O restante deste relatório está organizado da seguinte forma. O capítulo 2 apresenta uma visão geral sobre VoIP. Em seguida no capítulo 3 é feito um estudo sobre o software Asterisk e sua arquitetura. Nos capítulos 4 e 5 são apresentados conceitos e aplicativos de alta disponibilidade e balanceamento de carga, respectivamente. Por fim o capítulo 6 traça os objetivos do trabalho, as etapas para chegar ao objetivo e um cronograma das atividades que estão planejadas para o desenvolvimento completo do trabalho.

2. VoIP

A sigla VoIP vem de Voice over Internet Protocol (Voz sobre o protocolo IP), que significa a transmissão de voz usando o protocolo IP. Fazendo uso dessa tecnologia passa a existir um serviço especializado em telefonia sendo transmitido integralmente através das redes de computadores ou podendo, também, ser integrado ao sistema analógico de telefonia através de adaptadores que fazem tal conversão.

Segundo Meggelen, Madsen e Smith (2005) uma revolução está ocorrendo com o advento da tecnologia de voz sobre IP. Demorou um bom tempo para acontecer, mas agora que começou, não irá parar. Meggelen, Madsen e Smith tratam o processo de migração da telefonia para o mundo IP como inevitável principalmente pelas vantagens que se obtêm com essa tecnologia. Entre elas as principais são: a convergência, a mobilidade e a redução de custos.

A convergência é a integração ou migração dos serviços de telefonia convencional à rede de dados. Já foram mencionadas algumas vantagens da convergência no capítulo 1 desse trabalho. Onde se deixa de ter um custo operacional e de infra-estrutura dobrado. Como a estrutura de rede já é fundamental em qualquer ambiente empresarial nada de novo precisa ser feito para implementar VoIP na empresa. No entanto, podem ser necessárias alterações e melhorias no ambiente para garantir a qualidade das chamadas. Idealmente sugere-se que a rede por onde será trafegada a voz seja inteiramente separada da rede de dados da empresa, já que essa é muito mais sensível às intempéries do ambiente. Caso haja o compartilhamento dos recursos de rede para voz e dados é essencial que se tenha um grande controle nos requisitos de qualidade de serviço (QoS) exigidos por ambos os tipos de tráfego. Portanto a

robustez e qualidade da rede é um dos principais fatores a se considerar na elaboração de um projeto de voz sobre IP.

A mobilidade é possível pois o ramal deixa de estar atrelado a um ponto físico na rede de telefonia e passa a ser relacionado ao colaborador. Dessa forma ele pode se manter operacional de qualquer lugar interno ou externo à estrutura da empresa e ainda assim manter todas as funcionalidades relacionadas ao ramal. No caso do ramal estar em um ponto externo da empresa ele irá precisar de algum meio de acesso ao servidor de telefonia. Normalmente o meio utilizado é a internet por ter grande disponibilidade. Porém, deve-se ter um cuidado maior aos requisitos de segurança quando utilizando esse meio. Soluções de acesso via VPN, por exemplo, já podem dar maior confiabilidade de que os pacotes de voz não sejam interceptados.

A redução de custos vem como consequência dos dois fatores acima já que podem ser reduzidos os custos com ligações para os telefones dos funcionários, pois podem receber ou originar chamadas através de seus ramais. A diminuição dos custos é ainda mais abrupta quando se trata da comunicação entre matriz e filial ou entre filiais. A comunicação entre os sites será feita inteiramente pela rede IP, ou seja, livre de minutagens de qualquer tipo de operadora, o custo está restrito apenas ao link que interliga os ambientes. Como esse link normalmente já existe devido a necessidades de interligação de outros sistemas integrados comuns nas empresas, diz-se que esse tipo de chamada é feita a custo zero.

O preço das tarifas cobradas pelas operadoras de telefonia tradicional normalmente são mais altos que as cobradas pelas operadoras VoIP devido ao tipo de tecnologia utilizada por cada uma delas. Atualmente inclusive as operadoras de telefonia tradicional estão utilizando VoIP em algum momento da terminação das chamadas que provêm, também com o objetivo de reduzir o seu custo operacional.

O retorno do investimento feito para implementar a telefonia sobre IP é observado de médio a longo prazo variando de acordo com as tarifas conquistadas junto às operadoras e demais custos acima citados. Porém as funcionalidades e controle que se tem com um sistema que está no auge e no topo da tecnologia de comunicação atual devem ser considerados no cálculo do retorno do investimento.

2.1 Protocolos VoIP:

Os protocolos são responsáveis pelo estabelecimento, manutenção e finalização das chamadas. Em VoIP os protocolos mais utilizados são os seguintes:

2.1.1 SIP(Session Initiation Protocol):

O protocolo SIP encontra-se na sua segunda versão que está definida na RFC 3261 publicada no ano de 2002. Esse protocolo encaixa-se na camada de aplicação do modelo de referência OSI e é responsável por estabelecer, alterar e finalizar sessões multimídia, entre elas, as chamadas VoIP.

Em sua estrutura o protocolo SIP se parece com o protocolo HTTP, sendo também um protocolo baseado em texto e cliente/servidor, ou seja, implementa métodos de requisição e resposta na comunicação. Pode transportar qualquer carga independentemente, assim, de protocolos da camada de transporte.

Por ser um protocolo bastante difundido nos ambientes multimídia, o protocolo SIP também é amplamente utilizado em soluções de voz sobre IP. Sendo utilizado como padrão de várias arquiteturas atualmente.

O protocolo SIP não transporta qualquer tipo de mídia em seus pacotes. Para chamadas de voz sobre IP o protocolo utilizado para fazer tal transporte é o RTP.

2.1.2 RTP (Real-time Transport Protocol):

Segundo [RFC 1889], “o RTP provê funções de transporte de rede fim-a-fim ajustadas para aplicações transmitindo dados em tempo-real, como áudio, vídeo ou dados simulados, sobre serviços de rede multicast ou unicast. O RTP não contempla reserva de recursos e não garante qualidade de serviço para serviços em tempo-real.”

Em ambientes de voz sobre IP o RTP é utilizado em conjunto com o protocolo SIP para fazer o transporte do áudio das chamadas já que o segundo não implementa esse tipo de transporte.

2.1.3 IAX (Inter-Asterisk Exchange Protocol):

O protocolo IAX foi desenvolvido pela Digium (empresa que detém os direitos autorais do Asterisk e que iniciou seu desenvolvimento) com o principal objetivo de interligar servidores Asterisk. Atualmente encontra-se na versão 2 e é definido na RFC 5456.

Por enquanto tem pouca expressividade fora de ambientes que operam com servidores Asterisk e existem poucos dispositivos, tais como ATAs, telefones IP e softphones que trabalham com esse protocolo.

No entanto tem relevância, pois o mercado de Asterisk está em franco crescimento e também pela sua principal vantagem: usa apenas uma porta para transportar a sinalização e o áudio, o que o torna muito mais recomendável para operar em ambientes que tenham NAT – a maioria atualmente.

2.1.4 H.323:

A primeira versão da especificação do H.323 foi aprovada em 1996 pelo grupo de estudos 16 do ITU e a segunda em janeiro de 1998. O H.323 faz parte

de uma série de padrões de comunicações que permitem vídeo conferência e VoIP através de redes [SOUZA e CUNHA, 2004].

O H.323 é um dos protocolos mais maduros, robustos e utilizados nos equipamentos que utilizam VoIP por ser um dos mais antigos e definido especificamente com a finalidade de comunicações multimídia. Ele é o protocolo padrão utilizado em gateways de voz sobre IP das marcas Cisco, Polycom e Siemens, por exemplo, além de estar presente na maioria das centrais telefônicas que possibilitam VoIP.

2.2 CODECs:

A voz humana é transmitida em ondas senoidais através do meio. No entanto essas ondas têm de ser transformadas de sinal analógico para digital para que a comunicação na rede de dados seja possível. Esse processo de transformação é chamado digitalização do áudio.

No processo de digitalização o sinal contínuo do áudio é transformado em valores discretos por meio de amostragens analisadas com uma frequência pré-determinada. Para fazer a codificação e decodificação dos sinais fazendo compressão para que ocorra um melhor aproveitamento da banda existem os CODECs (enCOders/DECoders).

A tabela abaixo mostra alguns CODECs utilizados em VoIP, a taxa de amostragem que cada um exerce, a banda que é consumida pelos pacotes e a banda nominal – considerando o pacote e os cabeçalhos – consumida pelos CODECs:

Codec	Taxa de Amostragem (kHz)	Banda (Kbps)	Banda Nominal (Kbps)
G.711	8	64	87.2
G.723	8	5.3	20.8

		6.3	21.9
G.729	8	8	31.2
GSM	8	13	Desconhecido
ILBC	Desconhecido	15	Desconhecido
		24	47.2
G.726	8	32	55.2

Tabela 1: CODECs e banda consumida durante a chamada.

3. Asterisk

Asterisk é um PBX IP que roda como um serviço em um servidor Linux, Windows, Mac ou BSD.

Esse software foi desenvolvido inicialmente por Mark Spencer com o objetivo de montar um PBX IP que satisfizesse as necessidades de sua empresa de suporte em Linux. Em alguns meses ele abriu o código e lançou o software para a comunidade Open Source ajudá-lo no desenvolver. Desde então o Asterisk vem crescendo rapidamente em desenvolvimento e utilização, sendo atualmente o PBX IP mais utilizado no mundo, conforme mencionado no capítulo 1.

O Asterisk tem todas as funcionalidades de qualquer PBX do mercado, com a vantagem de ser de código aberto, podendo, assim, ser utilizado e desenvolvido sem qualquer custo ou necessidade de licença.

Ele é capaz de trazer grande interoperabilidade entre plataformas, pois faz a conversão entre vários CODECs e protocolos do mercado. Elimina a necessidade de grande parte dos módulos externos que disponibilizam recursos avançados fundamentais na maioria das centrais telefônica, como: música em espera, correio de voz e URA (Unidade de Resposta Audível).

O mercado já conta com várias soluções baseadas no aplicativo para as mais diversas atividades de negócios, agregando maior valor e confiabilidade a um projeto de VoIP baseado em software livre.

3.1 Arquitetura do Asterisk

Segundo Asterisk (2008), o Asterisk foi desenvolvido para prover máxima flexibilidade. APIs específicas são definidas em volta de um avançado núcleo que é o sistema PBX. Dessa forma o núcleo gerencia as interações entre as APIs gerenciando a conversão de CODECs e protocolos, executando as

funcionalidades e aplicações e estabelece a comunicação com o hardware, caso exista, para que possa ser feita a integração com o sistema tradicional de telefonia.

3.1.1 As APIs

São definidas quatro APIs básicas para dividir o funcionamento e entendimento do sistema. As APIs são conjuntos de módulos que podem ser carregados individualmente para que haja um maior controle de desenvolvimento e do que será utilizado na plataforma. Os módulos do Asterisk, por padrão encontram-se no diretório `/usr/lib/asterisk/modules`.

As APIs são:

- **API de Canal**

Essa API é responsável por identificar o tipo de sinalização da chamada entrante no sistema. Conforme o tipo de sinalização que se tem, são carregados módulos dinamicamente para fazer a comunicação de baixo nível com a aplicação. Os módulos que compõem essa API têm no início de seu nome a expressão “chan_”.

- **API de Aplicação**

A API de Aplicação é a responsável por executar todas as aplicações desenvolvidas modularmente no Asterisk. Alguns exemplos de aplicações são: Correio de voz através da aplicação Voicemail, DAC (Distribuição automática de chamadas) disponibilizada pela aplicação Queue e Conferência pela aplicação MeetMe. Os módulos dessa API são identificados pela expressão “app_”.

- **API de Tradução de CODECs**

Nessa API são carregados os módulos para fazer a codificação e decodificação de diversos formatos de áudio. Os módulos são identificados pela expressão “codec_”.

- **API de Formato de Arquivo**

Trabalha com a leitura e escrita de diversos formatos de áudio para serem armazenados em disco. Os módulos são identificados pela expressão “format_”.

3.1.2 O núcleo do sistema

Os módulos nos quais o núcleo do Asterisk é dividido são os seguintes:

- **Núcleo de Comutações**

Nessa parte do núcleo é onde são feitas as comutações entre as pontas de uma chamada. No Asterisk essa comutação é feita de forma transparente, ou seja, o usuário não sabe se está sendo feita conversão de CODECs ou protocolos. Pode-se considerar o Núcleo de Comutações como o ponto central de qualquer PBX.

- **Lançador de Aplicações**

É responsável por lançar as aplicações do Asterisk.

- **Tradutor de CODECs**

Essa parte do núcleo é onde são feitas as traduções de um codec para outro. O objetivo de tal tradução é permitir que o administrador ajuste o sistema para utilizar os recursos da melhor forma possível balanceando com qualidade das chamadas.

- **Scheduler e gerente de I/O**

Gerencia o escalonamento e faz o gerenciamento do sistema em baixo nível para que o sistema mantenha uma boa performance sob qualquer carga.

3.2 Modos de operação do Asterisk

O Asterisk é um software bastante flexível, devido a esse fator ele pode ser configurado para trabalhar como um SIP Proxy ou no modo B2BUA (Back to Back User Agent), sendo esse segundo o modo padrão e idealizado pelos desenvolvedores da plataforma. A seguir será descrito o funcionamento de cada modo de operação.

3.2.1 O Asterisk operando como SIP Proxy

O Asterisk pode operar como um SIP Proxy – apesar de não ser a função para a qual o software foi desenvolvido – fazendo apenas o encaminhamento de requisições SIP entre dois pontos.

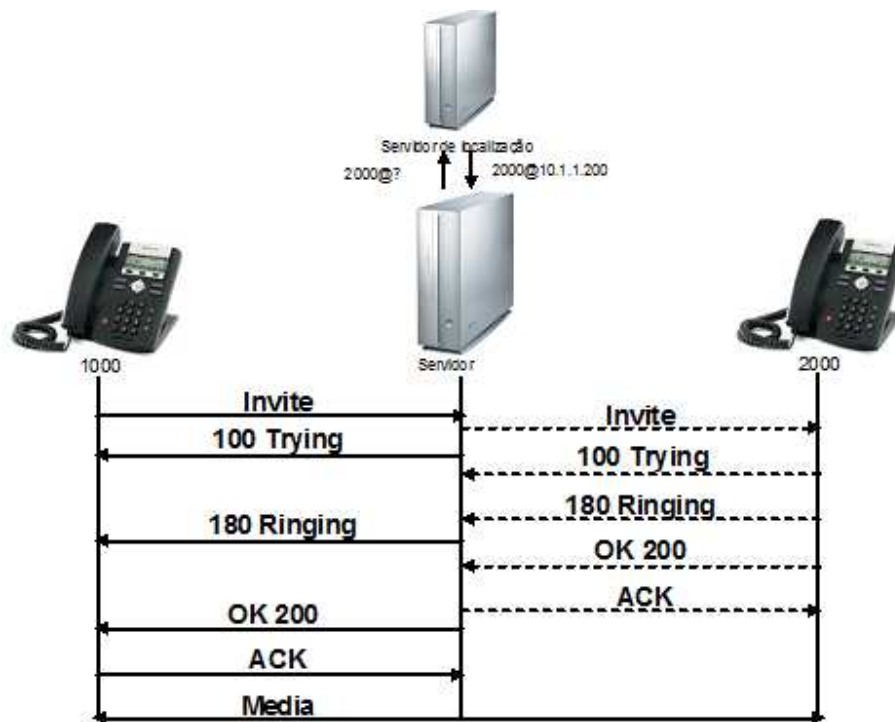


Figura 1: Funcionamento de um SIP Proxy.

