

Desenvolvimento Web II

Aula 10 - HTTP/2

Prof. Fabricio Bizotto

Instituto Federal Catarinense

fabricio.bizotto@ifc.edu.br

Ciência da Computação

31 de janeiro de 2024

- 1 HTTP/2
- 2 Definição
- 3 HTTP/2: Multiplexação
- 4 HTTP/2: Compressão de cabeçalhos
- 5 HTTP/2: Server Push
- 6 HTTP/2: Priorização de requisições
- 7 Exemplo
- 8 QUIZ
- 9 Lista de Exercícios

HTTP/2

Quais são as principais novidades?

Em breve, HTTP/3

- HTTP/2 é a segunda versão do protocolo HTTP.
- Foi publicado em 2015.
- Objetivo: melhorar a performance do protocolo.
- Baseado no protocolo SPDY, desenvolvido pelo Google.
- É binário, ao contrário do HTTP/1.1 que é textual.
- Principais novidades:
 - Multiplexação
 - Compressão de cabeçalhos
 - Server Push
 - Priorização de requisições

Multiplexing

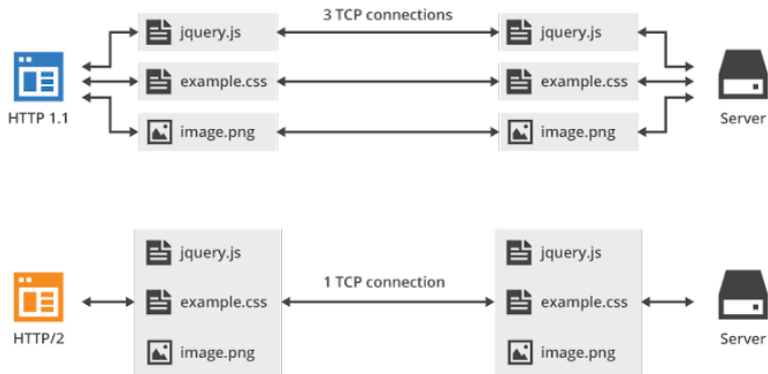


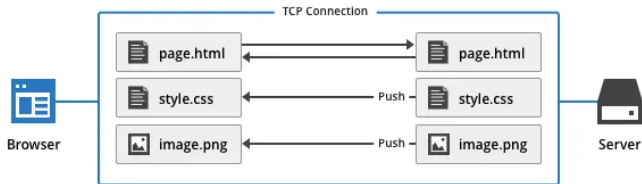
Figura: Multiplexação de requisições no HTTP/2.

- HTTP/1.1 precisa criar uma nova requisição para cada recurso.
- HTTP/2 pode enviar todos os recursos em uma única conexão, usando compressão de cabeçalhos.
- HTTP/1 usa **gzip**. É utilizado para compressão do corpo da mensagem em respostas HTTP.
- HTTP/2 usa **HPACK**. É utilizado para compressão de cabeçalhos HTTP.
- HPACK é um algoritmo de compressão de cabeçalhos baseado em Huffman¹.

¹<https://kristijansedlak.medium.com/hpack-huffman-encoder-explained-61102edd6ecc>

- Server Push é uma funcionalidade do HTTP/2 que permite ao servidor enviar recursos para o cliente antes que ele solicite.
- O servidor pode enviar recursos que ele acredita que o cliente irá solicitar.
- O cliente pode aceitar ou rejeitar os recursos enviados pelo servidor.
- Server Push é uma funcionalidade **opcional**.

HTTP/2 (With Server Push)



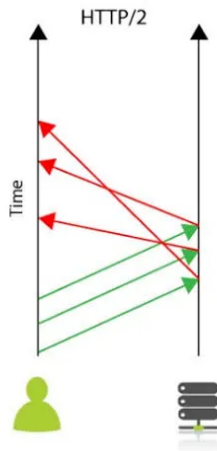
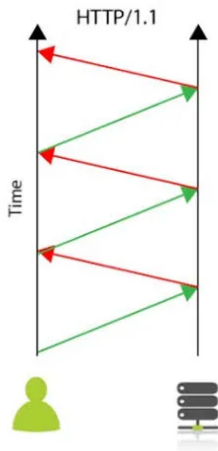
Single TCP Connection, Single HTTP Request

Figura: Server Push no HTTP/2.

HTTP/2

Priorização de requisições

- HTTP/2 permite que o cliente priorize as requisições.
- O cliente pode informar ao servidor quais requisições são mais importantes.
- O servidor pode usar essa informação para priorizar o envio de recursos.
- A priorização é feita através de um **peso** e um **dependência**.



HTTP/2

Exemplo prático

 Python + Flask + hypercorn

Qual é o principal objetivo do protocolo HTTP/2?

