

Análise Comparativa entre HTTP 1.1 e HTTP 2.0

Simeí Tabordes Gonçalves¹, Eduardo Maroñas Monks¹

¹Curso de Tecnologia em Redes de Computadores
Faculdade de Tecnologia SENAC Pelotas (FATEC)
Rua Gonçalves Chaves 602 – 96015560 – Pelotas – RS – Brazil

{tabordes, emmonks}@gmail.com

Abstract. *This article aims to show a comparative analysis in a test environment, the impact generated by the adoption of the HTTP 2.0 protocol in web servers. Will be used as web servers Apache, Nginx, G-WAN and IIS. The tests consists in multiple access to a server, where the results will be evidenced by the log analysis and packet capture.*

Keywords: HTTP2, Apache, Nginx, G-WAN, IIS

Resumo. *Este artigo se propõe a mostrar através de uma análise comparativa em ambiente de testes, o impacto gerado pela adoção da versão 2.0 do protocolo HTTP em servidores de páginas. Serão usados como servidores Apache, Nginx, G-WAN e IIS. Os testes consistem em múltiplos acessos em um servidor, onde os resultados serão evidenciados pela análise de logs e captura de pacotes.*

Palavras-Chave: HTTP2, Apache, Nginx, G-WAN, IIS

1. Introdução

Em maio de 2015 foi publicada a RFC7540 [Belsh 2015] que padroniza o HTTP 2.0, que teve como foco segurança e desempenho. Esta nova versão do protocolo tem como características principais a utilização da compactação como padrão, obrigatoriedade do uso de SSL e multiplexação de requisições. O resultado final é um protocolo mais seguro que usa menos conexões e consequentemente menos recursos de rede. Com a expansão constante da Internet e um número cada vez maior de usuários, a busca por eficiência entre os protocolos de rede é a chave para um melhor aproveitamento dos recursos de clientes e servidores, resultando numa melhor experiência para o usuário final. As características do HTTP 2.0 como multiplexação e compactação de cabeçalhos tornam a comunicação mais robusta e aumentam a escalabilidade dos servidores. Neste artigo serão mostrados testes que irão evidenciar o impacto dessas novas características.

2. Protocolos

O protocolo HTTP é um dos mais utilizados na Internet. Segundo NETCRAFT [NETCRAFT 2015], existem hoje aproximadamente 176.788.328 páginas disponíveis.

2.1. Protocolo HTTP 1.1

O HTTP 1.1 [Fielding 1999] é um protocolo no nível de aplicação que vem sendo usado desde a sua versão 0.9 em 1990. Na versão 1.0 ainda não era implementada a técnica de *pipelining*, que acabou sendo implementada na versão 1.1. O HTTP *pipelining* foi um grande avanço, porque consistia no envio de múltiplas requisições dentro de uma

conexão TCP, enquanto no HTTP 1.0 a cada requisição resultava em mais uma conexão TCP. Embora esta melhoria tenha sido significativa em 1999, quando as conexões eram mais lentas que hoje em dia, havia uma limitação. As respostas do servidor às requisições tinham que ser retornadas na mesma sequência que foram enviadas. A Figura 1 ilustra a técnica de *pipelining*.

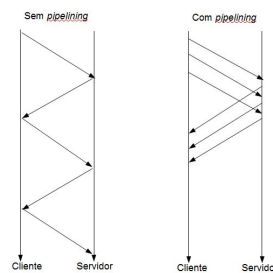


Figura 1. HTTP *pipelining*.

2.2. Protocolo SPDY

O SPDY [Google_inc 2012] é um protocolo que tem por objetivo reduzir o tempo de carregamento das páginas. Este protocolo permite compactar cabeçalho de respostas e requisições, o que reduz a banda utilizada quando os cabeçalhos se repetem, o que acontece frequentemente. Permite também a multiplexação de requisições, ou seja, substituiu o HTTP *pipelining*, já que agora as requisições são enviadas de forma assíncrona e, as respostas também retornam de forma assíncrona. No início de 2015 o time de desenvolvimento do SPDY achou por bem abandonar o projeto e se juntar ao time de desenvolvimento do HTTP 2.0, já que a maioria dos benefícios do SPDY já estavam presentes no HTTP 2.0 [Grigorik 2013]. Na Figura 2, um diagrama que compara o uso de *pipelining* do protocolo HTTP 1.1 com a multiplexação no SPDY.