A figura acima ilustra o funcionamento de um SIP Proxy, onde um ramal entra em contato com o servidor solicitando o ramal 2000 enviando uma solicitação chamada INVITE para o servidor. O servidor por sua vez consulta no servidor de localização onde está o ramal 2000. Ao ser encontrado é encaminhada a solicitação INVITE para o ramal desejado com o objetivo de iniciar o estabelecimento da chamada e enviada a resposta TRYING para o ramal chamador, informando que o servidor está tentando entrar em contato com o ramal chamado e qual a localização do mesmo. No momento que o destino for alcançado e estiver disponível é enviada a resposta RINGING ao servidor, que replica a resposta para a origem. No momento que o destino atende a chamada é enviada a mensagem OK ao servidor que novamente replica ao destino e a chamada é estabelecida.

Após o estabelecimento da chamada o fluxo da mídia é feito diretamente entre as duas pontas. Ou seja, nesse modo o Asterisk não tem qualquer controle sobre a mídia que será passada entre as partes e também não serão usadas as principais funcionalidades que o aplicativo disponibiliza.

Para a função de SIP Proxy recomenda-se que sejam usados outros softwares que foram desenvolvidos somente para esse fim. Duas opções de código aberto e bastante utilizadas no mercado são o OpenSIPS e o Kamailio.

3.2.2 O Asterisk operando como B2BUA

Somente nesse modo de operação se tem as funcionalidades do aplicativo disponíveis em sua totalidade, pois a mídia e a sinalização são controladas pelo servidor. Tornando possível o uso de gravações e caixas de mensagem, por exemplo.

Somente nesse modo que é possível trabalhar também com diversos protocolos e fazer a conversão de CODECS em uma mesma estrutura. A figura abaixo mostra o funcionamento desse modo:

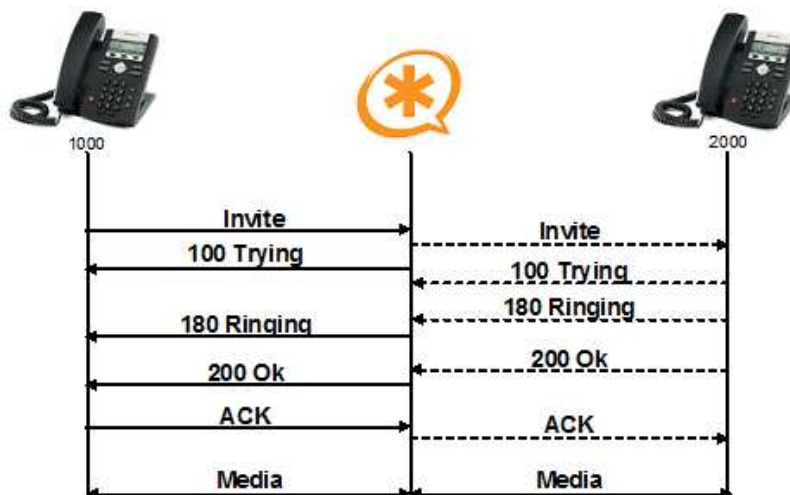


Figura 2: Funcionamento do Asterisk em modo B2BUA utilizando protocolo SIP.

Nota-se que a troca das requisições SIP entre as pontas da chamada é exatamente a mesma que a descrita acima no modo de operação SIP Proxy, porém o que os diferencia é que a mídia está sendo encaminhada do ramal para o servidor e vice-versa.

A figura acima mostra ambos os ramais trabalhando com o mesmo protocolo, no entanto não haveria problema se os protocolos fossem diferentes, já que o servidor iria receber a solicitação de um protocolo e enviaria a solicitação equivalente no protocolo utilizado pela outra ponta.

4. Alta-Disponibilidade de Servidores

Os sistemas computacionais são altamente importantes atualmente, no entanto esses sistemas são relativamente frágeis pois dependem de diversos fatores internos e externos que comprometem a sua disponibilidade.

Os componentes de hardware – componentes mecânicos – podem se comportar de forma inesperada quando interagirem já que não foram projetados para trabalharem necessariamente juntos e tratarem determinadas peculiaridades ou anormalidades que venham a ocorrer durante sua utilização.

Para que não se perca os benefícios dos sistemas informatizados quando ocorrerem tais faltas no ambiente, principalmente quando o sistema for de operação crítica, costuma-se usar o conceito de redundância. Esse conceito vem ganhando mais força nos últimos anos devido, entre outros fatores, à redução dos custos de hardware e ao aumento de serviços migrando para o computador. A queda nos preços impulsionou e trouxe maior confiabilidade a projetos que não tinham tantos recursos mas necessitavam de alta disponibilidade.

A redundância propõe que sejam eliminados o maior número de pontos de falha possíveis, assim quanto menor for o número de pontos únicos de falha maior será o nível de disponibilidade que se pode obter com o sistema. A figura a seguir é um exemplo clássico de alta-disponibilidade:

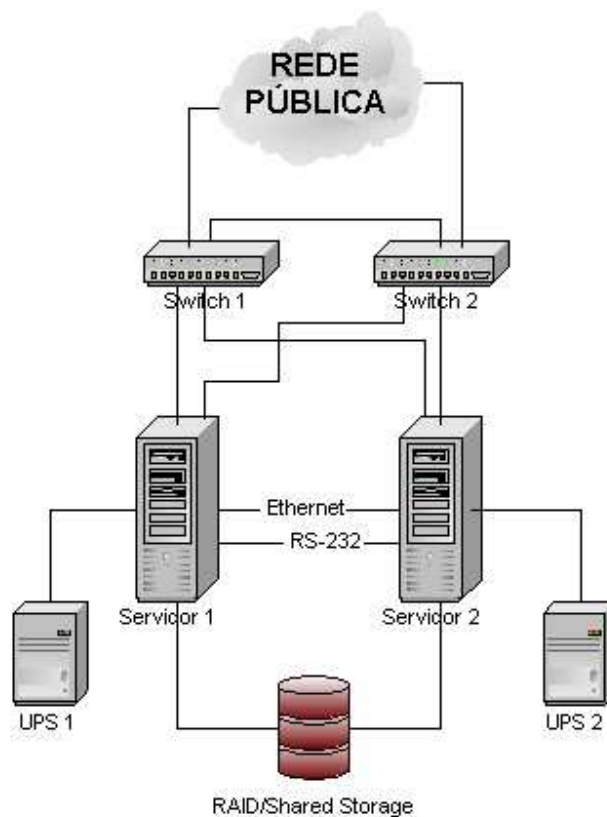


Figura 3: Exemplo clássico de uma estrutura de alta-disponibilidade.

Fonte: WIKIPEDIA, 2008

Nota-se na figura acima que todas as conexões de rede são redundantes, assim como os servidores e ativos de rede. Um ponto a ser salientado é a redundância, também, de no-breaks (UPS1 e UPS2) dessa forma se houver algum problema elétrico em um lado da estrutura o outro ainda pode se manter operacional.

É possível fazer o cálculo da disponibilidade que se vai ter em um ambiente através da fórmula: $\text{Disponibilidade} = \frac{\text{TMEF}}{\text{TMEF} + \text{TMDR}}$. Onde TMEF é o tempo médio entre falhas, TMDR é o tempo médio de reparo e o resultado –

Disponibilidade – é o percentual de tempo que o sistema ficará ativo durante o ano.

A tabela 2 traz exemplos de sistemas, disponibilidade apresentada por cada um e o tempo ocioso (downtime) por ano:

Disponibilidade	Downtime/ano	Exemplo de Ambiente
90%	36 dias 12:00:00	PC sem manutenção
99%	3 dias 15:36:00	PC com manutenção
99,9%	8:45:35	Cluster
99,99%	0:52:33	Multicomputador
99,999%	0:05:15	Sistema embutido (tecnologia PC)

Tabela 2: Disponibilidade, downtime e exemplos de ambientes computacionais
Fonte: Adaptado de LUNG, 2008

Geralmente o que se deseja é que quando um componente da estrutura falhe o seu equipamento redundante assuma de forma transparente as funções e com o menor impacto possível nessa transição. Para que isso seja possível as configurações aplicadas em um ponto da estrutura têm de ser espelhadas para o outro dispositivo. A solução open source para esse problema é uma suíte chamada de Linux HA (High Availability).

A aplicação da suíte Linux HA que faz o monitoramento de disponibilidade entre servidores é o software heartbeat.

Nesse aplicativo o servidor identificado como principal irá executar os serviços e terá um endereço IP virtual junto à sua interface de rede conectada à LAN (Local Area Network). Os clientes que forem acessar o sistema deverão fazê-lo através do IP virtual. O heartbeat manda sinais periódicos de keepalive entre um servidor e outro. Quando o servidor configurado como primário para aquele IP virtual ficar um tempo pré-configurado sem responder a esses sinais ele será considerado como indisponível pelo servidor redundante. A partir desse momento

o servidor redundante ou secundário irá assumir o IP virtual e os serviços que até então eram executados no servidor primário ou principal.

O espelhamento das partições do sistema de arquivos componente, também, da suíte Linux HA é feito pelo software DRBD (Distributed Replicated Block Device). Essa solução cria uma camada entre a partição que será espelhada e o sistema operacional. O sistema operacional deixa de interagir com a partição real e realiza o I/O destinado à partição nessa camada, que por sua vez replica as operações de I/O nas partições (sejam na mesma máquina ou em servidores separados) configuradas no software.

Outra opção quando não é necessário replicar uma partição inteira, mas sim somente alguns arquivos é aplicativo rsync. Esse aplicativo tem o funcionamento semelhante ao scp, porém ao ser executado ele não irá copiar todos os arquivos da origem para o destino incondicionalmente, mas sim fará uma comparação pela data de alteração dos arquivos da origem e fará a cópia somente dos que tiverem diferença entre os dois pontos. Caso o arquivo não exista no destino ele fará a cópia do arquivo normalmente. Um ponto importante a se salientar é que esse aplicativo somente pode ser executado em modo batch, assim para que hajam atualizações periódicas dos arquivos entre a origem e o destino, deve-se configurar o comando para ser executado em um agendador de tarefas como a cron do Linux, por exemplo.

5. Balanceamento de Carga

Uma arquitetura baseada em balanceamento de carga consiste em dividir as requisições feitas entre os servidores para evitar sobrecarga de um dos nós e aproveitar da forma mais igualitária possível os recursos disponíveis.

O balanceamento de carga é feito por uma aplicação que encaminha as requisições para os servidores de acordo com a estratégia que for configurada. Duas aplicações baseadas em DNS (Domain Name System) são bastante utilizadas em estruturas que demandam balanceamento de carga. São elas:

- **DNS Round Robin (DNS RR):** Nessa técnica de resolução de nomes o serviço deixa de resolver um nome para somente um endereço IP e passa a fazê-lo para uma lista de endereços que disponibilizam o mesmo serviço. Essa distribuição é feita de forma circular entre os endereços IPs configurados sendo respondido pelo endereço que estiver no topo da lista.

Supondo que se tenha três servidores (A, B e C) com os endereços IP 192.168.0.1, 192.168.0.2 e 192.168.0.3, respectivamente, sendo representados na lista de DNS nessa ordem padrão, ao chegar a primeira requisição ao servidor o nome será resolvido para o IP 192.168.0.1 e a lista será rotacionada. Quando chegar a segunda requisição o endereço que irá responder, pois está no topo é o 192.168.0.2, seguido pelo 192.168.0.3 e por fim está agora o 192.168.0.1.

Essa técnica, portanto, não emprega qualquer nível de priorização ou de controle de falhas das requisições para os servidores, portanto todos eles devem receber um número semelhante de requisições. Caso um dos servidores não esteja operante a requisição não será entregue, retornando erro para o cliente que a enviou.

- **DNS SRV:** As principais diferenças da técnica DNS SRV para a DNS Round Robin supracitada, são: pode-se configurar prioridades e pesos para

qual(is) servidor(es) irá(ão) responder às requisições preferencialmente; e os clientes caso detectem a falha na entrega na requisição a técnica permite que sejam atingidos servidores que tenham uma prioridade menor.

A prioridade é o primeiro fator considerado para fazer a resolução do nome, ou seja, todos os servidores que tiverem uma mesma prioridade tentarão ser atingidos antes que seja testado um servidor com prioridade menor.

Os pesos são utilizados para distribuir a carga de servidores com mesma prioridade. O valor do peso – entre 0 e 100 – será o percentual de requisições que o servidor irá receber.

A fim de exemplificar esse conceito considera-se essa tabela fictícia como sendo a tabela do DNS SRV:

Nome	Prioridade	Peso	Servidor
sip.empresa.com	10	70	192.168.0.1
sip.empresa.com	10	30	192.168.0.2
sip.empresa.com	20	0	192.168.0.3

Tabela 3: Exemplo de tabela DNS SRV

Seguindo os registros dessa tabela o servidor 192.168.0.1 irá receber 70% das requisições que chegarem ao nome sip.empresa.com e 30% serão encaminhados para o servidor 192.168.0.2. O servidor 192.168.0.3 irá receber todas as requisições quando os dois servidores com prioridade 10 não responderem.

6. VoIP com alta disponibilidade e balanceamento de carga

A telefonia é um recurso muito utilizado na comunicação das empresas devido à agilidade e maior facilidade que as pessoas têm de se comunicar falando que escrevendo através de e-mails ou sistemas de mensagem instantânea. Além da funcionalidade normal de interligar duas pessoas, foram agregadas várias funcionalidades que trazem ainda mais valor ao sistema telefônico, como URAs e serviços de contato com o cliente. Esse último é o ramo de negócios de algumas empresas bastante utilizado atualmente.

Com o VoIP sendo largamente utilizado, o sistema deve ser robusto e ficar o menor tempo indisponível possível, já que o sistema de telefonia legado tem como principal benefício a estabilidade, dessa forma um projeto de voz sobre IP não pode deixar a desejar nesse quesito mas sim aliá-lo aos demais pontos fortes da arquitetura. Devido a esses fatores surge a necessidade de se implementar técnicas de alta disponibilidade no ambiente.

Além de se eliminar os pontos únicos de falha podem-se aproveitar os equipamentos redundantes para executarem tarefas mesmo quando estão em segundo plano na estrutura. Para isso se implementa os conceitos de balanceamento de carga.

Esse conceito pode, também, ser utilizado quando o hardware que está à disposição individualmente não é suficiente para realizar todas as tarefas necessárias exigidas pelo projeto, então aplicam-se as técnicas de balanceamento de carga para que vários nós se comportem como um único nó central, ficando transparente para o usuário final quantos e quais são os dispositivos que compõem a arquitetura.

Esse trabalho de conclusão de curso tem por objetivo trazer uma solução de voz sobre IP que seja escalável e proveja confiabilidade baseado em softwares

open source. Dessa forma o projeto não fica limitado a disponibilidade de licenças ou compra de aplicativos dando mais liberdade para explorar as potencialidades do sistema.

7. Teste de carga em um servidor Asterisk

Para dar embasamento e identificarmos qual a carga real de chamadas simultâneas que um servidor comercial rodando Asterisk pode processar, elaboramos um plano de testes que irá focar no que existe de mais crítico no tocante a uso de CPU e memória durante a execução de muitas chamadas: fazer a passagem da mídia seja feita através do servidor e a conversão entre CODECS seja feita pelo Asterisk.

