

Um Servidor HTTP/2 Reativo em Scala

Daniel Q. Miranda

Orientador: Prof. Dr Daniel Macêdo Batista

Departamento de Ciência da Computação - Instituto de Matemática e Estatística - Universidade de São Paulo

danielqm@linux.ime.usp.br

batista@ime.usp.br



IME-USP

Problemática

O HTTP (**HyperText Transfer Protocol**) é o principal protocolo de comunicação entre aplicações na Internet. Porém, apresenta limitações devido ao seu propósito inicial de apenas transmitir documentos na Web. Atualmente, através dele trafegam aplicações complexas, *streaming* de áudio e vídeo, *downloads*, etc. em larga escala

Julgou-se necessário, então, modernizar o HTTP para atender estas novas demandas de maneira eficiente. Para isso, o **HTTP/2** foi criado pela IETF (Internet Engineering Task Force) como seu sucessor.



O Protocolo

O HTTP/2 introduz, dentre outras mudanças:

- ▶ **Paralelismo** de transmissões para aproveitar ao máximo recursos em uma única conexão;
- ▶ Representação de mensagens como **frames binários**, e não mais texto;
- ▶ **Iniciativa bi-direcional**: além de clientes, como é tradicional, servidores também podem iniciar transmissões de recursos (sujeitas a confirmação);
- ▶ **Compressão de metadados** (*headers*) eficiente e resistente a ataques contra criptografia e privacidade;
- ▶ **Priorização de transmissões**, com fracionamento de recursos configurável por servidor e cliente;

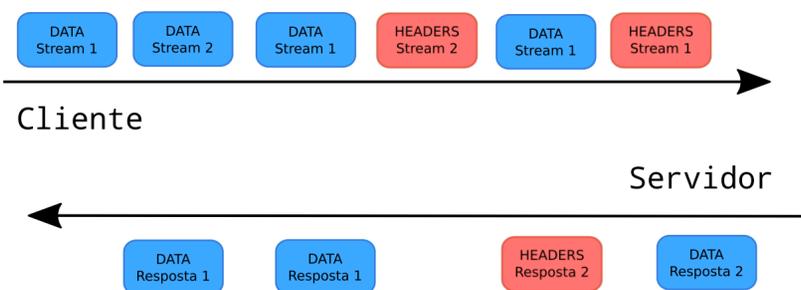


Figura: Representação visual de frames de uma sessão HTTP/2

Proposta

Implementar um servidor Web HTTP/2 na linguagem de programação **Scala**, utilizando a biblioteca de concorrência **Akka**, com foco em correção, elegância e escalabilidade, aplicando princípios da **Programação Reativa**; ela determina que um sistema deve ser:

- ▶ **Responsivo**: Fornecer tempos de resposta rápidos, consistentes e confiáveis;
- ▶ **Resiliente**: Detectar e recuperar-se de falhas sempre que possível;
- ▶ **Elástico**: Absorver variações de demanda e ser escalável;
- ▶ **Orientado a mensagens**: garantir baixo acoplamento e alto isolamento de componentes estabelecendo fronteiras claras entre eles;

Resultados

O sistema é capaz de manter uma sessão HTTP/2 com sucesso, comunicando-se com utilitários de linha de comando, e browsers que não apliquem obrigatoriedade do uso de Transport Layer Security; esse mecanismo não foi implementado devido à falta da extensão de negociação de protocolo (*ALPN*) nas bibliotecas disponíveis.

Desenvolvimentos futuros incluem aplicação automática de paralelismo, manejo automático de controle de fluxo, e otimizações de desempenho.

Scala, Akka e Programação Reativa

Scala é uma linguagem de programação que combina paradigmas funcional e orientado a objetos na Java Virtual Machine, com sistema de tipos avançado e alta expressividade, facilitando o desenvolvimento de sistemas robustos e elegantes.

Akka é um *toolkit* desenvolvido em Scala para programação concorrente com o **modelo de atores**, no qual entidades computacionais se comunicam apenas através de mensagens discretas. A biblioteca **Akka Streams** adiciona a esse modelo ferramentas para criação de fluxos de dados robustos, adaptáveis à demanda e de funcionamento transparente, como proposto pelo **Manifesto Reativo**.



Implementação

O servidor foi construído através da composição de diferentes **estágios de processamento** e atores responsáveis por diferentes tarefas. O fluxo de dados é controlado por **back-pressure**; cada estágio só produz dados em resposta direta a pedidos de seus vizinhos. A *demand* se propaga até a *fonte*, um dado emitido é processado, e finalmente *consumido* pelo destino.

Uma API para que aplicações incorporem o servidor é fornecida, em contraste com outros modelos nos quais o servidor hospeda a aplicação.

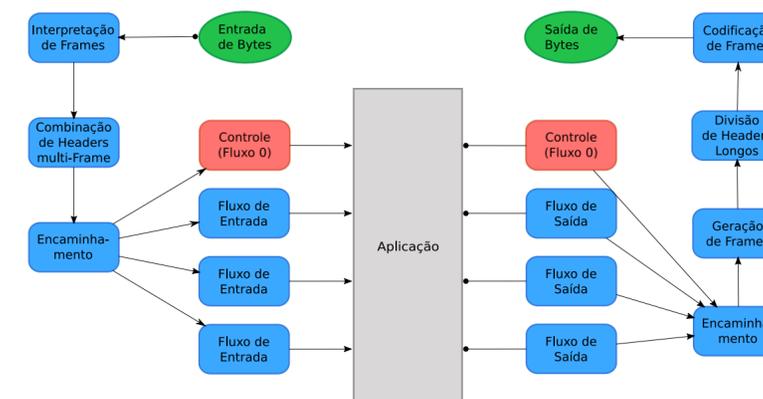


Figura: Diagrama de estágios e fluxo de dados

Carl Hewitt. "Viewing control structures as patterns of passing messages". Em: *Artificial Intelligence* 8.3 (1977), pp. 323–364. ISSN: 0004-3702. DOI: 10.1016/0004-3702(77)90033-9.

Hypertext Transfer Protocol Version 2 (HTTP/2). RFC 7540. RFC Editor, maio de 2015. URL: <http://www.rfc-editor.org/rfc/rfc7540.txt>.

Roland Kuhn et al. Jonas Bonér Dave Farley. *The Reactive Manifesto*. 2015. URL: <http://www.reactivemanifesto.org/>.