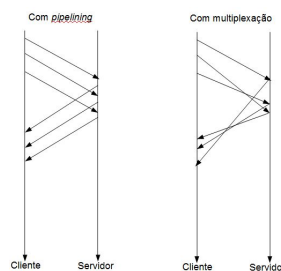


Figura 2. Comparação do uso de multiplexação do protocolo SPDY com o *pipelining* do protocolo HTTP 1.1.

2.3. Protocolo HTTP 2.0

No mês de maio de 2015, foi registrada a RFC 7540 [Belsh 2015] referente ao HTTP 2.0. Algumas características como o uso padrão de SSL e compressão, que já existiam no HTTP 1.1 apenas foram adotadas por padrão, e outras inteiramente novas tais como o uso de *streams*, multiplexação e compressão de cabeçalhos, foram adicionadas.

2.3.1. Streams e Multiplexação

A multiplexação se dá pelo uso de *streams*. Uma *stream* é uma sequência de *frames* trocados entre cliente e servidor de forma bidirecional. Uma conexão HTTP2 pode conter múltiplas *streams* concorrentes abertas. A *stream* pode ser estabelecida e usada unilateralmente ou compartilhada pelo cliente ou servidor, e podem ser fechadas por qualquer uma das pontas.

2.3.2. Server Push

O HTTP 2.0 permite ao servidor opcionalmente enviar *frames* antecipadamente para um cliente. Isto é útil quando o servidor sabe que o cliente irá necessitar de um objeto na página, antes mesmo dele fazer a requisição. Através da multiplexação, esses frames não requisitados pelo cliente chegam em paralelo com os que foram requisitados.

2.3.3. Gerenciamento de conexões

As conexões HTTP 2.0 são persistentes. É esperado que clientes não fechem suas conexões, até que nenhuma comunicação com o servidor seja necessária. Os clientes não devem abrir mais de uma conexão TCP.

2.3.4. Compressão de cabeçalhos

As listas de cabeçalho são uma coleção de zeros ou mais cabeçalhos. Quando transmitidos por uma conexão, uma lista de cabeçalhos é serializada em um bloco usando compressão de cabeçalho HTTP.

2.4. SSL obrigatório

Hoje ainda opcional nos servidores de páginas, devido a maioria ainda utilizar HTTP 1.1, no HTTP 2.0 tornou-se obrigatório o uso de SSL nas conexões.

3. Servidores de páginas

Foram testados os servidores G-WAN [Trueleap 2007], Nginx [Sysoev 2004], Apache [Behlendorf 1995] e o Microsoft IIS [Microsoft 2015]. Os três primeiros são de código aberto, enquanto que o IIS da Microsoft é de código fechado.

3.1. G-WAN

O servidor de páginas G-WAN, é focado em multiprocessamento. A idéia é que otimizar o aproveitamento de múltiplos cores dará os melhores resultados. Até o momento não há notícias de que o G-WAN passará a suportar HTTP 2.0, porém foi incluído nos testes para comparação com os outros servidores de páginas, devido ao desempenho otimizado com o protocolo HTTP 1.1.

3.2. Nginx

O Nginx é um servidor de páginas de código aberto e alta performance. Entre suas grandes vantagens estão a grande escalabilidade e baixo uso de memória RAM.

3.3. Apache

O Apache é um servidor de páginas de código aberto lançado em 1995. Tem um grande número de extensões e sempre se mantém em sincronia com os padrões HTTP atuais. É o servidor mais popular desde abril de 1996 NETCRAFT [NETCRAFT 2015].

