

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA POLITÉCNICA
ENGENHARIA DE COMPUTAÇÃO E INFORMAÇÃO

PROPOSTA DE PROJETO DE GRADUAÇÃO

Aluno: Renan Fasolato Basilio
renanbasilio@poli.ufrj.br

Orientador: Geraldo Zimbrão

1 Título

Streaming de Mídia Dinâmico-Adaptativo via WebSocket

2 Ênfase

Computação

3 Tema

O tema do trabalho é o estudo da viabilidade da utilização do protocolo WebSocket para streaming de mídia dinâmico-adaptativo, e suas vantagens e desvantagens em relação às tecnologias atualmente dominantes.

4 Delimitação

A realização do projeto estará restrita a mídias pré-fragmentadas e em container MPEG, estando fora do escopo do mesmo a análise e processamento de qualquer característica interna da mesma. Além disso, a maioria dos testes serão realizados em um ambiente simulado, podendo não refletir completamente todas as condições possíveis da rede real.

Por fim, a utilização do protocolo WebSocket para transferência dos dados implica na utilização do protocolo TCP, sobre o qual esse é implementado.

5 Justificativa

As tecnologias para streaming de mídia dominantes atualmente são baseadas no protocolo HTTP/S, sendo afetadas pelas desvantagens consolidadas do mesmo, dentre elas o consumo de banda com repetidos cabeçalhos do protocolo, a necessidade de reestabelecer conexão com o servidor (e, potencialmente, de refazer handshakes quando se trata de conexões seguras por SSL) para cada parte da mídia e a latência proveniente da necessidade de o cliente solicitar a próxima parte da mídia.

No entanto, com a recente standardização do protocolo WebSocket, que permite o estabelecimento de uma conexão duplex reutilizável entre a página visualizada pelo cliente e o servidor remoto, e sua implementação na maioria dos browsers, torna-se viável uma nova solução para streaming de mídia baseada neste protocolo. A utilização deste protocolo permitirá cortar a maioria das latências provenientes do protocolo HTTP/S, e, por consequência, melhorando a capacidade de adaptação da mídia às variações na condição da rede.

Um trabalho semelhante existe em [1], porém neste apesar de utilizar técnicas de Push para diminuir a latência a responsabilidade quanto à decisão sobre a necessidade de adaptação da stream pertence ainda ao cliente, o que implica em latência no necessitar adaptação na transferência da mídia.

6 Objetivo

O objetivo deste trabalho é o desenvolvimento de um par cliente-servidor para streaming de mídia dinâmico-adaptativo através da internet utilizando o protocolo WebSocket.

Essa implementação visa transferir a responsabilidade de decidir as características da mídia a ser enviada do cliente para o servidor, tal que este possa transmitir um fluxo constante de dados sem qualquer latência entre dois fragmentos consecutivos.

7 Metodologia

O cliente terá a forma de uma biblioteca em JavaScript, utilizando a API Media Source Extensions para controlar um reprodutor de vídeo HTML5, e se comunicando com o servidor através de uma WebSocket. O cliente enviará mensagens de controle ao servidor para determinação de elementos como por exemplo a seleção da mídia a ser transmitida.

O servidor será desenvolvido como um webapp em Java capaz de estabelecer conexão com um cliente, receber mensagens de controle deste e enviar e mediante análise heurística das condições da rede na comunicação com o mesmo decidir as características do próximo fragmento da mídia a ser enviado ao cliente.

A mídia a ser transmitida será pré-processada a fim de fragmentá-la, tornando assim desnecessário qualquer processamento sobre a mesma por parte do servidor.

Para a realização dos testes será utilizada uma máquina virtual na qual a aplicação tc será configurada para simular diversas condições de rede distintas. Esta escolha remete ao fato de que a utilização uma configuração simulada em cima de um sistema sem qualquer variação física resulta em dados altamente reprodutíveis, sem dependência das condições momentâneas da rede física.

8 Materiais

Para o desenvolvimento do sistema será utilizado software agnóstico ao sistema operacional, tornando o mesmo irrelevante a este projeto. As linguagens de programação utilizadas serão Java, JavaScript, HTML5 e CSS.

O ambiente de desenvolvimento será o Microsoft Visual Studio Code, publicado sob a licença MIT Open Source, acompanhado do Apache Maven como toolchain de compilação, publicado sob a licença Apache 2.0 de software livre.

O servidor utilizado será o Apache Tomcat 9, também publicado sob a licença Apache de software livre.

Para configuração do ambiente de testes será utilizada a aplicação de virtualização Oracle VirtualBox, publicada sob a licença MIT Open Source. O sistema operacional tanto na máquina hospedeira quanto virtual será o Ubuntu Linux, publi-

cado sob licença GPL share-alike, em suas versões Desktop (no hospedeiro) e Server (no servidor). Os testes serão realizados os browsers Mozilla Firefox e Google Chrome.

A mídia utilizada para os testes será o vídeo Big Buck Bunny, publicado pela Blender Foundation sob a licença GPL. A ferramenta FFmpeg, publicada sob a mesma licença, será utilizada para todo pré-processamento realizado sobre a mídia.

9 Cronograma

Fase 1: Desenvolvimento da Arquitetura do Cliente e Servidor

Fase 2: Desenvolvimento do Serviço de Streaming de Mídia

Fase 3: Desenvolvimento do Cliente para Extração de Dados

Fase 4: Configuração das Máquinas e Realização dos Testes de Desempenho

Fase 5: Produção de Relatório e Conclusões

Referências

- [1] Xiaona Wu, Cheng Zhao, Rong Xie, and Li Song. *Low Latency MPEG-DASH System Over HTTP 2.0 and WebSocket*, pages 357–367. 02 2018.

Rio de Janeiro, 27 de março de 2019

Renan Fasolato Basilio - Aluno

Geraldo Zimbrão - Orientador