Os testes forma realizados utilizando a seguinte estrutura física:

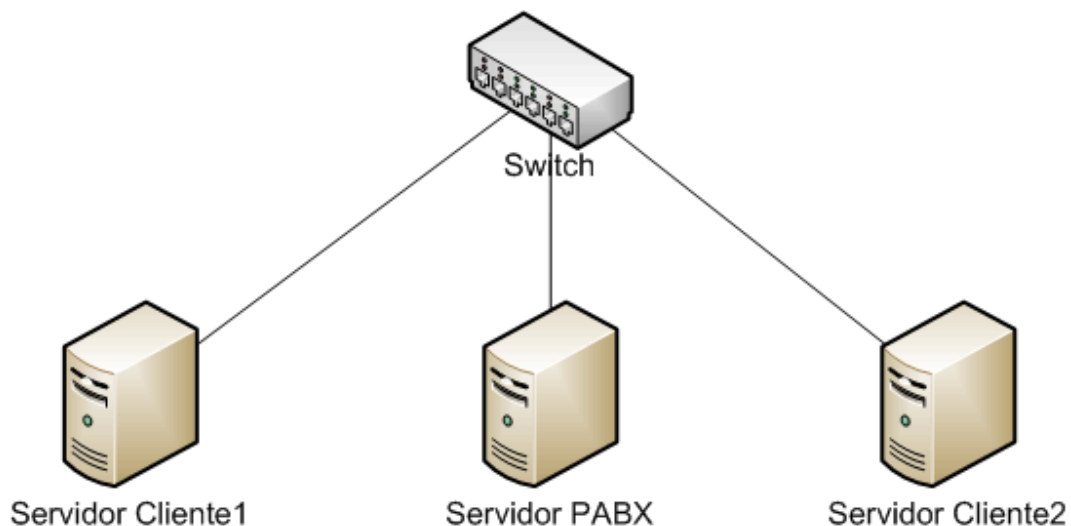


Figura 4: Estrutura física do ambiente de testes.

Onde:

- Servidor Cliente1: servidor responsável por gerar chamadas consecutivas e ininterruptamente para o Servidor PABX;
- Servidor PABX: recebe as chamadas originadas pelo Servidor Cliente1 e encaminha para o Servidor Cliente2. Esse servidor é o principal alvo dos testes.

- Servidor Cliente2: recebe as chamadas encaminhadas pelo Servidor PABX.

No Servidor PABX foram configuradas duas contas SIP para estabelecer a comunicação entre os três servidores. Sendo que cada conta permite o uso de somente um CODEC. Sendo que a estrutura lógica configurada é ilustrada na figura a seguir:



Figura 5: Estrutura lógica do ambiente de testes.

No Servidor Cliente1 foi executado um programa desenvolvido pelo próprio aluno na linguagem PHP para fazer a geração das chamadas. O aplicativo consiste em gerar um número pré-configurado de chamadas em um período também configurado. Ao detectar que chamadas foram desligadas o aplicativo irá gerar tantas chamadas quantas ele detectou que faltam para atingir a média de chamadas simultâneas que foi configurada.

No aplicativo, também é possível configurar incremento de chamadas em intervalos de tempo até chegar a um limite ou mesmo sem limite, dessa forma serão geradas chamadas no intervalo e número configurados até que o servidor sature ou venha a travar.

No Servidor Cliente2 as chamadas geradas pelo aplicativo são atendidas e um áudio começa a ser tocado. Ao receber esse atendimento o Servidor Cliente1 também começa a reproduzir um arquivo de áudio. Como cada um dos servidores clientes está reproduzindo áudios e enviando em CODECs diferentes para o Servidor PABX, esse fica obrigado a fazer a conversão dos CODECs em ambas as direções.

O monitoramento do uso da CPU e memória foi feito através do software dstat. Esse aplicativo faz o monitoramento desses recursos no servidor em intervalos de um segundo e escreve a saída em um arquivo no formato CSV juntamente com o timestamp de cada intervalo da verificação.

O aluno desenvolveu um script que também em intervalos de 1 segundo monitora a quantidade de chamadas simultâneas realmente ativas no Servidor PABX. Esse número é enviado juntamente com o timestamp de cada verificação para um segundo arquivo CSV.

Após o intervalo de geração e monitoramento das chamadas os dois arquivos CSV foram importados no software MS Excel e usando o recurso Pivot Table os registros são unidos através do timestamp, gerando um gráfico de número de chamadas simultâneas e uso da CPU versus o intervalo em que ocorram as ligações.

Para os testes apresentados forem analisados de maneira igualitária foi decidido que os testes seriam feitos em intervalos de dez minutos e incremento de cinquenta chamadas, ou menos de acordo com a necessidade, a cada novo teste. A captura dos resultados começou em cem ligações simultâneas, pois foi onde começou a se observar certa diferença na estabilidade do sistema.

Como o teste visa testar a capacidade do servidor gerenciar a mídia e a sinalização das chamadas, optou-se por não gerar ligações que ficassem ativas durante todo o período, mas sim que essas tivessem um tempo aproximado de 25 segundos e terminadas. Devido a esse fato há diferença no tempo de desligamento de uma chamadas, geração e estabelecimento de uma nova, pois as iniciais foram geradas simultaneamente e existe um intervalo de tempo até que todas pudessem ser completadas e posteriormente finalizadas. Também, enquanto algumas delas estão sendo geradas outras estão sendo terminadas e o aplicativo não consegue lidar paralelamente com esses dois fatos.

Outra consideração a se fazer é que os testes sempre passam dos dez minutos pois esperou-se que todas as chamadas em andamento fossem desligadas.

Foram realizadas duas chamadas a partir de um softphone com o mesmo CODEC que as chamadas eram geradas afim de verificar a qualidade das mesmas. As ligações foram feitas nos segundos de número duzentos e quatrocentos, ou seja, ao fim do primeiro-quarto e no início do quarto-quarto do tempo total de testes. Nesse teste foi analisada a qualidade das chamadas de forma empírica avaliando a qualidade do áudio pelo sentido e através da captura dos pacotes da chamada pelo aplicativo Wireshark.

Os resultados das medições feitas no servidor e pelo aplicativo Wireshark estão apresentados na tabela a seguir:

Teste	Limite de Chamadas Geradas	Média de Chamadas	Média de CPU Usada(%)	Jitter Médio das Chamadas de Teste(ms)	Atraso médio das Chamadas de Teste(ms)
1	100	91,66	26,92	0,719	19,983
2	150	136,15	38,22	0,683	19,994
3	200	181,40	50,22	0,666	19,990
4	250	228,64	62,36	0,467	19,990
5	300	271,65	70,18	0,507	19,993
6	350	310,65	77,12	0,919	20,012

Tabela 4: Resultado geral dos testes

Com os resultados apresentados acima observa-se que não há uma variação significativa no jitter e atraso médio das chamadas de teste. Portanto, esses dois campos não serão incluídos nas análises que virão a seguir apesar de serem campos importantes e que devem ser testados sempre para detectar possíveis problemas de rede que podem afetar a comunicação entre dispositivos de comunicação IP.

7.1 Teste 1 – Cem chamadas geradas

No primeiro teste foi configurado o software de geração de chamadas para que fossem feitas no máximo 100 chamadas simultâneas.

Como pode ser observado na Tabela 4 o uso da CPU ficou em um nível seguro de utilização. Os dois testes feitos com expectador humano apresentaram resultados satisfatórios, onde a qualidade do áudio que foi reproduzido não teve qualquer ruído.

O gráfico abaixo apresenta o resultado desse teste mostrando o número de chamadas livres e a porcentagem de CPU livre ao longo do período total desse teste.

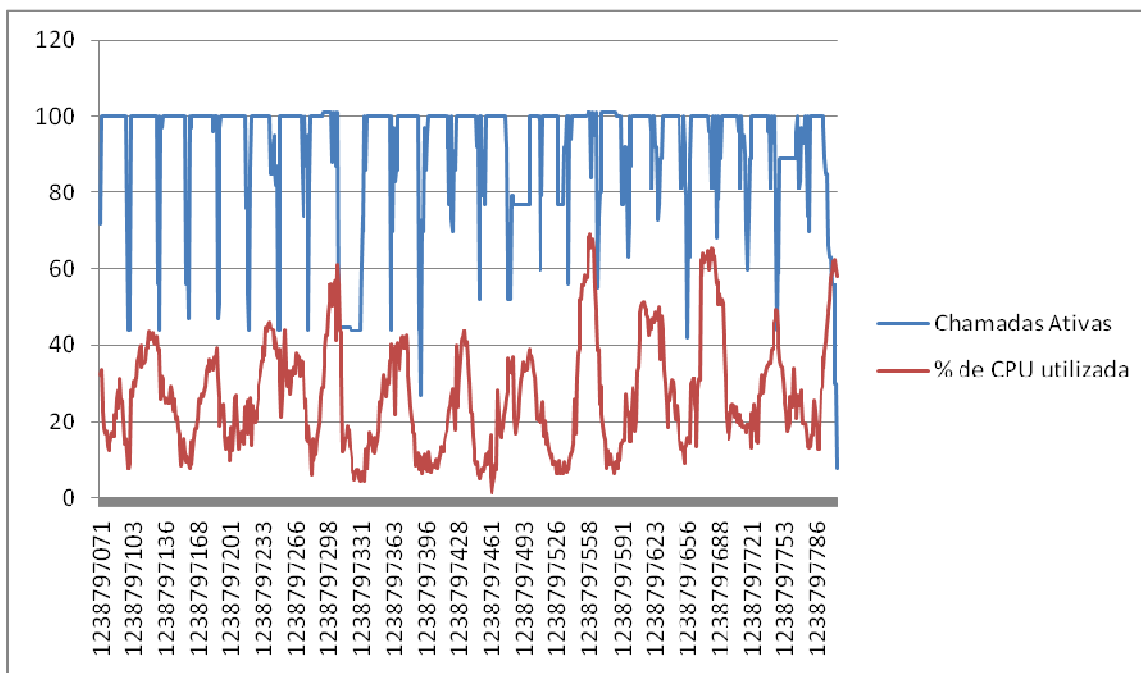


Figura 6: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 1.

7.2 Teste 2 – Cento e cinquenta chamadas geradas

No segundo teste o aplicativo de geração foi configurado para gerar no máximo 150 chamadas simultâneas.

Observando a média do uso da CPU nota-se que esse valor ainda ficou dentro de uma margem de utilização normal, porém a variação entre os pontos de utilização máxima e mínima da CPU foi mais ampla que a obtida no teste anterior. O gráfico abaixo ilustra o resultado desse teste:

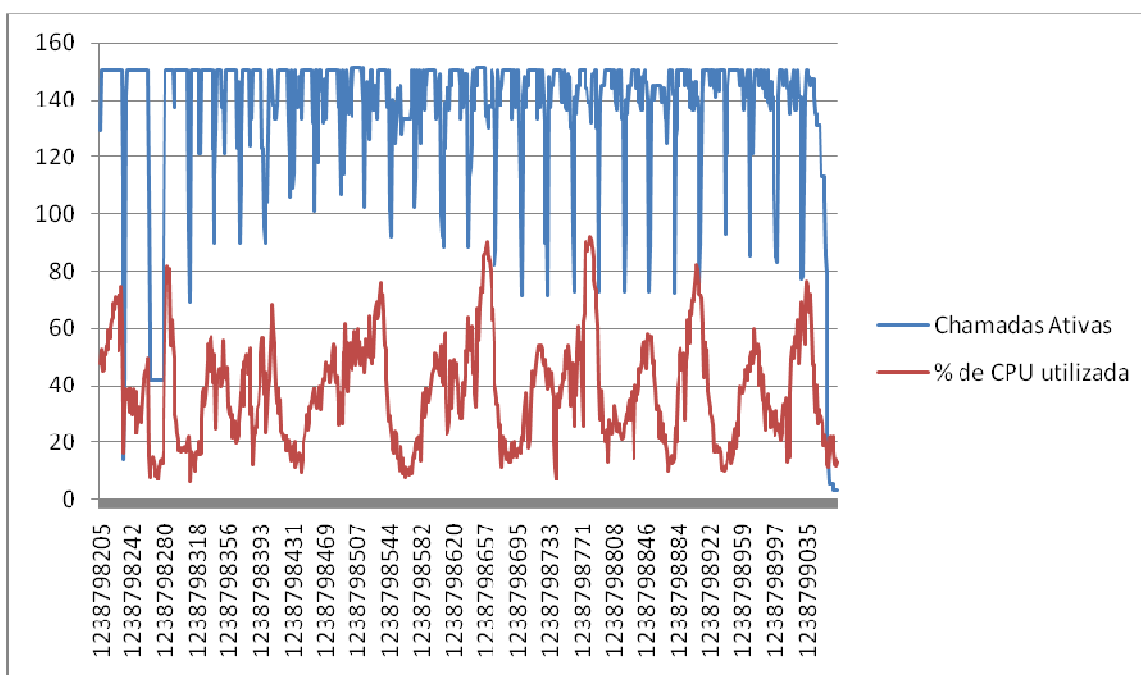


Figura 7: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 2.

A qualidade das chamadas feitas pelo expectador humano foram boas, porém houveram indícios de ruído que não chegam a atrapalhar o entendimento da mensagem nem geraram desconforto na comunicação, porém em relação ao primeiro teste a ligação teve uma ligeira deterioração.

7.3 Teste 3 – Duzentas chamadas geradas

No teste 3 foi configurado para que o número máximo de chamadas simultâneas geradas fosse de 200.

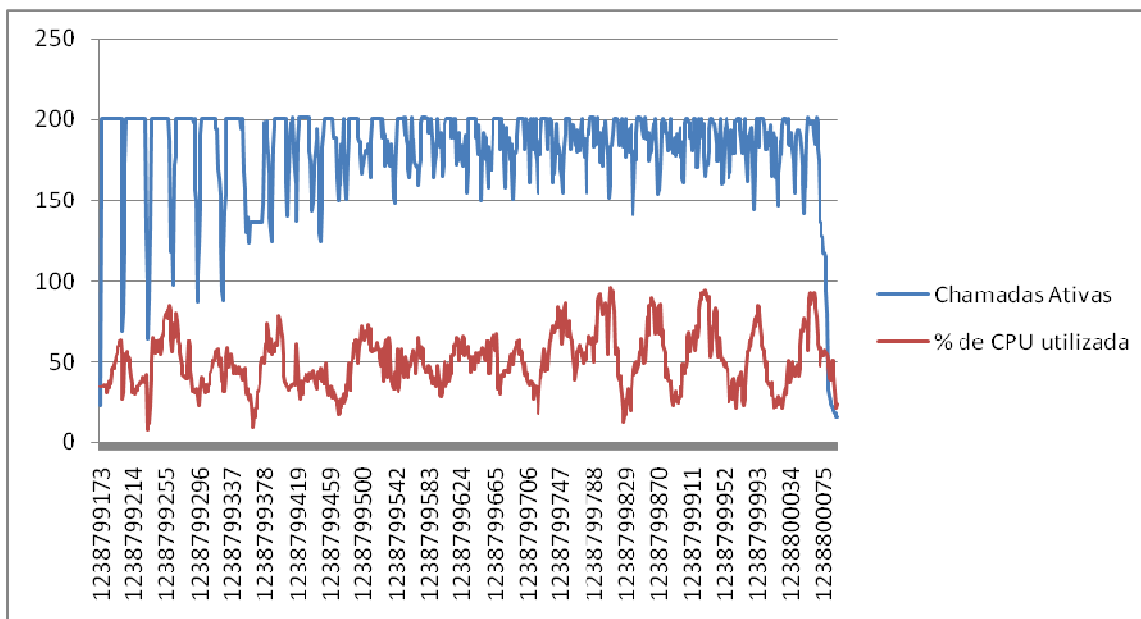


Figura 8: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 3.

Analisando o gráfico do testes 3, nota-se que aproximadamente após a metade do período de testes a instabilidade no uso da CPU aumentou, gerando picos e depressões mais salientes do que havia sendo apresentado. Esse fato se explica pelo aumento da frequência de chamadas sendo geradas e terminadas no servidor nesse período.

