

Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM

UMA ANÁLISE COMPARATIVA ENTRE SERVIDORES PROXY DE ALTO DESEMPENHO.

Aluno: Pedro Henrique Alves Ribeiro
Matricula: 06.1.4154

Orientador: Reinaldo Silva Fortes

Ouro Preto
9 de novembro de 2012

Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM

UMA ANÁLISE COMPARATIVA ENTRE SERVIDORES PROXY DE ALTO DESEMPENHO.

Proposta de monografia apresentada ao curso de Bacharelado em Ciência da Computação, Universidade Federal de Ouro Preto, como requisito parcial para a conclusão da disciplina Monografia I (BCC390).

Aluno: Pedro Henrique Alves Ribeiro
Matricula: 06.1.4154

Orientador: Reinaldo Silva Fortes

Ouro Preto
9 de novembro de 2012

Resumo

A utilização da Internet cresce a cada dia. Com isso, a quantidade e a natureza do conteúdo acessado exigem uma velocidade de transferência de dados cada vez maior, principalmente em redes internas que compartilham o acesso à Internet. Esse avanço provoca nessas redes, se não bem administradas, problemas como navegação lenta, falta de segurança e acesso a conteúdo impróprio. Uma das formas de amenizar esses problemas é incluir um servidor na rede antes do acesso à Web, isso pode acontecer através da implantação de um *Proxy*. Esse documento apresenta como proposta de trabalho de conclusão de curso a implantação e análise de performance de três soluções que implementam um servidor *Proxy* de alto desempenho, são elas o *Squid*, *Varnish* e o *ISA Server*. De forma que ao final ele possa contribuir com a comunidade de profissionais de TI em projetos que implantam soluções como essa.

Palavras-chave: proxy, caching, squid, varnish, isa server.

Sumário

1	Introdução	1
2	Justificativa	3
3	Objetivos	4
3.1	Objetivo geral	4
3.2	Objetivos específicos	4
4	Metodologia	5
5	Desenvolvimento	6
5.1	Cenários e Métricas	6
5.2	Estações de Trabalho	6
5.3	Implantação	7
5.3.1	Squid	7
5.3.2	Varnish	8
5.3.3	ISA Server	9
5.3.4	Dificuldades Encontradas	9
6	Considerações Finais e Trabalhos Futuros	11
7	Cronograma de atividades para BCC 391 - Monografia II	12

Lista de Figuras

1 Funcionamento de um servidor *Proxy* 1

Lista de Tabelas

1 Cronograma de Atividades. 12

1 Introdução

Nos últimos anos, a Internet tem evoluído de forma espantosa. Sua aceitação foi muito rápida e são diversas as aplicações desenvolvidas para este que se transformou em um gigantesco meio de acesso à informação e comunicação. Atualmente é um universo crescente de páginas interligadas e aplicações repletas de vídeos, fotos, comércio eletrônico, voz e muito conteúdo interativo.

O número de usuários que se conectam na rede cresce de forma acentuada. Esse crescimento rápido provocou a elevada demanda de requisições de conteúdo e a especialização das aplicações Web, de forma que tem gerado diversos problemas, principalmente em grandes redes corporativas. Em muitos casos, a estrutura dessas redes não está sendo capaz de atender o usuário com eficiência. Segundo *Brian* [2] alguns dos problemas causados podem ser citados: alto tempo nas respostas para o usuário, redução na disponibilidade das informações, utilização incorreta de recursos e alta carga nos servidores.

Um dos grandes desafios desse tema é a minimização do tempo para se recuperar um documento Web, que chamamos de tempo de resposta. Ele depende de determinados fatores, como tipo de conexão, tamanho do documento e localização do mesmo. Para minimizar as consequências deste crescimento, alguns métodos podem ser adotados, como a utilização de um servidor mais rápido, alteração dos meios físicos (cabos, roteadores e switches) de forma que se aumente a taxa de transferência. Contudo, este método, além de não ser economicamente viável, pode não resolver o problema, uma vez que são numerosos os fatores que envolvem uma única transação Web. Alternativas [9] como o cache de páginas utilizando servidores *Proxy* e espelhamento de arquivos têm sido aplicadas para resolver determinadas situações em que a rede não consegue atender todas as requisições de seus usuários de forma eficiente.

Segundo *Tanenbaum* [14], *Proxy* pode ser definido como um tipo de servidor que redireciona as requisições de um usuário por algum serviço fornecido por outro servidor, em uma outra rede. Ele funciona como um filtro, analisando a requisição e definindo se ela pode ser redirecionada ou não.