3.4. Microsoft IIS

O servidor de páginas IIS, é um serviço disponível em sistemas operacionais Windows, desde o Windows NT 3.51. Segundo NETCRAFT [NETCRAFT 2015] o IIS é junto com Apache e Nginx, um dos servidores de páginas mais usados na internet.

4. Testes

Os testes foram realizados utilizando-se um notebook, como servidor de virtualização executando o VMware Workstation [VMware 2015], para os servidores de páginas, e cinco máquinas clientes usando o Linux Ubuntu 14.04.

4.1. Página de teste (Flags)

A página Flags [Liu 2015] é composta de duzentas e cinquenta e quatro bandeiras de países. As várias bandeiras geram múltiplas conexões com o servidor, e é usada para comparação de desempenho entre o HTTP 1.1 e HTTP 2.0. Cada bandeira tem aproximadamente 1,6KB. Durante os testes, as conexões usadas para fazer as requisições das bandeiras se comportaram de forma diferente entre o HTTP 1.1 e HTTP 2.0. No HTTP 1.1 o *download* das bandeiras, criou diversas conexões TCP. No HTTP 2.0, foi criada apenas uma conexão para o conteúdo da página, exemplificando o uso de *streams*. na Figura 3 a página *flags*.



Figura 3. Página de testes *Flags*

4.2. Ambiente de Testes 1

No primeiro ambiente de testes, foi usado apenas um computador, que rodava o servidor a ser testado no VMware, e também era usado como cliente que acessava o servidor a partir de um navegador com suporte a HTTP 2.0 e SPDY. Quando havia controle de banda, era feito via VMware.

4.3. Ambiente de Testes 2

Os testes foram feitos em laboratório na Faculdade de Tecnologia SENAC Pelotas. Foram usadas cinco máquinas virtuais em cinco computadores, que abriam simultaneamente várias abas com o endereço do servidor de páginas a ser testado. Foram feitos testes com

dez e cem abas sendo abertas simultaneamente em cada cliente. Os acessos eram disparados por comandos remotos via SSH. Foi utilizado o WANem para simular o ambiente de rede da Internet. Foram configurados 30ms de latência, 0.148% de perda de pacotes e 4,95Mb/s de banda entre servidores e clientes. Estes números foram baseados em estatística encontradas em um estudo feito por Chen [Chen 2012] sobre o desempenho de redes Wifi, 3G e 4G em universidades americanas.

4.4. Testes com o G-WAN

O G-WAN foi instalado em um servidor Debian Linux versão 7. Como o G-WAN não tem suporte a HTTP 2.0 ou SPDY, houveram apenas 2 testes, um de 50 conexões simultâneas e outro com 500 conexões simultâneas, ambos com HTTP 1.1. Na Tabela 1 são apresentados os resultados dos testes com o G-WAN.

Tabela 1. Resultados dos testes com o G-WAN.

Abas	Conexões(TCP)	Duração(s)	Dados(MB)
50	30	11,03	6,7
500	32	16,81	9,9

4.5. Testes com o Nginx

O servidor de páginas Nginx foi instalado em um servidor Linux CentOS 6.3 e foi ativado um módulo do SPDY 3.1. Foram feitos quatro testes de 50 e 500 abas, sendo dois com HTTP 1.1 e dois com SPDY 3.1. Na Tabela 2 estão listados os resultados deste teste.

Tabela 2. Resultados dos testes com o Nginx

Abas	Conexões(TCP)	Duração(s)	Protocolo	Dados(MB)
50	30	11,21	HTTP 1.1	6,8
500	61	13,81	HTTP 1.1	13,1
50	31	11,38	HTTPS 1.1	7
500	56	10,05	HTTPS 1.1	10,4
50	5	10,05	SPDY 3.1	7
500	5	16,81	SPDY 3.1	10,3

4.6. Testes com Apache

O servidor de páginas Apache foi instalado em um servidor Linux Debian 7.0 e ativado o módulo do SPDY 3.1. Foram feitos quatro testes de 50 e 500 conexões, sendo dois com HTTP 1.1 e dois com SPDY 3.1. Na Tabela 3 estão listados os resultados deste teste.