A qualidade das chamadas feitas pelo humano tiveram resultado semelhante à do teste 2, ainda considerada boa e totalmente inteligível.

7.4 Teste 4 – Duzentas e cinqüenta chamadas geradas

O teste 4 teve como objetivo observar o comportamento do servidor recebendo no máximo 250 chamadas simultaneamente.

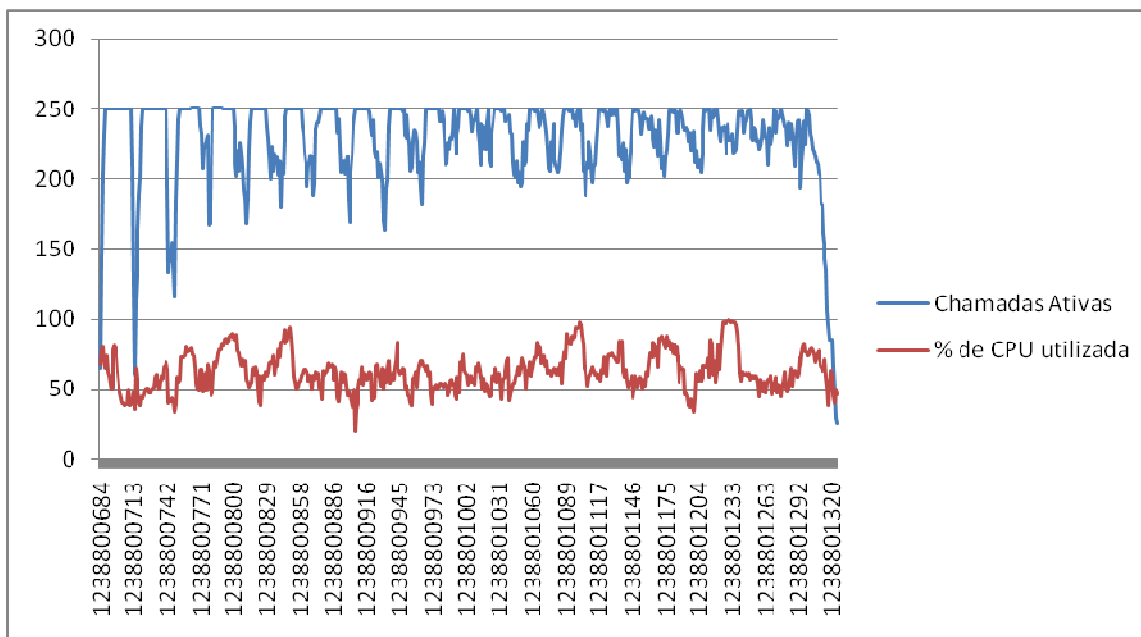


Figura 9: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 4.

Nesse teste, assim como no anterior, nota-se uma depreciação na performance do servidor na metade final do período. A primeira chamada gerada pelo usuário humano teve qualidade semelhante aos dois testes anteriores, no entanto a segunda chamada, que é gerada durante o último quarto do período de testes teve um aumento nos ruídos. Esse aumento nos ruídos ainda não atrapalhou o entendimento da mensagem porém gera certo desconforto para o usuário.

7.5 Teste 5 – Trezentas chamadas geradas

Com a configuração passando para no máximo 300 chamadas geradas simultaneamente no teste 5, foi obtido o seguinte gráfico:

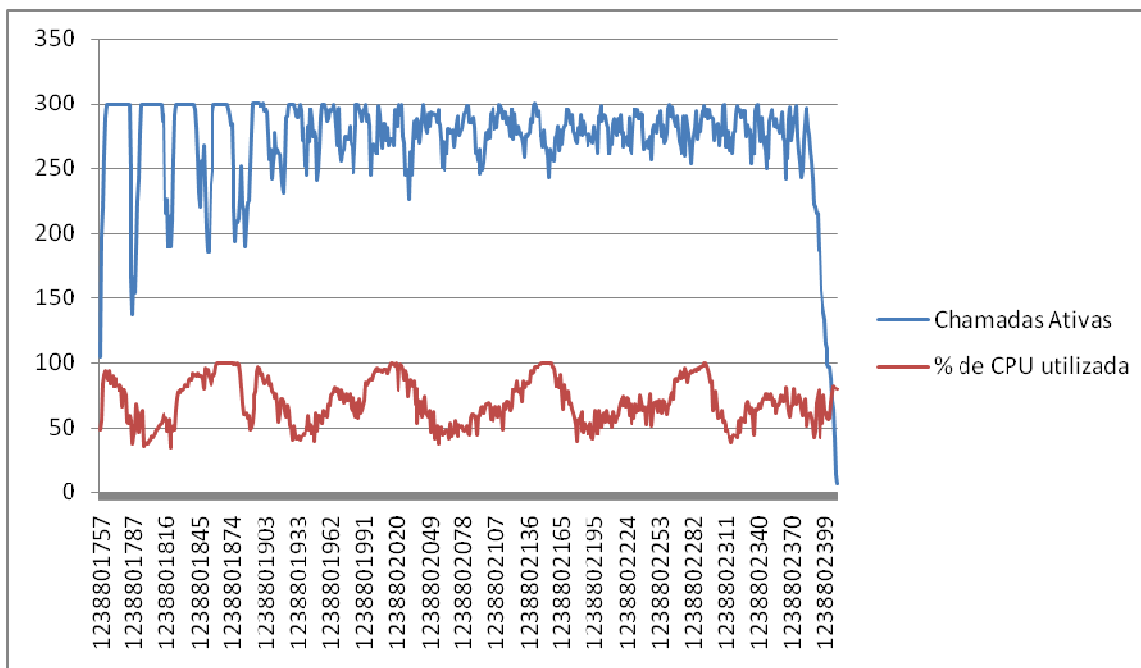


Figura 10: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 5.

Observou-se que com a diminuição da média de CPU livre, começaram a aparecer períodos maiores e mais frequentes de saturação do processador. Ao realizar as chamadas de verificação de qualidade o usuário observou que o desconforto que começou a ser observado no final do teste anterior continuou nas duas chamadas realizadas, da mesma forma que anteriormente os ruídos não atrapalharam o entendimento da mensagem.

7.6 Teste 6 – Trezentas e cinqüenta chamadas geradas

O teste 6 consistiu em gerar no máximo 350 chamadas ocorrendo ao mesmo tempo no servidor.

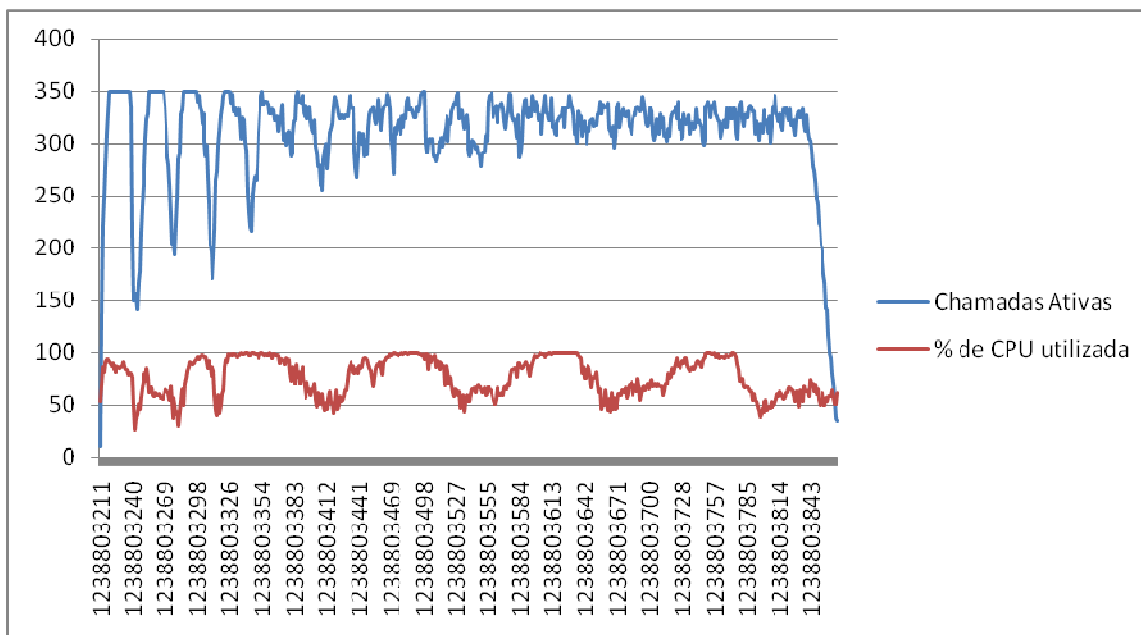


Figura 11: Gráfico de chamadas ativas e porcentagem de CPU livre ao longo do tempo como resultado obtido no teste 6.

Nesse teste a inconstância foi bastante grande no que se refere ao uso da CPU. A qualidade também apresentou nova degradação e aumento de ruídos, a partir desses testes notou-se picotes nas chamadas. Ainda a mensagem pode ser entendida completamente, mas o desconforto sentido pelo usuário foi relativamente grande.

7.7 Teste 7 – Trezentas e setenta e cinco chamadas geradas

O teste 7 não pôde ser completado pois o servidor de geração de chamadas, não suportou a carga imposta e derrubou o serviço durante os testes.

Por isso o último testes considerado plenamente válido é o teste 6. A seguir está o gráfico do teste enquanto o servidor Cliente 1 estava ativo.

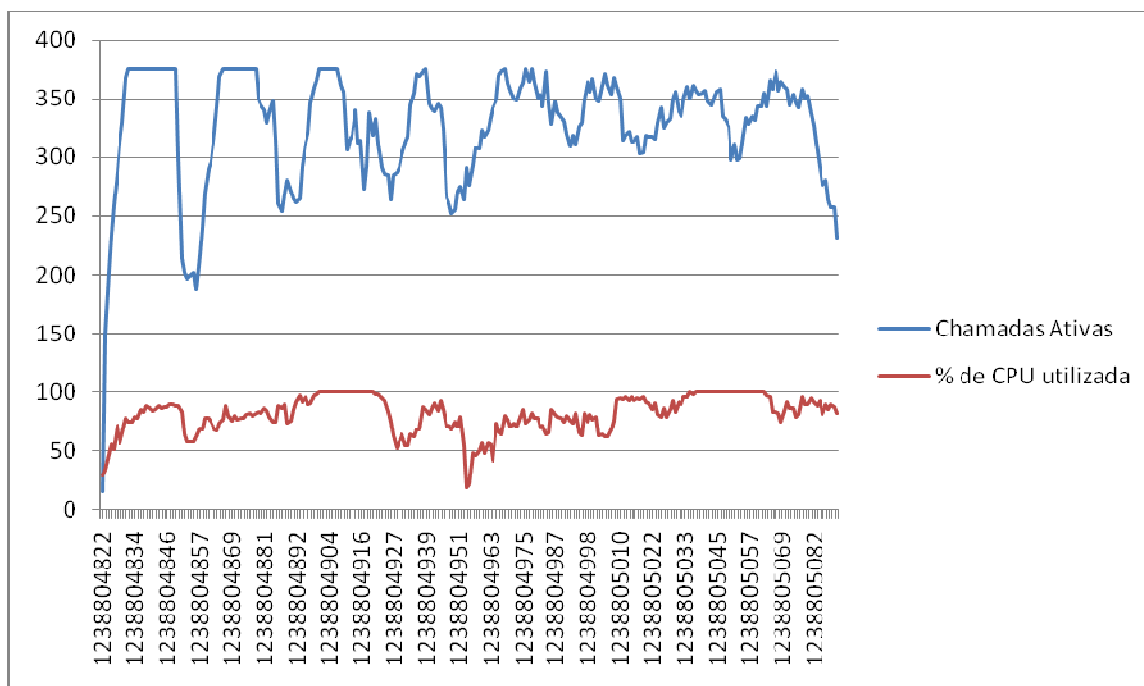


Figura 12: Gráfico de chamadas ativas e porcentagem de CPU utilizada ao longo do tempo como resultado obtido no teste 7.

Devido ao grande volume de chamadas sendo geradas e derrubadas o número de chamadas simultâneas ficou aquém do resultado desejado que seria acima de 340. Além desse fator o uso da CPU ficou totalmente abaixo do limiar considerado aceitável para manter a qualidade das chamadas. Quando realizada a primeira chamada de teste pelo usuário teve-se a confirmação de que não há mais condições de comunicação, pois os picotes e o ruído aumentaram, pela primeira vez não se teve completo entendimento da mensagem.

7.8 Análise dos testes

Com os testes realizados, constatou-se que o servidor começa a perder em estabilidade com o passar do tempo quando submetido a uma grande carga de chamadas. Conseqüentemente a qualidade dos serviços de telefonia de voz sobre IP tende a piorar proporcionalmente.

O gráfico abaixo mostra a comparação entre o número média de chamadas ativas e o uso de CPU ilustrando os dados que foram apresentados na Tabela 4.

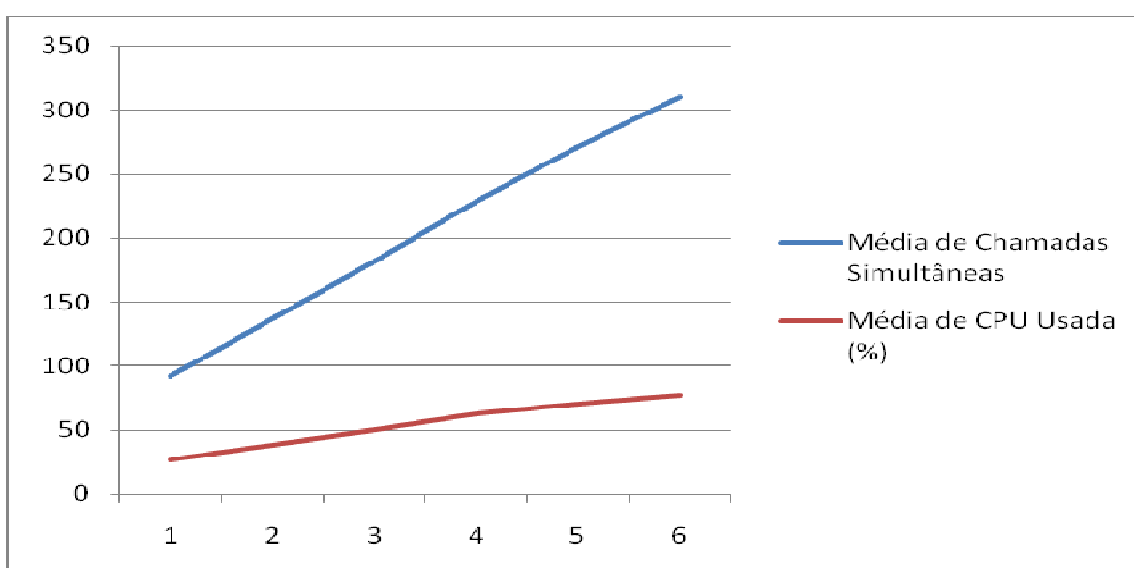


Figura 13: Gráfico de média de chamadas simultâneas e uso da CPU nos testes aplicados

Observa-se pelo gráfico um aumento quase que linear do uso da CPU do servidor nos seis testes com o aumento das chamadas ativas. Sendo que a partir já do teste 3 é ultrapassado o nível de 50% de utilização média.

O teste 7 não foi considerado válido, pois não pode ser realizado durante o intervalo de pelo menos 10 minutos estabelecido como parâmetro de validação dos testes. Apesar do servidor testado não ter falhado não foi possível obter todos os dados que estavam sendo medidos.