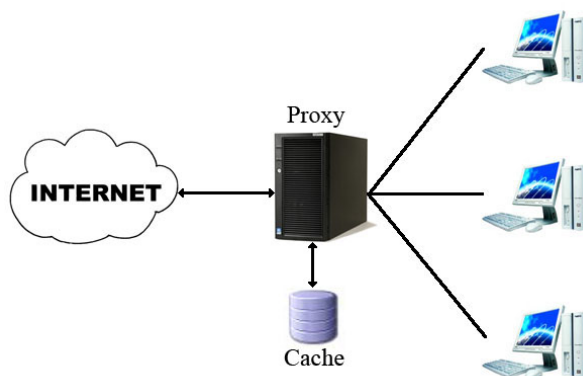


Figura 1: Funcionamento de um servidor *Proxy*

Ao longo do tempo os servidores *Proxy* receberam modificações e passaram a disponibilizar uma série de serviços além de filtrar conteúdo, como o cache de páginas Web e a segurança das informações. A Figura 1 mostra o funcionamento de um *Proxy*, os

computadores da rede (à direita da figura) obtêm acesso à Internet através do servidor *Proxy*. Todas as requisições de conteúdo são encaminhadas a ele, que busca e retorna as repostas, se aprovadas, ao usuário. Elas podem estar no cache, nesse caso o tempo de resposta é menor. O cache é um dos responsáveis por essa redução nas requisições de páginas Web.

Dos anos 90 aos dias atuais muitas soluções utilizando servidores *Proxy* foram desenvolvidas. Dessa forma surgiram alternativas para melhorar o tráfego da rede e minimizar as consequências desse impacto causado pela popularização da Internet. Hoje, existem diversas ferramentas que implementam um servidor *Proxy*, porém apenas algumas conseguem obter um alto desempenho quando implantados em uma rede de grande porte, são elas: *Squid*, *Varnish* e o *ISA Server*.

Com esse trabalho pretende-se contribuir com a uma elaboração de uma análise aprofundada das ferramentas anteriormente citadas. De forma que um profissional de TI possa escolher a solução ideal para seu cenário, uma vez que esses servidores podem apresentar performance diferentes em variados contextos (ou condições). Adicionalmente adquirir conhecimento técnico específico, uma vez que a implantação dessas ferramentas não é algo considerado trivial.

Dessa forma esse documento apresenta uma proposta de um trabalho para a implantação e análise de performance das três soluções de *Proxy* citadas anteriormente sob diferentes métricas e variados cenários.

Nas próximas seções vamos apresentar de forma clara e objetiva as razões que justificam a realização desse trabalho. Em seguida são apresentados os objetivos que pretendemos atingir. Logo após, apresentamos a metodologia que será aplicada no desenvolvimento do mesmo. E por fim, um cronograma das atividades desenvolvidas durante o projeto é apresentado.

2 Justificativa

Um estudo realizado por *Howard e Jansen* [6] analisando a viabilidade de utilizar um servidor *Proxy* geraram alguns questionamentos, como por exemplo, com a web evoluindo de estática para dinâmica, haverá um futuro para esses servidores? Entretanto, um estudo mais recente [8] mostra que, apesar de cerca de 70% do conteúdo requisitado para esse tipo de servidor ser dinâmico, ele ainda consegue apresentar uma melhora significativa no desempenho da rede e continua sendo uma boa opção para reduzir o tempo de resposta as requisições Web e a segurança dos usuários.

Com o sucesso do uso desses servidores, várias ferramentas foram desenvolvidas para implementar um *Proxy*, a mais conhecida e utilizada é o *Squid* [17]. Por estar em desenvolvimento há quase 15 anos, ele detém uma grande base de usuários, além de ser disponível virtualmente em todos os sistemas operacionais *Unix-like* (como *Linux*, *Solaris*, *BSD's*) e foi, há alguns anos, portado para o kernel do *Windows NT*.

Em 2006 foi lançado um *Proxy* de alto desempenho, o *Varnish* [16]. Afirmando ter um design moderno ele foi trabalhado desde o início para obter uma ótima performance no tempo de resposta as requisições Web. A sua última versão (3.0) é projetada para ter melhor desempenho em arquiteturas 64-bits e para usar ao máximo recursos otimizados presentes nos SO's *Unix-like*. Os desenvolvedores do *Varnish* disponibilizam, nessa versão, módulos adicionais que garantem a flexibilidade. Eles alegam que sua performance é superior à do *Squid*.