Tabela 3. Resultados dos testes com o Apache

Abas	Conexões(TCP)	Duração(s)	Protocolo	Dados(MB)
50	32	13,85	HTTP 1.1	6,8
500	60	11,62	HTTP 1.1	10,1
50	34	13,34	HTTPS 1.1	7,1
500	56	13,03	HTTPS 1.1	10,5
50	5	9,72	SPDY 3.1	6,4
500	5	14,47	SPDY 3.1	9,25

4.7. Testes com IIS

A versão do IIS utilizada foi a versão beta do Windows 10 Server. Através de uma modificação no registro, é possível habilitar o HTTP 2.0 [Nazim 2014].

Tabela 4. Resultados dos testes com o IIS

Aba	Conexões(TCP)	Duração(s)	Protocolo	Dados(MB)
50	31	9,43	HTTP 1.1	6,9
500	31	20,24	HTTP 1.1	10,3
50	45	11,53	HTTPS 1.1	6,9
500	30	11,41	HTTPS 1.1	7
50	5	8,6	HTTP 2.0	5,7
500	5	12,36	HTTP 2.0	8,4

5. Análise dos resultados

Na Tabela 5 fica evidente a melhoria de desempenho de todos os servidores que usaram SPDY ou Http 2.0 no teste de 50 conexões. Nos teste com 500 conexões nenhum obteve melhora.

Tabela 5. Resultado geral dos testes

	HTTP 50	HTTP 500	HTTPS 50	HTTPS 500	SPDY/H2 50	SPDY/H2 500
Apache	13,85	11,62	13,34	13,03	9,72	14,47
Nginx	11,21	13,81	11,38	10,05	10,55	16,81
IIS	9,43	20,24	11,53	11,41	8,63	12,36
G-WAN	11,03	16,81	-	-	-	-

5.1. G-WAN

O G-WAN foi o segundo melhor colocado nos testes com 50 conexões ficando atrás apenas do Microsoft IIS. Com 500 conexões ficou em na penúltima colocação, considerando a média de tempo das requisições.

5.2. Nginx

Na Tabela 2, o SPDY utilizado com o Nginx se mostrou com bom desempenho, já que no teste de 50 requisições houve uma diferença na média de tempo das requisições de 7.3% utilizando o SPDY comparado com HTTPS, enquanto que no teste de quinhentas conexões houve um acréscimo de 40.2%.

5.3. Apache

Nos testes com 50 conexões o SPDY teve um ganho de 27,13% na média de tempo das requisições, enquanto que com 500 conexões houve uma perda de 9.95% comparando com HTTPS.

5.4. IIS

No Microsoft IIS O ganho foi de 25.41% na média de tempo das requisições no teste com 50 conexões e perda 7.68% com 500 conexões comparando o HTTP 2.0 com SPDY.

6. Considerações finais

A única situação em que o SPDY 3.1 ou HTTP 2.0 ficaram sempre na frente foi com os testes de 50 conexões. Nos testes com 500 conexões os resultados foram aleatórios. Devido aos resultados encontrados pode-se concluir que os módulos de ativação do protocolo SPDY/HTTP2 disponibilizados nos servidores testados ainda não se encontram otimizados. Entretanto, o protocolo SPDY/HTTP2 gerou um volume menor de dados trafegados e um número menor de conexões TCP em todos os testes.

6.1. Habilitação de módulos

As maiores dificuldades se deram na habilitação dos módulos para Apache e Nginx. Embora haja pacotes já disponíveis do SPDY para estas distribuições Linux, há alguns problemas de compilação não devidamente documentados. Foi necessário buscar informações em diversos sites e blogs de como habilitar os módulos.

6.2. IIS

O Windows 10 Server, talvez por ser uma versão beta ainda, tem problemas de instabilidade, gerando tela azul, e obrigando a uma segunda instalação durante os testes. Entretanto o processo de habilitação do HTTP2 é o mais fácil de todos, necessitando apenas uma modificação no registro.