Para o hardware do Servidor PABX percebeu-se que quando usado somente com o serviço do Asterisk rodando, o volume de ligações não poderia ultrapassar de duzentas chamadas. Pois, apesar dos testes mostrarem bons resultados ainda em 250 e 300 chamadas, não foi considerado que estejam sendo utilizados outros serviços usuais e atividades também corriqueiras como consulta de ramais e escritas de CDR em bancos de dados; requisições de registro de dispositivos IP ou ainda aplicações do próprio aplicativo, como correio de voz e gravações por exemplo, que exigem algum processamento extra e aumentam a utilização da CPU e memória do servidor.

Mesmo que esses serviços individualmente não apresentem grande perturbação nas medições feitas, rodando em conjunto e sob forte carga – como foi o caso do Asterisk – com certeza influem no processamento e qualidade de ligações.

8. Soluções de alta-disponibilidade e balanceamento de carga para VoIP

Para como foi observado nos resultados de carga sobre um servidor Asterisk, ficou claro que esse serviço consegue gerenciar com qualidade um número bastante grande de chamadas. Porém, uma grande empresa que tenha diversos ramais utilizando com grande frequência os recursos de telefonia não pode aceitar que apenas um servidor seja o ponto central da sua arquitetura. Caso isso ocorra, o sistema de telefonia estará muito frágil e qualquer falha que venha a ocorrer nesse ponto central irá comprometer totalmente o funcionamento do sistema telefônico.

Em empresas que têm o telefone como principal ferramenta de trabalho, como call e contact centers, se a telefonia parar a produção da empresa pára também. Para evitar isso foram propostas algumas arquiteturas que visam eliminar o ponto único e central de falha fazendo redundância do servidor Asterisk utilizando apenas aplicativos open source.

O trabalho se limita a apresentar essa implementação somente nos servidores. Cabe lembrar que a alta disponibilidade também deve estar presente em todos os nós ativos de rede e fontes de energia para que a indisponibilidade desses equipamentos não venha a prejudicar ou que impacte o mínimo possível na operação normal do ambiente.

A seguir nesse capítulo serão apresentadas algumas das arquiteturas possíveis a se implantar em um ambiente nesses moldes citando prós e contras de cada uma delas. Uma delas foi escolhida e implementada em laboratório para verificar a facilidade de implantação e expansão do ambiente.

Em todas as arquiteturas optou-se por trabalhar com o recurso ARA (Asterisk Realtime Architecture) disponível no Asterisk. Essa funcionalidade

possibilita que algumas configurações sejam armazenadas no banco de dados e qualquer alteração que for feita nesses parâmetros não exige que seja feito o reload no módulo que sofreu modificação. Diferentemente do que acontece da configuração em arquivo, que a cada alteração que for feita o módulo que utiliza esses parâmetros deverá sofrer um processo de reload para que seja atualizado os seus valores.

Devido a algumas limitações encontradas no ARA tocante ao plano de discagem, preferiu-se por usá-lo somente nas configurações referentes a ramais SIP e IAX, filas de atendimento e correio de voz (voicemail). As demais configurações de arquivo serão replicadas entre os servidores utilizando o aplicativo rsync que foi detalhado no capítulo 4.

O SGBD selecionado para fazer a interação com o Asterisk foi o MySQL, pois esse tem suporte nativo no Asterisk, não sendo necessário a compilação de qualquer módulo adicional para haver a comunicação entre os dois softwares. Outro fator que levou à escolha desse SGBD foi a facilidade de configuração e o excelente funcionamento do sistema de replicação que ele disponibiliza, dando a opção tanto de trabalhar em modo mestre/escravo (dados são replicados de um servidor primário para outro secundário) como mestre/mestre (os dados são replicados entre os dois bancos de dados). Esse recurso é indispensável para se trabalhar com alta disponibilidade de bancos de dados, como visto acima, um ponto crucial nas arquiteturas que seguirão.

O aplicativo escolhido para fazer a verificação de disponibilidade dos servidores foi o heartbeat detalhado, também, no capítulo 4. Essa escolha foi feita devido à grande facilidade de configuração do aplicativo e escalabilidade que ele proporciona, facilitando o gerenciamento e manutenção do sistema.

Foram elaboradas arquiteturas que utilizam o aplicativo DUNDi para fazer a localização dos ramais.

8.1 DUNDi

O DUNDi (Distributed Universal Number Discovery) foi desenvolvido pela Digium – mesma fabricante do Asterisk – para localização de números entre Asterisk de forma totalmente distribuída.

Ao contrário do ENUM, por exemplo, que mantém um repositório de números e a forma de como eles são contatados, o DUNDi faz essa consulta dinamicamente.

O software é configurado para procurar em servidores que ele conhece se uma determinada extensão é conhecida. Caso afirmativo o nó envia uma resposta a quem solicitou com a forma que ele deve ser acessado para que encaminhe a chamada até a extensão desejada. A figura abaixo ilustra essa consulta:

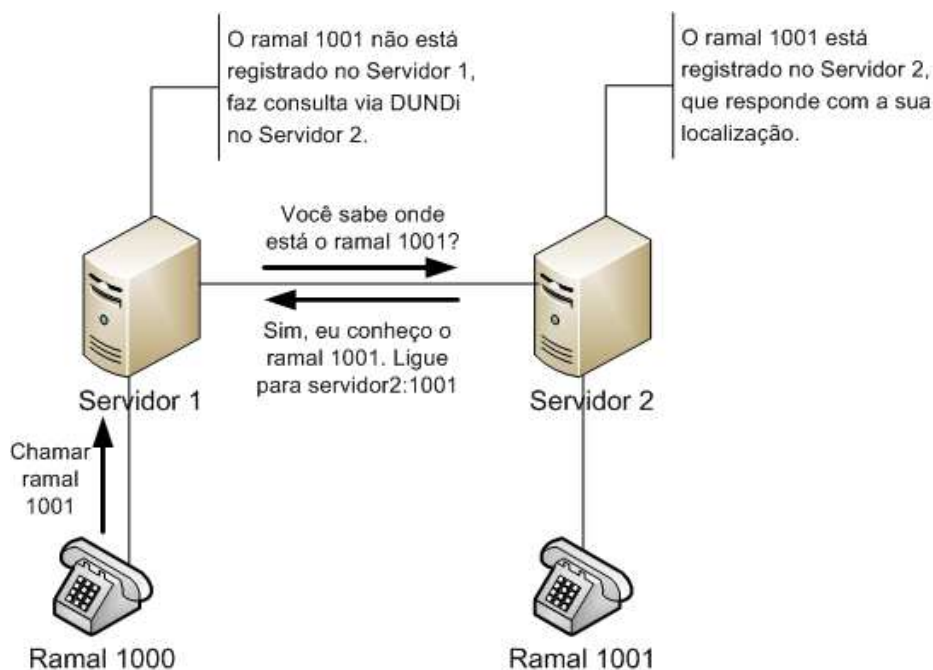


Figura 14: Consulta DUNDi entre dois servidores

Caso ele não conheça a extensão que se deseja, esse nó pergunta para outros servidores conhecidos por ele se eles possuem a extensão desejada e

assim sucessivamente até se chegar à localização do número ou até que todos os nós da malha forem atingidos e respondido negativamente à busca. A figura a seguir ilustra essa situação:

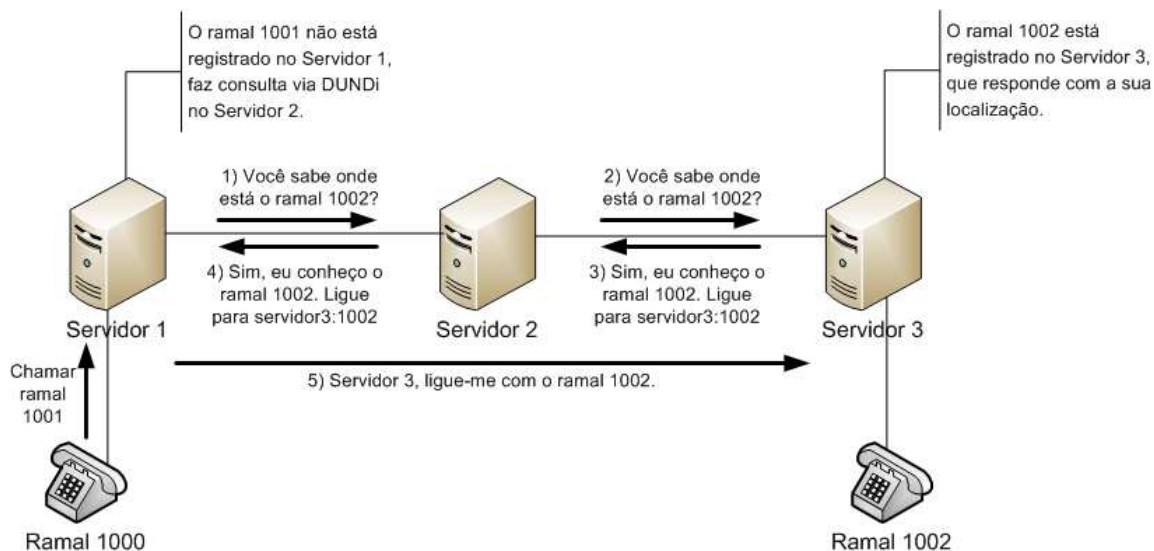


Figura 15: Consulta DUNDi entre servidores

No caso acima o Servidor 1 está buscando o ramal 1002, como ele não sabe a localização do ramal faz a consulta DUNDi ao Servidor 2. Esse, por sua vez, também não conhece a localização do ramal 1002 então replica a consulta para o Servidor 3. Esse servidor conhece a localização do ramal e responde ao Servidor 2 como deve ser feita a comunicação com ele. Como o Servidor 2 apenas encaminhou a consulta, ele irá repassar novamente para o Servidor 1 a resposta que recebeu do Servidor 3. Com posse da localização do ramal o Servidor 1 entra em contato com o Servidor 3 estabelecendo a comunicação.

Se o ramal 1002 não estivesse registrado no Servidor 3, esse enviaria uma resposta negativa no passo 3 da figura, sendo repassada para o Servidor 1, que iria dar o código de erro para o ramal 1000 informando que a extensão não foi encontrada.

8.2 Ambiente ideal

A figura abaixo ilustra a arquitetura considerada ideal para ser implementada em um ambiente de VoIP em alta disponibilidade e que ocorra balanceamento de carga entre os servidores de telefonia.

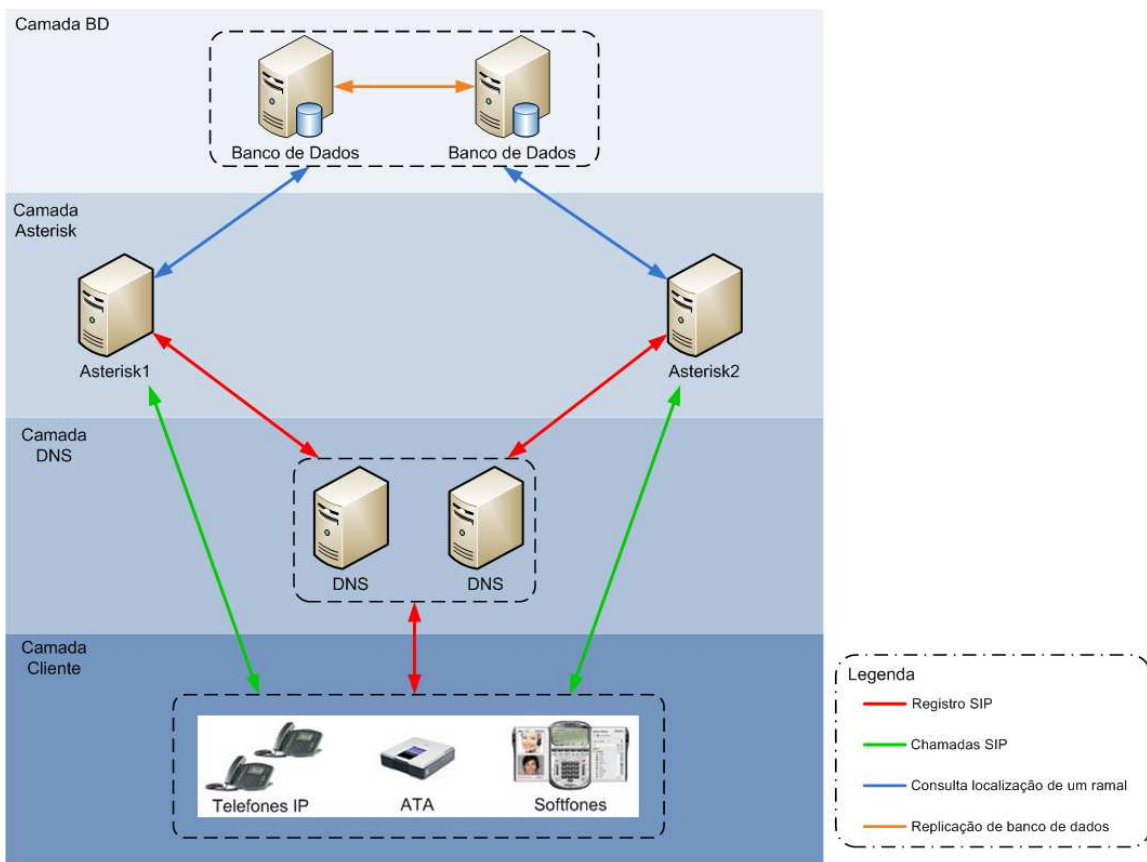


Figura 16: Ambiente ideal

Esse ambiente é totalmente redundante em servidores, ou seja, não há pontos únicos de falha em nenhuma etapa do processo de estabelecimento de chamadas VoIP.

O ambiente como um todo foi dividido em quatro camadas para facilitar o entendimento e organização: camada DB, camada Asterisk, camada DNS e camada Cliente; detalhadas a seguir.

8.2.1 Camada BD

É onde está o cluster de banco de dados. Essa camada pode operar de dois modos:

- Os dois servidores ativos: dessa forma o sistema de telefonia irá consultar o DNS implementando uma das formas de balanceamento de carga citadas no capítulo 5 para resolver um determinado nome para o endereço IP de um dos dois servidores do cluster. Caso seja escolhido o balanceamento RR os nós deverão ter IPs virtuais sendo gerenciados pelo heartbeat, pois essa técnica não implementa controle de falhas e caso um falhe o outro servidor deverá assumir o endereço e responder pelos dois até que o servidor danificado volte à ativa. Caso contrário não é necessário executar o software heartbeat já que o DNS SRV consegue fazer o gerenciamento de falhas e caso um dos IPs não responda ele irá encaminhar a solicitação para outro endereço.
- Um servidor ativo e outro redundante: operando dessa forma os servidores de telefonia não irão precisar consultar necessariamente o DNS para acessar o banco de dados, eles podem acessar diretamente o IP virtual gerenciado pelo aplicativo heartbeat que está sendo executado entre os dois nós.

Em ambos os casos o banco de dados deverá ser configurado para que haja replicação mestre/mestre dos dados. No primeiro caso pois não se sabe qual dos dois servidores irá responder à solicitação vinda do sistema de telefonia. No segundo devido à recuperação após o servidor primário ficar inativo ao retorna

esse deve ter seus dados sincronizados com o servidor que assumiu temporariamente a operação.

O banco de dados irá armazenar, entre outras, as informações das contas que serão usadas pelos dispositivos IP e, depois de registrados, qual a localização completa do dispositivo para ser localizado.