Outra solução bastante conhecida quando se necessita usar o sistema operacional *Windows* é o *Microsoft Internet Security & Acceleration (ISA) Server* [7], que apesar de não ser apenas um servidor *Proxy*, oferece uma solução completa para fornecer acesso entre a rede corporativa e a Internet. O *ISA Server* é um *gateway* de camada de aplicação e inspeção avançada de pacotes, um servidor de rede privada virtual (*VPN*) e uma solução de cache Web que permite aos clientes maximizarem facilmente os investimentos em TI aumentando a segurança e o desempenho da rede.

Diante de um cenário com muitas soluções e diversas ferramentas que podem ser utilizadas, considera-se importante apresentar um estudo comparativo entre as três soluções aqui citadas, a fim de concluir qual dessas soluções apresentam melhor desempenho em variados cenários que simulam situações reais e frequentes. É importante ressaltar que um trabalho desenvolvido por Lucas Silva [11] analisou dois servidores *Proxy* de alto desempenho, o *Squid* e o *Varnish*. Neste trabalho, apenas duas soluções existentes no mercado foram analisadas, e nenhuma utilizando sistema operacional *Windows*. Além disso, essa análise foi feita somente sobre duas métricas, o tempo de resposta e o número de requisições suportadas. Acreditamos que esta análise não é suficiente para concluir, em uma abordagem ampla de cenários, qual servidor deveremos usar. Nossa proposta se diferencia deste trabalho por incluir um *Proxy* muito utilizado atualmente, o *ISA Server*, e por propor uma análise em diferentes cenários e métricas, que, ao longo do desenvolvimento do projeto, serão estudados e selecionados de forma apropriada.

Por fim, acreditamos que esse trabalho contribuirá com os administradores de redes, gerentes de TI e todas as pessoas que utilizam um servidor *Proxy* como solução para a melhora do desempenho e da segurança no acesso à Internet. Adicionalmente será possível adquirir conhecimento técnico específico que poderá ser muito útil no futuro, uma vez que a implantação dessas ferramentas não é trivial.

3 Objetivos

3.1 Objetivo geral

- Com esse trabalho pretende-se apresentar um estudo comparativo entre três soluções para implantar um servidor *Proxy*. São elas: *Varnish*, *Squid* e o *ISA Server*. De forma que ao final possa afirmar qual solução apresentou melhor desempenho em cada cenário simulado. E por fim, esse trabalho pretende auxiliar profissionais da área de TI na escolha de boas soluções para resolver problemas relacionados ao acesso e a segurança na Internet, através de servidores *Proxy*.

3.2 Objetivos específicos

- Adquirir conhecimento técnico específico para a implantação de todas as soluções que serão estudadas.
- Contribuir com a comunidade de profissionais de TI, uma vez que a implantação e otimização das três soluções discutidas, *Squid*, *Varnish* e o *ISA Server* não é trivial.
- Modelar e simular situações em diferentes cenários de teste, visando obter uma análise de desempenho abrangente sobre diferentes aspectos.
- Fazer uma análise completa dos dados coletados, apontando conclusões claras e objetivas.

4 Metodologia

Este trabalho será desenvolvido em quatro fases, a primeira consiste em fazer uma revisão bibliográfica em busca de métricas e cenários que possam analisar de forma eficiente o desempenho de servidores *Proxy* sobre diferentes aspectos. A segunda fase será a implantação das três soluções propostas, nessa fase também realizaremos alguns testes simples com o objetivo de encontrar algum cenário diferente dos encontrados na revisão bibliográfica. A terceira fase tem como objetivo simular os cenários para que os testes possam ser feitos, na quarta e última fase acontecerá a análise dos dados apresentados pelos testes. A primeira fase e a segunda serão desenvolvidas na disciplina de Monografia I e o restante, bem como a conclusão do trabalho, será desenvolvida na disciplina de Monografia II.

5 Desenvolvimento

Nessa seção descreveremos as duas primeiras etapas do projeto, elas foram propostas e desenvolvidas na disciplina de Monografia I. Na primeira etapa o objetivo era buscar na literatura algumas métricas e cenários possíveis para realizar os testes de performance nos servidores implantados. Na segunda etapa o objetivo era implantar as três soluções com configuração padrão e realizar testes simples. O teste realizado foi apenas o de verificar o funcionamento básico do *Proxy*, ou seja se as requisições estão passando pelo servidor.