6.3. Testes

Os testes primeiramente, foram feitos com apenas um navegador, onde era medido apenas o tempo de carregamento da página no browser. Os resultados se mostraram favoráveis ao SPDY e HTTP 2.0, porém essa medição inclui junto o tempo de renderização do navegador. Então optou-se por criar um teste de acessos simultâneo, de 5 máquinas virtuais, em 5 computadores diferentes, o que só foi possível com o uso de scripts em shell.

6.4. Ferramentas de captura

O Wireshark ainda não possui suporte total de filtragem do SPDY e HTTP2. Ele detecta no *frame*, o tipo de pacote, mas não filtra corretamente, por que ainda está em fase de implementação. Nos testes foi feita a filtragem por IP e porta de acesso para serem analisados os pacotes.

6.5. Análise de logs

Uma das dificuldades encontradas, foram as análises de desempenho através do log dos servidores de páginas. Acontece que no SPDY ou HTTP 2.0, utilizando o IIS e o Nginx, as conexões de elementos da página flags.html, como as bandeiras, ficam com valores zerados ou muito próximos de zero. O mesmo não ocorre no Apache, onde todas as conexões, sejam elas feitas como um *stream* do HTTP 2.0 ou não, são logadas corretamente.

6.6. Trabalhos Futuros

Quando o Wireshark tiver suporte completo a filtragem de HTTP 2.0, será possível fazer um teste mais apurado. Será interessante também testar, com outros navegadores, e servidores de páginas que já tem implementado o HTTP 2.0.

Referências

- Behlendorf, B. (1995). Apache webserver. Disponível em: <http://httpd.apache.org/ABOUT_APACHE.html/>. Acesso em 15/05/2015.
- Belsh, M. (2015). Hypertext transfer protocol version 2 (http/2). Disponível em: <<https://tools.ietf.org/html/rfc7540/>>. Acesso em: 22/05/2015.
- Chen, Y.-C. (2012). Characterizing 4g and 3g networks: Supporting mobility with multi-path tcp. Disponível em: <http://people.cs.umass.edu/~yungchih/publication/12-mtcp_4g_tech_report.pdf>. Acesso em 18/05/2015.
- Fielding, R. (1999). Hypertext transfer protocol – http/1.1. Disponível em: <<https://www.ietf.org/rfc/rfc2616.txt>>. Acesso em 15/06/2015.
- Google_inc (2012). Spdy protocol. Disponível em: <<https://tools.ietf.org/html/draft-mbelshe-httpbis-spdy-00>>. Acesso em 18/04/2015.
- Grigorik, I. (2013). High performance browser networking. Disponível em: <http://chimera.labs.oreilly.com/books/1230000000545/ch12.html#_brief_history_of_spdy_and_http_2>. Acesso em 18/06/2015.
- Liu, G. (2015). World country flags demo. Disponível em: <<https://h2ohttp2.centminmod.com/flags.html>>. Acesso em 18/05/2015.
- Microsoft (2015). Iis webserver. Disponível em: <<https://technet.microsoft.com/pt-br/library/hh831725.aspx>>. Acesso em 15/05/2015.
- Nazim, S. B. (2014). Http/2 for iis in windows 10 technical preview. Disponível em: <<http://blogs.iis.net/nazim/http-2-for-iis-in-windows-10-technical-preview>>. Acesso em 18/04/2015.
- NETCRAFT (2015). May 2015 web server survey. Disponível em: <<http://news.netcraft.com/archives/2015/05/19/may-2015-web-server-survey.html>>. Acesso em 18/05/2015.
- Sysoev, I. (2004). Nginx webserver. Disponível em: <<http://wiki.nginx.org/Main/>>. Acesso em 15/05/2015.
- Trueleap (2007). G-wan webserver. Disponível em: <<http://gwan.ch/>>. Acesso em 15/05/2015.
- VMware (2015). VMware workstation. Disponível em: <<http://www.vmware.com/br/products/workstation>>. Acesso em 30/03/2015.