8.2.2 Camada Asterisk

Nessa camada ficam os dois servidores responsáveis pelos recursos de telefonia do ambiente. Neles é instalado somente o aplicativo Asterisk para que a carga recebida por esse serviço não sofra influência de outros serviços paralelos.

Os servidores têm configurações idênticas, ou seja, não importa qual dos servidores da camada está sendo acessado pelo cliente que todas as funcionalidades serão gerenciadas da mesma forma.

Toda solicitação de registro que chegar aos servidores de telefonia devem ser encontradas no banco de dados. Ao tentar fazer uma chamada para um dispositivo registrado na plataforma o Asterisk consulta qual a URI do dispositivo que está tentando atingir.

A URI fornecida segue a seguinte sintaxe:

<protocolo>:<ramal>@<endereço IP>:<porta>

Ao fazer a discagem, portanto, o Asterisk irá fazê-la para a URI completa do dispositivo.

Como as informações dos ramais encontram-se na camada BD, as demais configurações necessárias são replicadas entre os servidores e os endereços dos servidores são resolvidos na camada DNS, o ambiente apresenta uma grande escalabilidade. A cada servidor adicionado basta fazer a inclusão do novo endereço IP na camada DNS e configurar a replicação dos dados dos servidores para o recém adicionado.

8.2.3 Camada DNS

O DNS é implementado na estrutura para realizar o balanceamento de carga entre os servidores da camada Asterisk e BD se for o caso. Os servidores dessa camada são replicados de forma idêntica e no modo mestre/escravo, ou seja, somente um servidor resolve os nomes, quando o servidor mestre ficar inativo o escravo assume enquanto o mestre não voltar a operar normalmente.

O bind9 foi o aplicativo de resolução de nomes selecionado para ser usado nessa camada, pois ele disponibiliza os recursos de Round Robin e SRV. O administrador pode escolher qual modo pode implementar de acordo com a necessidade de seu ambiente.

8.2.4 Camada Cliente

Onde se localizam os dispositivos IP como ATAs, softphones, telefones IP e gateways. Esses dispositivos serão registrados no nome resolvido pela camada DNS.

Caso seja o caso de terminar chamadas pela rede pública de telefonia, nessa camada são colocados gateways de E1. Esse equipamento recebe os links E1 e registra-se no Asterisk como um ramal normal, porém no plano de discagem ao realizar uma chamada que utilizará esse equipamento é passada a identificação da porta que será usada e o número que deseja-se enviar para a PSTN.

8.3 Ambiente com localização de extensões via DUNDi

A estrutura física desse ambiente pode ser tanto a do capítulo 8.1 como a do 8.2. No entanto os ramais não serão buscados no banco de dados, mesmo as informações de registro estando lá será utilizado o software DUNDi para localizar

onde determinada extensão está disponível. Abaixo está a ilustração de como é o ambiente em três camadas utilizando o DUNDi.

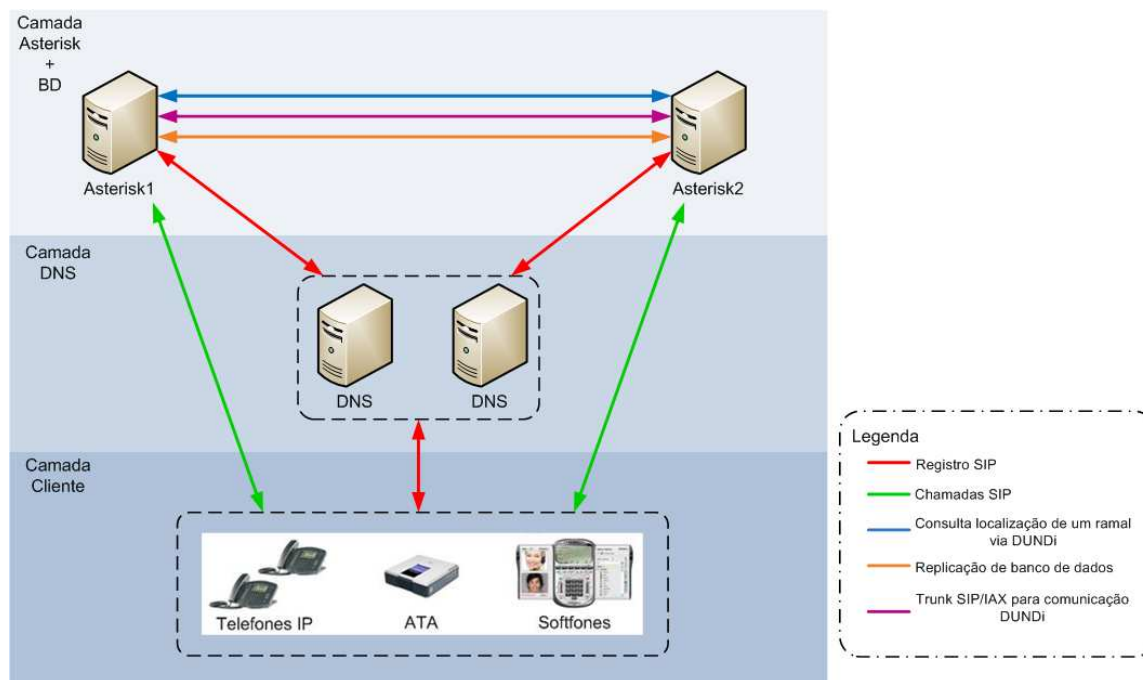


Figura 17: Ambiente com DUNDi

A vantagem dessa estrutura é que deixam de ser feitas consultas no banco de dados que causam um processamento adicional nos servidores. No entanto é criado um tráfego na rede que antes não existia: o das consultas das extensões e do trunk via protocolo SIP ou IAX entre os servidores que farão parte da comunicação quando o DUNDi for utilizado.. Outro fator que passa a ocorrer é que as chamadas feitas entre extensões que estão em servidores diferentes são manipuladas pelos dois nós envolvidos e não mais diretamente entre os pontos, aumentando o número de chamadas simultâneas em cada um dos nós da camada Asterisk + BD.

Quanto à escalabilidade não há perda, pois a cada nó adicionado basta que sejam feitas as mesmas configurações necessárias nos outros ambientes e adicionar a configuração do novo nó a pelo menos um dos servidores Asterisk.

8.4 Ambiente experimental

Foi testada em laboratório uma variante do ambiente acima descrito. A principal diferença desse para o ambiente apresentado acima consiste em fundir as camadas BD e Asterisk, ou seja, em todos os servidores Asterisk existe uma instância do banco de dados que ainda segue o modelo de replicação mestre/mestre para que apesar de distribuídos o banco de dados em cada um dos servidores tenha a mesma informação.

A principal vantagem desse ambiente em relação ao ambiente ideal é que toda consulta SQL feita pelos servidores Asterisk no banco de dados é feita localmente, podendo ser mais ágil e menos custosa caso essa tivesse de ser feita pela rede.

No entanto a carga aceita pelos servidores de telefonia seria reduzida, já que o banco de dados é mais um aplicativo rodando no servidor, comprometendo parte de sua performance.

A figura abaixo ilustra o ambiente que foi implementado e testado.

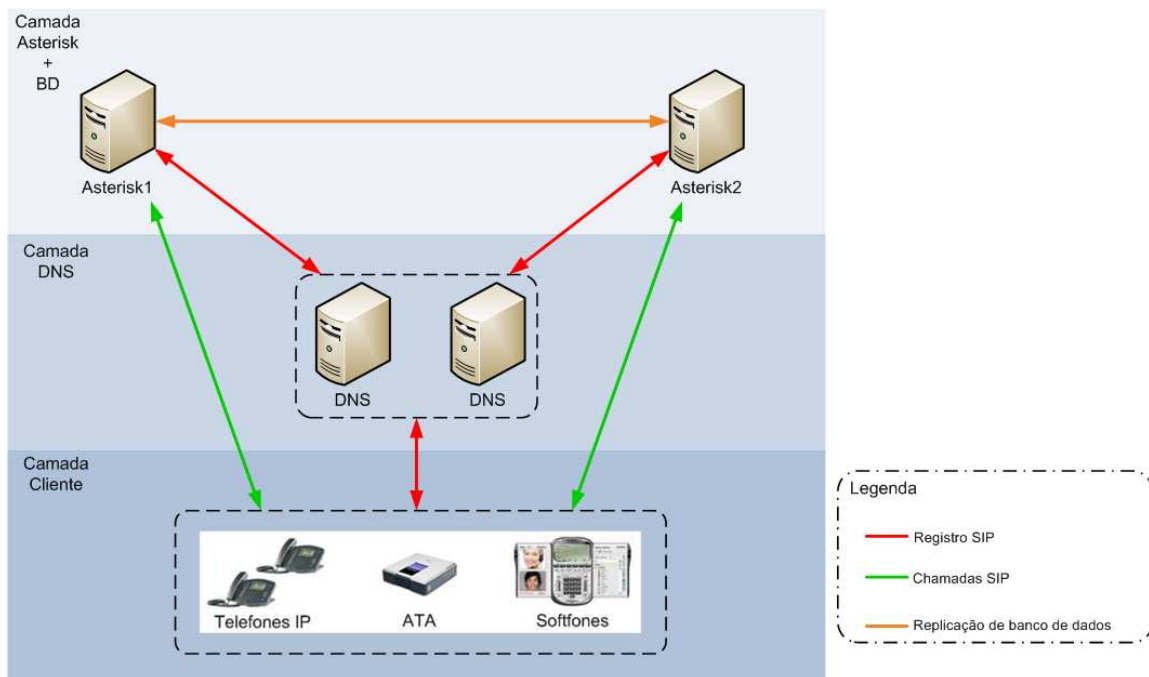


Figura 18: Ambiente experimental

Apesar desse ambiente não separar o banco de dados em um servidor dedicado para essa função, a solução se apresentou estável e facilmente escalável.

Na camada DNS foi implementado o DNS Round Robin pois existem poucos clientes de telefonia IP que suportam o DNS SRV. No entanto, como visto no capítulo 5 o DNS Round Robin não implementa qualquer estratégia de controle de falhas. Dessa forma caso um dos servidores caísse metade das requisições enviadas para a camada Asterisk seriam descartadas.

Na camada DNS não foi implementada redundância devido à falta de equipamentos para tal, no entanto a implementação desse serviço é trivial e não deve apresentar qualquer dificuldade caso seja necessário implementar em um ambiente de produção.

Para realizar controle de falhas entre os servidores o heartbeat implementado entre os servidores da camada Asterisk define um IP virtual em cada um dos nós. Dessa forma quando um dos servidores dessa camada falhar o outro nó assume o IP virtual desse e mantém o seu, ou seja, o DNS continua resolvendo o nome para requisições para os dois endereços porém ambos estão em somente um servidor.

9. Testes realizados no Ambiente Experimental

No ambiente citado no capítulo 8.4 foram realizados testes de alta-disponibilidade e de balanceamento de carga para validar se as configurações se comportam conforme esperado em projeto.

9.1 Testes de alta-disponibilidade

Para testar a redundância dos nós da camada Asterisk do ambiente implementado foram registrados dois softphones, um em cada servidor através do IP virtual de cada um deles e realizada uma chamada para comprovar o sucesso do registro.

Então desligou-se um dos servidores simulando uma falha de hardware ou rede no nó. Notou-se que o servidor que continuou ativo assumiu com sucesso o endereço virtual do nó que falhou. Como o Asterisk mantém em uma base interna e local a tabela de ramais registrados e seu estado o softphone que estava registrado no servidor que teve a falha teve de passar por novo processo de registro, caso contrário o aplicativo gerou um erro indicando que o servidor não pôde ser localizado. Após o novo registro as chamadas puderam ser completadas normalmente.

Após algum tempo o servidor desligado foi reativado e, também com sucesso, obteve seu IP virtual e o removeu do outro servidor. O softphone que inicialmente estava registrado nesse servidor permaneceu registrado e funcional no outro devido à tabela do servidor que permaneceu ativo não ter sofrido qualquer influência com o retorno do outro nó à camada. Ao cancelar o registro e registrar novamente o softphone esse procurou se autenticar no servidor que recém foi reiniciado, a autenticação e chamadas a partir desse momento foram bem sucedidas.

9.2 Testes de balanceamento de carga

Os testes processados no quesito balanceamento de carga foram feitos com a estratégia DNS Round Robin, conforme citado no capítulo 8.4. Foram feitas requisições de registro alternadamente entre dois softphones para verificar se elas seriam distribuídas igualmente entre os nós.

Após se fazer dez requisições obteve-se o resultado de 50% dos registros sendo processados por cada um dos servidores. Porém, não foram processadas completamente de forma alternada como era esperado. As requisições 3 e 4 foram processadas consecutivamente pelo mesmo nó assim como as requisições 7 e 8.

9.3 Conclusão sobre os testes

Os testes realizados comprovaram que implementar um ambiente VoIP com alta-disponibilidade e balanceamento de carga é viável, de fácil configuração e administração.

A única ressalva é quanto ao procedimento de registrar novamente os softphones após a falha e reativação dos nós. Esse procedimento pode diminuir a transparência no momento de falhas de um dos servidores da camada Asterisk. Exigindo talvez até mesmo a intervenção do administrador do ambiente para auxiliar usuários que tenham dificuldade ou disponibilidade para efetuar esse processo.

Apesar disso, se forem utilizados equipamentos que suportem DNS SRV esse problema deixa de existir.

10. Conclusão e Trabalhos Futuros

10.1 Conclusão

Com a abrupta expansão do mercado de voz sobre IP o software Asterisk se destaca como uma das soluções com maior adoção atualmente por ser de baixo custo, facilmente configurável e gerenciável.

Devido a esse crescimento na utilização ambientes cada vez maiores começam a surgir onde o software pode ser utilizado, desde que se faça um bom planejamento da arquitetura que será aplicada, pois essa deve ficar operante o maior tempo possível e com qualidade nas operações que irá realizar.

Existe grande dificuldade de se encontrar uma fonte única que reúna várias informações de como implementar uma estrutura que abranja todas essas premissas de alta-disponibilidade e balanceamento de carga entre Asterisks.

As arquiteturas e testes aqui apresentados servem como uma referência para os administradores de sistemas que desejam iniciar ou expandir seu sistema de telefonia utilizando VoIP somente com aplicativos open source.

10.2 Trabalhos Futuros

- Ampliar o conceito de clusterização dos servidores Asterisk para que esses trabalhem de uma forma mais unificada.

- Elaborar um instalador que facilite a instalação inicial e adição de novos nós ao ambiente.

- Desenvolver uma interface de administração do ambiente.

Anexos

Anexo 1 – Implementando uma Solução VoIP Baseada em Asterisk com Alta-Disponibilidade e Balanceamento de Carga

Implementado uma Solução VoIP Baseada em Asterisk com Alta-Disponibilidade e Balanceamento de Carga

Fábio Danieleski Gross

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina(UFSC)
Florianópolis, SC – Brasil.

fdgross@gmail.com

Abstract. *The Asterisk is building every day as a leading IP PBX market, primarily due to the low cost of implementation as open source and its flexibility.*

But there are several limitations that prevent its use in environments where telephone service is essential. Among these we can highlight the need for high-availability and limitations of hardware. However, nowadays it is difficult to find documentation for projects that describe how these environments.

This study was designed to help support and implement a project to an environment with the characteristics mentioned above, using only free software tools.

Resumo. *O Asterisk vem consolidando a cada dia como um dos principais PABX IP do mercado devido, principalmente, ao baixo custo de implementação por ser de código aberto e sua flexibilidade.*

Porém existem diversas limitações que impedem o seu uso em ambientes onde o serviço de telefonia é fundamental. Entre essas podemos destacar a necessidade de alta-disponibilidade e limitações de hardware. No entanto, atualmente, é difícil encontrar documentações que descrevam projetos para ambientes como esses.