5.1 Cenários e Métricas

O objetivo de busca por cenários e métricas ideais é muito importante para obter bons resultados no fim do projeto. Nesse sentido buscamos na literatura algumas formas de realizar esse *benchmark*. Inicialmente não é possível construir os cenários sem realizar algum teste, pois os mesmos é que vão defini-los. Porém escolhemos parâmetros que serão usados nessa construção, por exemplo o número de usuários requisitando páginas Web simultaneamente, o tipo dessas páginas (dinâmicas ou estáticas), o tamanho dos dados requisitados e a localização do conteúdo (cache ou externo). Esses parâmetros serão utilizados por dois softwares para construir um possível cenário de teste. Eles fornecem os resultados em métricas que são importantes de serem analisadas, como tempo de resposta, transações atendidas por segundo e a velocidade de transferência. É importante ressaltar que os cenários serão montados até atingirmos a capacidade máxima de *hardware* da estação de teste. A seguir descreveremos a maneira que cada software será utilizado na construção de um cenário de teste, como as possíveis métricas que usaremos no *benchmark*.

- O primeiro é utilizando o *Siege*. Nesse software poderemos realizar testes simulando usuários acessando páginas da *Web* simultaneamente [10]. O número de usuários que simularão esses acessos pode ser escolhido. Com esse software poderemos obter resultados como o número de transações atendidas por segundo, a quantidade de dados processadas por segundo, o tempo médio de resposta, dentre outras métricas que durante os resultados verificaremos se serão relevantes ou não na conclusão do trabalho.
- O segundo cenário seria utilizando o *http_load*, um software que possibilita testar a velocidade de transferência [5]. Dessa forma criaremos diversos arquivos com diferentes tamanhos e colocaremos em um servidor Web. Depois utilizaremos o software para simular um número de usuários solicitando várias requisições simultâneas à massa de dados criada.

5.2 Estações de Trabalho

Para implantar os *Proxies Squid* e *Varnish* instalamos o sistema *FreeBSD*, na sua versão 9.0. O que justifica a escolha desse sistema é o fato dele ser considerado robusto e estável, geralmente é amplamente utilizado por empresas em servidores, provedores de serviço Internet, por pesquisadores, profissionais de informática e estudantes. [3]

Para implantar o *ISA Server* utilizamos o sistema operacional *Windows Server*, por ser o único sistema que o suporta. Inicialmente escolhemos a versão 2003.

O *hardware* utilizado foi um *Intel Core 2 Duo 2.00 GHz 2MB de Cache, 2 GB de RAM* e com *250 GB* de espaço em disco.

5.3 Implantação

Nessa seção descreveremos como foi realizada a implantação das três ferramentas. Como foi dito anteriormente a configuração delas, nesse primeiro momento, foi básica. E por fim iremos relatar as principais dificuldades encontradas para cada uma.

5.3.1 Squid

Utilizamos a versão 3.2 do *Squid* por ser a última versão estável disponível. A seguir descreveremos os passos para a configuração básica que fizemos.

A instalação foi feita pelo repositório *ports* do *FreeBSD* (*/usr/ports/www/squid3.2/*), a única dependência a ser contemplada no sistema operacional é o *Perl*.

O próximo passo é a criação das pastas de cache e log. Em máquinas que são utilizadas para esse tipo de servidores normalmente é criado uma partição só para o cache, nesse momento não era o objetivo portanto não o fizemos. Logo após criamos um usuário e um grupo do sistema operacional para o *Squid*.

O *Squid* é baseado em *ACLs* (Access Control List), elas são nada mais que regras para navegação via proxy. Elas são responsáveis por liberar ou bloquear o conteúdo.

Tanto *ACLs* quanto as configurações gerais do *Squid* são definidas no arquivo *squid.conf*. No *Arquivo 1* vamos colocar a configuração inicial utilizada com comentários para facilitar o entendimento.

```
#configura porta do squid
http_port 3128

# Esse valor a principio é a metade da memória RAM,
#estudaremos com mais detalhes depois.
cache_mem 1000 MB
cache_swap_low 90
cache_swap_high 95

# Aqui é o tamanho máximo de sua cache
cache_dir ufs /usr/local/squid3.2/ 4000 16 256
access_log /usr/local/squid3.2/access.log

#configurações de log
cache_log /usr/local/squid3.2/cache.log
cache_store_log /usr/local/squid3.2/store.log

#configurações básica das ACL's
acl manager proto cache_object
acl localhost src 127.0.0.1/32
```

```

acl vlan24 src 10.1.1.0/24 # Representa a sua rede e máscara de sub-rede
acl SSL_ports port 443 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 563 1863 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT

#libera o acesso ou nega o acesso para as acls definidas anteriormente
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

#seta usuários e define nome do host
cache_effective_user squid
cache_effective_group squid
visible_hostname server