Resposta

- ☐ a Reduzir a complexidade do HTTP/1.1
- ☐ b Melhorar a segurança das comunicações na web
- ☐ c Aumentar a eficiência no carregamento de páginas web
- ☐ d Adicionar mais funcionalidades aos servidores web

Qual é o principal objetivo do protocolo HTTP/2?

Resposta

- ☐ a Reduzir a complexidade do HTTP/1.1
- ☐ b Melhorar a segurança das comunicações na web
- ☒ c Aumentar a eficiência no carregamento de páginas web
- ☐ d Adicionar mais funcionalidades aos servidores web

O HTTP/2 utiliza um novo formato de compressão de cabeçalhos. Qual é o nome deste formato?

Resposta

- ☐ a GZIP
- ☐ b HPACK
- ☐ c Brotli
- ☐ d Deflate

O HTTP/2 utiliza um novo formato de compressão de cabeçalhos. Qual é o nome deste formato?

Resposta

- ☐ a GZIP
- ☒ b HPACK
- ☐ c Brotli
- ☐ d Deflate

Quais são algumas das principais características do HTTP/2?

Resposta

- ☐ a Multiplexação, Priorização e Compressão de Cabeçalhos
- ☐ b Roteamento, Criptografia e Autenticação
- ☐ c Redirecionamento, Fragmentação e Cache
- ☐ d Assinatura Digital, Segurança de Transporte e Sessões Persistentes

Quais são algumas das principais características do HTTP/2?

Resposta

- ☒ a Multiplexação, Priorização e Compressão de Cabeçalhos
- ☐ b Roteamento, Criptografia e Autenticação
- ☐ c Redirecionamento, Fragmentação e Cache
- ☐ d Assinatura Digital, Segurança de Transporte e Sessões Persistentes

O HTTP/2 introduziu um conceito que permite que o servidor envie ativamente recursos para o cliente antes mesmo de serem solicitados. Qual é esse conceito?

Resposta

- ☒ a Server Push
- ☐ b Requisições Antecipadas
- ☐ c Respostas Pró-ativas
- ☐ d Inicialização Rápida

O HTTP/2 introduziu um conceito que permite que o servidor envie ativamente recursos para o cliente antes mesmo de serem solicitados. Qual é esse conceito?

Resposta

- ☒ a Server Push
- ☐ b Requisições Antecipadas
- ☐ c Respostas Pró-ativas
- ☐ d Inicialização Rápida

Em que situações é especialmente benéfico utilizar o HTTP/2?

Resposta

- ☐ a Em aplicações web com requisitos mínimos de segurança
- ☐ b Em ambientes com largura de banda limitada e conexões instáveis
- ☐ c Em aplicações onde o suporte a WebSockets é essencial
- ☐ d Em páginas web dinâmicas com muitos recursos a serem carregados

Em que situações é especialmente benéfico utilizar o HTTP/2?

Resposta

- ☐ a Em aplicações web com requisitos mínimos de segurança
- ☐ b Em ambientes com largura de banda limitada e conexões instáveis
- ☐ c Em aplicações onde o suporte a WebSockets é essencial
- ☒ d Em páginas web dinâmicas com muitos recursos a serem carregados

O HTTP/2 é retrocompatível com o HTTP/1.1. O que isso significa?

Resposta

- ☐ a Aplicações HTTP/1.1 não podem ser executadas em servidores HTTP/2
- ☐ b Aplicações HTTP/2 não podem acessar recursos servidos por servidores HTTP/1.1
- ☐ c Aplicações HTTP/1.1 podem continuar funcionando normalmente em um ambiente HTTP/2
- ☐ d Aplicações HTTP/2 precisam ser reescritas para serem executadas em ambientes HTTP/1.1

O HTTP/2 é retrocompatível com o HTTP/1.1. O que isso significa?

Resposta

- ☐ a Aplicações HTTP/1.1 não podem ser executadas em servidores HTTP/2
- ☐ b Aplicações HTTP/2 não podem acessar recursos servidos por servidores HTTP/1.1
- ☒ c Aplicações HTTP/1.1 podem continuar funcionando normalmente em um ambiente HTTP/2
- ☐ d Aplicações HTTP/2 precisam ser reescritas para serem executadas em ambientes HTTP/1.1

Lista de Exercícios

HTTP/2

- 1 Quais são as principais diferenças entre o HTTP/1.1 e o HTTP/2?
- 2 Quais são as principais vantagens do HTTP/2 em relação ao HTTP/1.1?
- 3 Quais são as principais desvantagens do HTTP/2 em relação ao HTTP/1.1?
- 4 O que são e como funcionam os "cabeçalhos de push" no HTTP/2? Dê um exemplo de situação em que eles podem ser úteis.
- 5 Sobre HTTP/3. Quais são as principais diferenças em relação ao HTTP/2?

Entrega

- Até a próxima aula.
- Manuscrito.

Desafio

Implementar um servidor HTTP/2 usando um servidor web (NGinx, Apache, etc.).