Esse trabalho foi desenvolvido para ajudar a fundamentar e implementar um projeto para um ambiente com as características acima citadas, utilizando apenas ferramentas de software livre.

11. Introdução

O mercado abraçou fortemente o conceito de VoIP e hoje vem investindo muito nessa idéia. As corporações estão migrando boa parte de seus sistemas para a internet e agora se observa que é possível, e mais, é muito vantajoso migrar também o serviço de telefonia para trafegar sobre a rede de computadores que já abrange a maior parte das empresas e uma boa parte das residências no mundo. Deixa-se de ter dois serviços totalmente independentes aumentando a necessidade de dois tipos totalmente distintos de estrutura, mão de obra, etc. para ter uma única solução toda baseada na tecnologia IP que vem se difundindo em larga escala.

A principal vantagem que conquista a atenção para a maioria dos projetos é a redução dos custos mensais com telefonia, já que as tarifas de operadoras VoIP são, em geral, mais reduzidas em relação às tarifas cobradas pelas operadoras de telefonia tradicional. A convergência dos serviços, conforme comentado acima, também é uma vantagem bastante interessante.

A qualidade das chamadas e a disponibilidade são dois pontos-chave que os administradores dos sistemas VoIP estão sempre atentos, já que esse sistema de comunicação é bem mais sensível em relação à telefonia tradicional. A

qualidade da rede, dos dispositivos, dos servidores, etc. interfere diretamente na qualidade do serviço.

Esse trabalho visa abordar uma solução para dar mais robustez aos sistemas VoIP, trazendo opções de alta-disponibilidade e balanceamento de carga entre servidores utilizando Asterisk e alguns softwares que darão suporte à implantação da arquitetura.

12. VoIP

A sigla VoIP vem de Voice over Internet Protocol (Voz sobre o protocolo IP), que significa a transmissão de voz usando o protocolo IP. Fazendo uso dessa tecnologia passa a existir um serviço especializado em telefonia sendo transmitido integralmente através das redes de computadores ou podendo, também, ser integrado ao sistema analógico de telefonia através de adaptadores que fazem tal conversão.

Existem diversas vantagens que pode-se obter com essa tecnologia, entre elas podemos citar: a convergência, a mobilidade e a redução de custos.

A convergência é a integração ou migração dos serviços de telefonia convencional à rede de dados. Onde se deixa de ter um custo operacional e de infra-estrutura dobrado. Como a estrutura de rede já é fundamental em qualquer ambiente empresarial nada de novo precisa ser feito para implementar VoIP na empresa. No entanto deve-se ter cautela quanto à qualidade dos serviços de rede já que a telefonia é bastante sensível às intempéries do meio.

A mobilidade é possível pois o ramal deixa de estar atrelado a um ponto físico na rede de telefonia e passa a ser relacionado ao colaborador. Dessa forma ele pode se manter operacional de qualquer lugar interno ou externo à estrutura da empresa e ainda assim manter todas as funcionalidades relacionadas ao ramal.

A redução de custos vem como consequência dos dois fatores acima já que podem ser reduzidos os custos com ligações para os telefones dos funcionários, pois podem receber ou originar chamadas através de seus ramais. A diminuição dos custos é ainda mais abrupta quando se trata da comunicação entre matriz e filial ou entre filiais.

12.1 Protocolos VoIP:

Os protocolos são responsáveis pelo estabelecimento, manutenção e finalização das chamadas. Em VoIP os protocolos mais utilizados são os seguintes:

12.1.1 SIP(Session Initiation Protocol):

O protocolo SIP encontra-se na sua segunda versão que está definida na RFC 3261 publicada no ano de 2002. Esse protocolo encaixa-se na camada de aplicação do modelo de referência OSI e é responsável por estabelecer, alterar e finalizar sessões multimídia, entre elas, as chamadas VoIP.

Em sua estrutura o protocolo SIP se parece com o protocolo HTTP, sendo também um protocolo baseado em texto e cliente/servidor, ou seja, implementa métodos de requisição e resposta na comunicação. Pode transportar qualquer carga independentemente, assim, de protocolos da camada de transporte.

Por ser um protocolo bastante difundido nos ambientes multimídia, o protocolo SIP também é amplamente utilizado em soluções de voz sobre IP.

12.1.2 RTP (Real-time Transport Protocol):

Em ambientes de voz sobre IP o RTP é utilizado em conjunto com o protocolo SIP para fazer o transporte do áudio das chamadas já que o segundo não implementa esse tipo de transporte.

12.1.3 IAX (Inter-Asterisk Exchange Protocol):

O protocolo IAX foi desenvolvido pela Digium (empresa que detém os direitos autorais do Asterisk e que iniciou seu desenvolvimento) com o principal objetivo de interligar servidores Asterisk. Atualmente encontra-se na versão 2 e é definido na RFC 5456.

12.1.4 H.323:

O H.323 é um dos protocolos mais maduros, robustos e utilizados nos equipamentos que utilizam VoIP por ser um dos mais antigos e definido especificamente com a finalidade de comunicações multimídia.

12.2 CODECs:

A voz humana é transmitida em ondas senoidais através do meio. No entanto essas ondas têm de ser transformadas de sinal analógico para digital para que a comunicação na rede de dados seja possível. Esse processo de transformação é chamado digitalização do áudio.

No processo de digitalização o sinal contínuo do áudio é transformado em valores discretos por meio de amostragens analisadas com uma frequência pré-determinada. Para fazer a codificação e decodificação dos sinais fazendo compressão para que ocorra um melhor aproveitamento da banda existem os CODECs (enCOders/DECoders).

13. Asterisk

Asterisk é um PBX IP que roda como um serviço em um servidor Linux, Windows, Mac ou BSD e vem crescendo rapidamente em desenvolvimento e utilização, sendo atualmente o PBX IP mais utilizado no mundo.

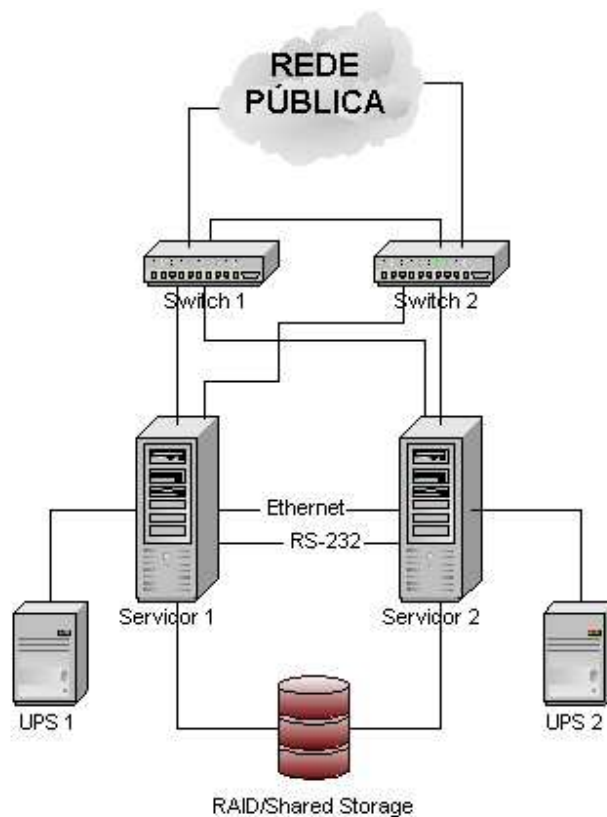
O Asterisk tem todas as funcionalidades de qualquer PBX do mercado, com a vantagem de ser de código aberto, podendo, assim, ser utilizado e desenvolvido sem qualquer custo ou necessidade de licença.

Ele é capaz de trazer grande interoperabilidade entre plataformas, pois faz a conversão entre vários CODECs e protocolos do mercado. Elimina a necessidade de grande parte dos módulos externos que disponibilizam recursos avançados fundamentais na maioria das centrais telefônica, como: música em espera, correio de voz e URA (Unidade de Resposta Audível).

14. Alta-Disponibilidade de Servidores

Os sistemas computacionais são altamente importantes atualmente, no entanto esses sistemas são relativamente frágeis pois dependem de diversos fatores internos e externos que comprometem a sua disponibilidade.

Para que não se perca os benefícios dos sistemas informatizados quando ocorrerem tais faltas no ambiente, principalmente quando o sistema for de operação crítica, costuma-se usar o conceito de redundância. Esse conceito propõe que sejam eliminados o maior número de pontos de falha possíveis. Assim quanto menor for o número de pontos únicos de falha maior será o nível de disponibilidade que se pode obter com o sistema. A figura a seguir é um exemplo clássico de alta-disponibilidade:



A aplicação da suíte Linux HA que faz o monitoramento de disponibilidade entre servidores é o software heartbeat.

Nesse aplicativo o servidor identificado como principal irá executar os serviços e terá um endereço IP virtual junto à sua interface de rede conectada à LAN (Local Area Network). Os clientes que forem acessar o sistema deverão fazê-lo através do IP virtual. O heartbeat manda sinais periódicos de keepalive entre um servidor e outro. Quando o servidor configurado como primário para aquele IP virtual ficar um tempo pré-configurado sem responder a esses sinais ele será considerado como indisponível pelo servidor redundante. A partir desse momento o servidor redundante ou secundário irá assumir o IP virtual e os serviços que até então eram executados no servidor primário ou principal.

Uma opção para quando não é necessário replicar uma partição inteira, mas sim somente alguns arquivos entre os servidores replicados é aplicativo rsync. Esse aplicativo tem o funcionamento semelhante ao scp, porém ao ser executado ele não irá copiar todos os arquivos da origem para o destino incondicionalmente, mas sim fará uma comparação pela data de alteração dos arquivos da origem e fará a cópia somente dos que tiverem diferença entre os dois pontos. Caso o arquivo não exista no destino ele fará a cópia do arquivo normalmente.

15. Balanceamento de Carga

Uma arquitetura baseada em balanceamento de carga consiste em dividir as requisições feitas entre os servidores para evitar sobrecarga de um dos nós e aproveitar da forma mais igualitária possível os recursos disponíveis.

O balanceamento de carga é feito por uma aplicação que encaminha as requisições para os servidores de acordo com a estratégia que for configurada. Duas aplicações baseadas em DNS (Domain Name System) são bastante utilizadas em estruturas que demandam balanceamento de carga. São elas:

DNS Round Robin (DNS RR): Nessa técnica de resolução de nomes o serviço deixa de resolver um nome para somente um endereço IP e passa a fazê-lo para uma lista de endereços que disponibilizam o mesmo serviço. Essa distribuição é feita de forma circular entre os endereços IPs configurados sendo respondido pelo endereço que estiver no topo da lista.

DNS SRV: As principais diferenças da técnica DNS SRV para a DNS Round Robin supracitada, são: pode-se configurar prioridades e pesos para qual(is) servidor(es) irá(ão) responder às requisições preferencialmente; e os clientes caso detectem a falha na entregue na requisição a técnica permite que sejam atingidos servidores que tenham uma prioridade menor.

16. VoIP com alta disponibilidade e balanceamento de carga

A telefonia é um recurso muito utilizado na comunicação das empresas devido à agilidade e maior facilidade que as pessoas têm de se comunicar falando que escrevendo através de e-mails ou sistemas de mensagem instantânea. Além da funcionalidade normal de interligar duas pessoas, foram agregadas várias funcionalidades que trazem ainda mais valor ao sistema telefônico, como URAs e serviços de contato com o cliente. Esse último é o ramo de negócios de algumas empresas bastante utilizado atualmente.

Com o VoIP sendo largamente utilizado, o sistema deve ser robusto e ficar o menor tempo indisponível possível, já que o sistema de telefonia legado tem como principal benefício a estabilidade, dessa forma um projeto de voz sobre IP não pode deixar a desejar nesse quesito mas sim aliá-lo aos demais pontos fortes da arquitetura. Devido a esses fatores surge a necessidade de se implementar técnicas de alta disponibilidade no ambiente.

Além de se eliminar os pontos únicos de falha podem-se aproveitar os equipamentos redundantes para executarem tarefas mesmo quando estão em segundo plano na estrutura. Para isso se implementa os conceitos de balanceamento de carga.

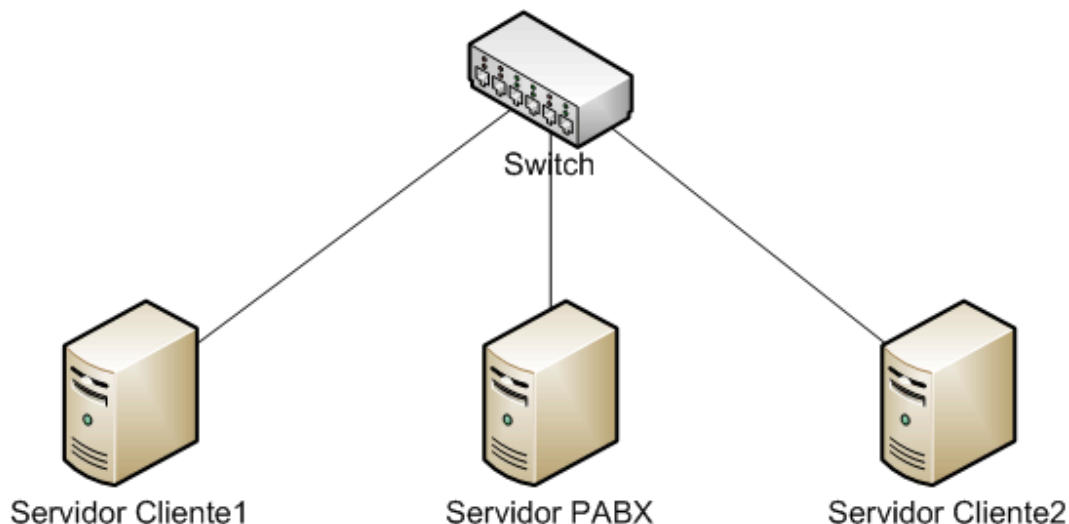
Esse conceito pode, também, ser utilizado quando o hardware que está à disposição individualmente não é suficiente para realizar todas as tarefas necessárias exigidas pelo projeto.

17. Teste de carga em um servidor Asterisk

Para dar embasamento e identificarmos qual a carga real de chamadas simultâneas que um servidor comercial rodando Asterisk pode processar, elaboramos um plano de testes que irá focar no que existe de mais crítico no tocante a uso de CPU e memória durante a execução de muitas chamadas: fazer

a passagem da mídia seja feita através do servidor e a conversão entre CODECS seja feita pelo Asterisk.

Os testes forma realizados utilizando a seguinte estrutura física:



17.1 Resultado dos testes

Com os testes realizados, constatou-se que o servidor começa a perder em estabilidade com o passar do tempo quando submetido a uma grande carga de chamadas. Conseqüentemente a qualidade dos serviços de telefonia de voz sobre IP tende a piorar proporcionalmente.

Para o hardware do Servidor PABX percebeu-se que quando usado somente com o serviço do Asterisk rodando, o volume de ligações não poderia ultrapassar de duzentas chamadas. Pois, apesar dos testes mostrarem bons resultados ainda em 250 e 300 chamadas, não foi considerado que estejam sendo utilizados outros serviços usuais e atividades também corriqueiras como consulta de ramais e escritas de CDR em bancos de dados; requisições de registro de dispositivos IP ou ainda aplicações do próprio aplicativo, como correio de voz e

gravações por exemplo, que exigem algum processamento extra e aumentam a utilização da CPU e memória do servidor.

Mesmo que esses serviços individualmente não apresentem grande perturbação nas medições feitas, rodando em conjunto e sob forte carga – como foi o caso do Asterisk – com certeza influem no processamento e qualidade de ligações.