```

Arquivo 1. O *squid.conf* utilizado

Para o funcionamento do *Squid* como um *Proxy* deveremos ter um *firewall* instalado para redirecionar as requisições na porta 80 para a porta 3128, a porta que ele utiliza. No nosso caso utilizamos o *IPFW*.

Essas configurações foram pesquisadas em três fóruns o *FUG-BR* [4] , o *Squid-Cache* [13] e o *Squid-Cache.br* [12], que alimentam boa parte da comunidade *Squid* em tutoriais, artigos e discussões.

5.3.2 Varnish

A última versão estável disponível do *Varnish* é a 3.0.3, por isso ela foi utilizada nesse primeiro momento. A seguir descreveremos os primeiros passos para uma configuração básica da ferramenta.

De forma análoga ao *Squid*, instalamos o *Varnish* pelo repositório *ports*. Tudo ocorreu normalmente e ele foi instalado no diretório */usr/local*.

A primeira tarefa pós-instalação foi fazer com que as requisições para os servidores Web sejam feitas para o *Varnish*. Isto é feito no arquivo de configuração padrão o *default.vcl*.

Para entender melhor o arquivo de configuração introduziremos o *VCL*(*Varnish Configuration Language*), que nada mais é que uma linguagem de configuração de pequeno porte, especificamente desenvolvida para ser usada na definição e na manipulação de políticas de cache para o *Varnish*. [1]

A configuração básica que definiu a rede local e a porta que ele escutará segue abaixo:

```
backend default {  
    .host = "10.1.1.1";  
    .port = "80";  
}
```

Logo após executamos o *Varnish* em modo root com alguns parâmetros que esclareceremos após.

```
$ varnishd -f /usr/local/varnish/default.vcl -s malloc,1G -T 127.0.0.1:2000 -a 0.0.0.0:8080
```

- `-f /usr/local/varnish/default.vcl` - indica o caminho para o arquivo de configuração
- `-s malloc,1G` - define o tipo de armazenamento utilizado para o cache
- `-T 127.0.0.1:2000 -a 0.0.0.0:8080` - especifica que o Varnish escutará a porta 8080

O *Varnish* possui um conjunto de ferramentas que vamos explorar até o fim desse trabalho. Dentre as de destaque, um módulo completo de interface *Web* que possui diversos recursos, como por exemplo um sistema que oferece gráficos em tempo real das estatísticas de cache do *Proxy*.

5.3.3 ISA Server

O *ISA Server* possui uma licença da *Microsoft* de 180 dias para teste, ele pode ser baixado pelo no próprio site¹ da gigantesca na área de soluções em TI. A versão escolhida foi a última versão lançada, a 2006 Service Pack 1. Apesar de ter sido lançada há algum tempo, ela ainda possui suporte e é uma solução encontrada para softwares proprietários.

A instalação não é complicada, a única parte que merece atenção nesse momento é a definição da rede interna que fizemos durante a própria instalação.

O *ISA* desfruta de um gerenciador com uma interface bem amigável, o que facilita a configuração. O princípio básico para o funcionamento de um *Proxy* é configurar o tráfego para passar por ele. O *ISA* bloqueia todo o tráfego por padrão, o que fizemos então foi liberá-lo e fazê-lo escutar a porta 8080. Por fim, configuramos os arquivos de log e cache para realmente testar se o tráfego estava passando pelo servidor [15].

5.3.4 Dificuldades Encontradas

É importante fazer um comparativo das dificuldades encontradas para cada uma delas, uma vez que encontramos mais dificuldade no *Squid* e no *Varnish*, do que no *ISA Server* nesse primeiro momento. A dificuldade encontrada em comum na implantação do *Squid* e do *Varnish* foi a utilização do *FreeBSD* como sistema operacional, pois a instalação e configuração dessas duas ferramentas não é tão simples em sistemas sem

¹<http://www.microsoft.com/brasil/servidores/isaserver/prodinfo/trial-software.msp>

interface gráfica. Nas configurações para o *Squid*, a maior dificuldade foi encontrar em todas as comunidade dele na Internet alguma que o configure para a versão 3.2, não encontramos nenhuma e tivemos que testar configurações de versões anteriores. Para o *Varnish* a falta de artigos que auxiliem na sua configuração foi a principal dificuldade. A implantação e configuração do *ISA Server* não proporcionou muitos trabalhos inicialmente, não tivemos nenhuma maior dificuldade para instalá-lo e configurá-lo em suas interfaces amigáveis. Além disso, a comunidade da *Microsoft* disponibiliza diversos tutoriais para o *ISA Server* na Internet, desde configurações básica as mais complexas.