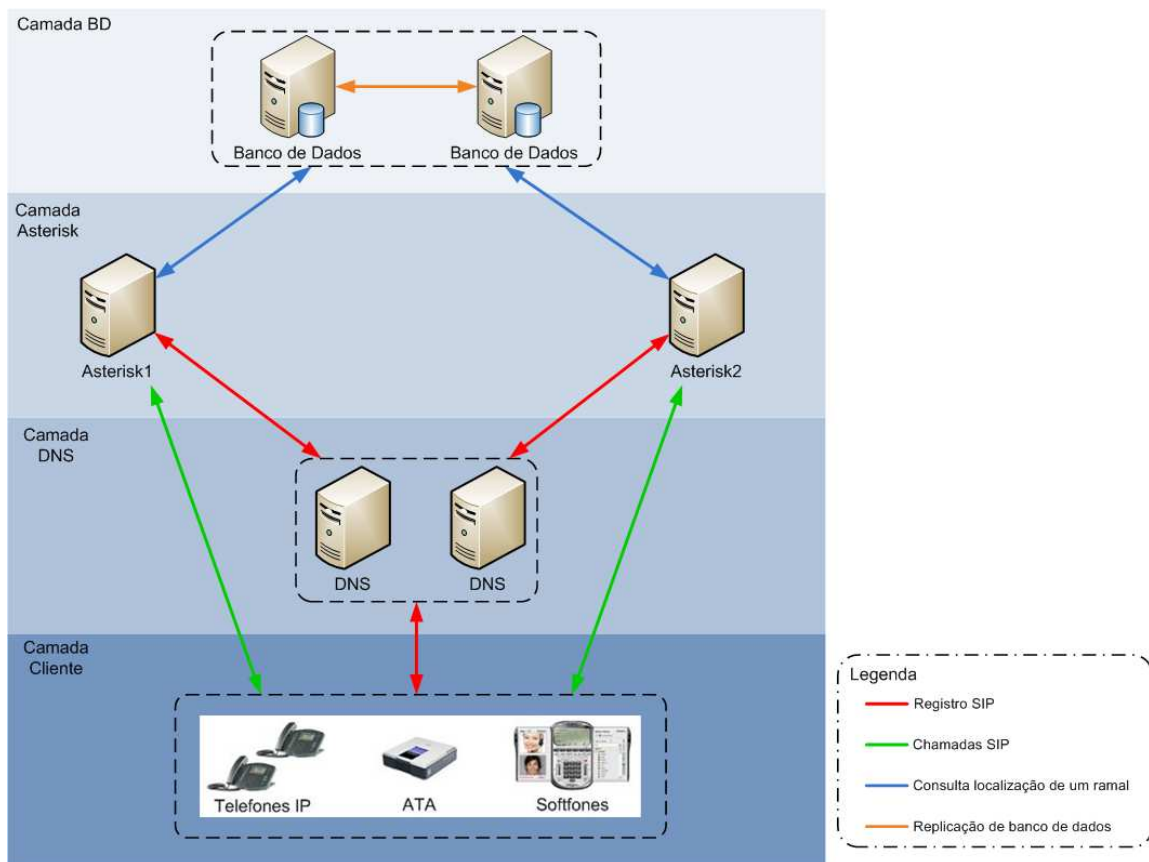
18. Soluções de alta-disponibilidade e balanceamento de carga para VoIP

Para como foi observado nos resultados de carga sobre um servidor Asterisk, ficou claro que esse serviço consegue gerenciar com qualidade um número bastante grande de chamadas. Porém, uma grande empresa que tenha diversos ramais utilizando com grande frequência os recursos de telefonia não pode aceitar que apenas um servidor seja o ponto central da sua arquitetura. Caso isso ocorra, o sistema de telefonia estará muito frágil e qualquer falha que venha a ocorrer nesse ponto central irá comprometer totalmente o funcionamento do sistema telefônico.

Para evitar isso foram propostas algumas arquiteturas que visam eliminar o ponto único e central de falha fazendo redundância do servidor Asterisk utilizando apenas aplicativos open source.

18.1 Ambiente ideal

A figura abaixo ilustra a arquitetura considerada ideal para ser implementada em um ambiente de VoIP em alta disponibilidade e que ocorra balanceamento de carga entre os servidores de telefonia.



Esse ambiente é totalmente redundante em servidores, ou seja, não há pontos únicos de falha em nenhuma etapa do processo de estabelecimento de chamadas VoIP.

O ambiente como um todo foi dividido em quatro camadas para facilitar o entendimento e organização: camada DB, camada Asterisk, camada DNS e camada Cliente.

18.1.1 Camada BD

É onde está o cluster de banco de dados. Nessa camada os servidores podem operar de forma ativo/passivo, ou seja, somente com um servidor

respondendo às requisições; ou na forma ativo/ativo com mais de um servidor respondendo às requisições e seus nós atualizando-se entre si.

18.1.2 Camada Asterisk

Nessa camada ficam os dois servidores responsáveis pelos recursos de telefonia do ambiente.

Os servidores têm configurações idênticas, ou seja, não importa qual dos servidores da camada está sendo acessado pelo cliente que todas as funcionalidades serão gerenciadas da mesma forma.

18.1.3 Camada DNS

O DNS é implementado na estrutura para realizar o balanceamento de carga entre os servidores da camada Asterisk e BD se for o caso. Os servidores dessa camada são replicados de forma idêntica e no modo mestre/escravo, ou seja, somente um servidor resolve os nomes, quando o servidor mestre ficar inativo o escravo assume enquanto o mestre não voltar a operar normalmente.

18.1.4 Camada Cliente

Onde se localizam os dispositivos IP como ATAs, softphones, telefones IP e gateways. Esses dispositivos serão registrados no nome resolvido pela camada DNS.

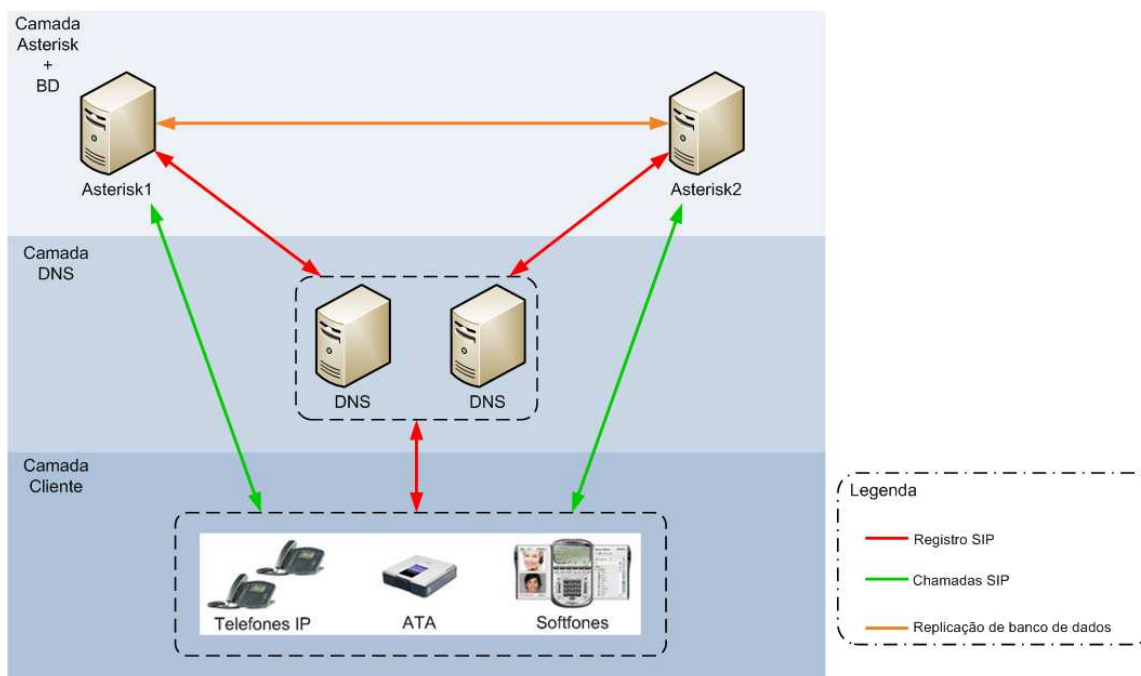
18.2 Ambiente experimental

Foi testada em laboratório uma variante do ambiente acima descrito. A principal diferença desse para o ambiente apresentado acima consiste em fundir as camadas BD e Asterisk, ou seja, em todos os servidores Asterisk existe uma instância do banco de dados que ainda segue o modelo de replicação mestre/mestre para que apesar de distribuídos o banco de dados em cada um dos servidores tenha a mesma informação.

A principal vantagem desse ambiente em relação ao ambiente ideal é que toda consulta SQL feita pelos servidores Asterisk no banco de dados é feita localmente, podendo ser mais ágil e menos custosa caso essa tivesse de ser feita pela rede.

No entanto a carga aceita pelos servidores de telefonia seria reduzida, já que o banco de dados é mais um aplicativo rodando no servidor, comprometendo parte de sua performance.

A figura abaixo ilustra o ambiente que foi implementado e testado.



Apesar desse ambiente não separar o banco de dados em um servidor dedicado para essa função, a solução se apresentou estável e facilmente escalável.

Na camada DNS foi implementado o DNS Round Robin pois existem poucos clientes de telefonia IP que suportam o DNS SRV.

Para realizar controle de falhas entre os servidores o heartbeat implementado entre os servidores da camada Asterisk define um IP virtual em cada um dos nós. Dessa forma quando um dos servidores dessa camada falhar o outro nó assume o IP virtual desse e mantém o seu, ou seja, o DNS continua resolvendo o nome para requisições para os dois endereços porém ambos estão em somente um servidor.

19. Testes realizados no Ambiente Experimental

No ambiente citado no capítulo 8.4 foram realizados testes de alta-disponibilidade e de balanceamento de carga para validar se as configurações se comportam conforme esperado em projeto.

19.1 Testes de alta-disponibilidade

Para testar a redundância dos nós da camada Asterisk do ambiente implementado foram registrados dois softphones, um em cada servidor através do IP virtual de cada um deles e realizada uma chamada para comprovar o sucesso do registro.

Então desligou-se um dos servidores simulando uma falha de hardware ou rede no nó. Notou-se que o servidor que continuou ativo assumiu com sucesso o endereço virtual do nó que falhou. Como o Asterisk mantém em uma base interna e local a tabela de ramais registrados e seu estado o softphone que estava registrado no servidor que teve a falha teve de passar por novo processo de registro, caso contrário o aplicativo gerou um erro indicando que o servidor não pôde ser localizado. Após o novo registro as chamadas puderam ser completadas normalmente.

Após algum tempo o servidor desligado foi reativado e, também com sucesso, obteve seu IP virtual e o removeu do outro servidor.

19.2 Testes de balanceamento de carga

Os testes processados no quesito balanceamento de carga foram feitos com a estratégia DNS Round Robin. Foram feitas requisições de registro alternadamente entre dois softphones para verificar se elas seriam distribuídas igualmente entre os nós.

Após se fazer dez requisições obteve-se o resultado de 50% dos registros sendo processados por cada um dos servidores.

19.3 Conclusão sobre os testes

Os testes realizados comprovaram que implementar um ambiente VoIP com alta-disponibilidade e balanceamento de carga é viável, de fácil configuração e administração.

A única ressalva é quanto ao procedimento de registrar novamente os softphones após a falha e reativação dos nós. Esse procedimento pode diminuir a transparência no momento de falhas de um dos servidores da camada Asterisk.

20. Conclusão

Com a abrupta expansão do mercado de voz sobre IP o software Asterisk se destaca como uma das soluções com maior adoção atualmente por ser de baixo custo, facilmente configurável e gerenciável.

Devido a esse crescimento na utilização ambientes cada vez maiores começam a surgir onde o software pode ser utilizado, desde que se faça um bom planejamento da arquitetura que será aplicada, pois essa deve ficar operante o maior tempo possível e com qualidade nas operações que irá realizar.

As arquiteturas e testes aqui apresentados servem como uma referência para os administradores de sistemas que desejam iniciar ou expandir seu sistema de telefonia utilizando VoIP somente com aplicativos open source.

Anexo 2 – Configuração do plano de discagem para busca de ramais em um Banco de Dados

Foi feita a seguinte regra no plano de discagem para ser aplicada no plano de discagem da camada Asterisk do ambiente, dessa forma será feita a localização dos dispositivos em uma base única e sem a necessidade de se estabelecer um SIP ou IAX trunk entre os servidores, reduzindo o volume de informação a ser processada por cada servidor.

```
exten => ${FAIXARAMAIS},1,MySQL(Connect connid <host do bd> <user
do bd> <senha do bd> <nome do dB>)
exten => ${FAIXARAMAIS},2,MySQL(Query resultid ${connid} SELECT\
fullcontact\ FROM\ sip_buddies\ WHERE\ name=${EXTEN})
exten => ${FAIXARAMAIS},3,MySQL(Fetch fetchid ${resutid} fullcontact)
exten => ${FAIXARAMAIS},4,MySQL(Clear ${resultid})
exten => ${FAIXARAMAIS},5,MySQL(Disconnect ${connid})
exten => ${FAIXARAMAIS},6,Dial(SIP/${fullcontact:4},60,wWtT)
exten => ${FAIXARAMAIS},7,Hangup
```

Onde:

- <host do bd>: o nome ou endereço IP do servidor de banco de dados;
- <user do bd>: usuário que irá conectar no banco de dados;
- <senha do bd>: senha do usuário;
- <nome do bd>: nome do banco de dados onde está as tabelas de configurações dos ramais;
- sip_buddies: tabela que contém as informações dos ramais.

Conclusão

[ASTERISK EXPERTS, 2008]. **Asterisk já é o PBX mais usado no mundo inteiro.** 15 de maio de 2008. Disponível em: <<http://www.asteriskexperts.com.br/content/view/248/1/>>. Acesso em: 14 set. 08.

[SMITH, MEGGELEN e MADSEN, 2005] SMITH, Jared; MEGGELN, Jim Van; MADSEN, Leif. **Asterisk: O Futuro da Telefonia.** Rio de Janeiro: Altos Books, 2005.

[SOUZA e CUNHA, 2004] Souza, Rodrigo Jean de; Wagner. **Ferramenta para Comunicação VoIP Usando o Padrão H.323 em Redes com Servidores NAT.** Florianópolis: UFSC, 2004. 71p. Monografia (Departamento de Informática e Estatística) – Universidade Federal de Santa Catarina, Florianópolis, 2004.

[ASTERISK, 2008]. **Asterisk Architecture.** 2008. Disponível em: <<http://www.asterisk.org/support/architecture>>. Acesso em: 11 nov. 08.

[LUNG, 2008]. LUNG, Lau Cheuk. **Aula 4 – Tolerância a Faltas.** 2008. Disponível em: < <http://www.inf.ufsc.br/~lau.lung/INE5625/SD-Aula4-ToleranciaAFaltas.pdf>>. Acesso em: 12 nov. 08.

[WIKIPEDIA, 2008]. **Sistema de alta disponibilidade.** 2008. Disponível em: <http://pt.wikipedia.org/wiki/Sistema_de_alta_disponibilidade>. Acesso em: 17 nov. 08.

[VOIP-INFO, 2008]. **DUNDi.** 2008. Disponível em: < <http://www.voip-info.org/wiki-DUNDi>>. Acesso em: 22 nov. 08.

[RFC 1889]. **RTP: A Transport Protocol for Realtime Applications.** SCHULZRINNE, H.; et al. 01/1996. RFC 1889.