6 Considerações Finais e Trabalhos Futuros

Os resultados preliminares obtidos até o momento são a implantação de todos os *Proxies* que iremos estudar durante o trabalho. Como também a elaboração de cenários e métricas que serão testadas no intuito de obter resultados que possam ser usados na construção de um quadro comparativo entre as soluções.

Nesse primeiro momento não foi possível concluir um resultado sobre qual solução é melhor ou pior para um determinado cenário, pois está análise estaria sendo feita apenas em cima da implantação desses servidores, o que seria muito pouco.

Para os trabalhos futuros, que serão realizados na disciplina de Monografia II, pretende-se buscar na literatura maneiras de otimizar as configurações dos *softwares* implantados. Todavia essa melhora só acontecerá de acordo com a execução dos testes, que também serão realizados nessa fase. De forma que eles mostrarão qual a melhor configuração para o cenário testado, e que ao final os resultados sejam suficientes para realizar uma análise comparativa.

Pretendemos concluir o trabalho analisando as soluções implantadas, de maneira que seja mostrado os principais pontos ao escolher uma ferramenta que implemente um *Proxy*. E por fim, criaremos um guia passo-a-passo de como configurar cada uma dos servidores implantados.

7 Cronograma de atividades para BCC 391 - Monografia II

Na Tabela 1, apresentamos o cronograma proposto para as atividades da disciplina de Monografia II.

Atividades	Dez	Jan	Fev	Mar	Abr
Configuração do <i>Squid</i> .	X				
Configuração do <i>Varnish</i> .		X			
Configuração do <i>ISA Server</i> .			X		
Realização dos testes.		X	X		
Redação da Monografia				X	X
Apresentação.					X

Tabela 1: Cronograma de Atividades.

Referências

- [1] <http://www.catalystframework.org/calendar/2008/14>, acesso em Outubro de 2012.
- [2] Brian D. Davison. A web caching primer. *IEEE Transaction on Consumer Electronics*, 5(4):38–45, August 2001.
- [3] http://www.freebsd.org/doc/pt_BR.IS08859-1/books/faq/introduction.html, acesso em Setembro de 2012.
- [4] www.fug.com.br/, acesso em Setembro de 2012.
- [5] http://acme.com/software/http_load/, acesso em Agosto de 2012.
- [6] R. Howard and B. J. Jansen. A proxy server experiment: an indication of the changing nature of the web. *Proc of the Seventh International Conference on computer Communications and Networks ICCCN '98*, pages 646–649, 1998.
- [7] Informações sobre o produto. <http://www.microsoft.com/brasil/servidores/isaserver/prodinfo/default.msp>, acesso em Abril de 2012.
- [8] C. Onime O. Abiona, T. Anjali and L. Kehinde. Proxy server experiment and the changing nature of the web. *IEEE Transaction on Consumer Electronics*, 2008.
- [9] Gerson Raymond. Projeto squid. Trabalho de conclusão de curso, Universidade Ibirapuera, 2007.
- [10] <http://www.joedog.org/siege-home/>, acesso em Agosto de 2012.
- [11] Lucas Silva. Aceleração http: Um comparativo de performance entre as soluções squid e varnish. Monografia, Universidade Federal de Lavras, 2009.
- [12] www.squid-cache.org.br/, acesso em Outubro de 2012.
- [13] <http://www.squid-cache.org/>, acesso em Outubro de 2012.
- [14] Andrew S. Tanenbaum. *Redes de Computadores*. Campus, 3ª edition, 1997.
- [15] <http://technet.microsoft.com/pt-br/library/cc668550.aspx>, acesso em Outubro de 2012.
- [16] Kristian Lyngstol Tollef Fog Heen and Jérôme Renard. *Varnish Book*. Redpill Linpro, 1ª edition, 2012.
- [17] Duane Wessels. *Squid: The Definitive Guide*. O'Reilly and Associates, 1ª edition, 2004.