

MyYoutube-v2

Aluno: Guilherme Araújo França

A arquitetura do sistema de armazenamento e streaming de vídeo é composta por quatro entidades principais: Cliente, Servidor Flask, Servidor Principal e Nós de Dados.

- O Cliente refere-se ao navegador ou a qualquer interface de usuário utilizada para interagir com o sistema através de solicitações HTTP.
- O Servidor Flask atua como uma extensão do cliente, tratando as solicitações HTTP e intermediando a comunicação com o Servidor Principal. Ele é responsável por fornecer uma interface web para o usuário e encaminhar as solicitações para o Servidor Principal.
- O Servidor Principal é o coração do sistema, lidando com a lógica de negócio principal e gerenciando as operações de armazenamento e recuperação de vídeos. Ele mantém o índice de vídeos e coordenada o armazenamento distribuído, abstraindo a complexidade de interação direta com os Nós de Dados.
- Os Nós de Dados são responsáveis pelo armazenamento físico dos vídeos. Eles operam sob a coordenação do Servidor Principal, que distribui os arquivos de vídeo entre eles e os recupera para streaming conforme necessário.
- O índice distribuído é mantido no Servidor Principal e não no Banco de Dados como sugerido na imagem original. Este índice é utilizado para rastrear onde os vídeos estão armazenados entre os vários Nós de Dados e facilitar a recuperação eficiente para operações de listagem, busca e streaming.

1. Componentes Arquiteturais do cliente

a. Servidor de Aplicação Flask

- Responsabilidade: Gerenciar as solicitações HTTP e executar a lógica do servidor.
- Tecnologia: Flask, um micro-framework web em Python.

b. Socket Cliente-Servidor

- Responsabilidade: Facilitar a comunicação de rede entre o cliente e o servidor para transferência de arquivos e informações.
- Tecnologia: Sockets TCP/IP em Python.

2. Serviços Definidos

a. Inserção (Upload)

- Descrição: Envio de arquivos de vídeo para o servidor.
- Rota Relacionada: /upload.

b. Listagem

- Descrição: Exibir uma lista de arquivos de vídeo disponíveis.
- Rota Relacionada: /listar.

c. Streaming

- Descrição: Transmitir vídeo do servidor para o cliente.
- Rota Relacionada: /stream.

3. Protocolos Implementados

a. HTTP para Comunicação Web

- Descrição: Usado para gerenciar as solicitações e respostas web entre o cliente (navegador) e o servidor Flask.

b. TCP/IP para Transferência de Arquivos

- Descrição: Usado para a transferência de arquivos e dados entre o cliente e o servidor através de sockets.

4. Funcionamento Detalhado

a. Listar Arquivos

- Cliente faz uma requisição HTTP para a rota /listar.
- O servidor Flask estabelece uma conexão socket com o servidor de arquivos.
- Solicita a lista de arquivos e retorna para o cliente.

b. Buscar Arquivos

- Funcionamento semelhante à listagem, mas filtrando por um nome específico através da rota /search.

c. Upload de Arquivos

- Cliente envia um arquivo através da rota /upload.
- O servidor Flask recebe e transmite este arquivo para o servidor de arquivos através de uma conexão socket.

d. Streaming de Vídeo

- Solicitação feita através da rota /stream.
- O servidor Flask inicia uma conexão socket e começa a transmitir o arquivo de vídeo em pedaços para o cliente.

1. Componentes Arquiteturais do servidor

a. Servidor de Rede

- Responsabilidade: Gerenciar conexões de rede e processar requisições de clientes.
- Tecnologia: Socket em Python, utilizando TCP/IP.

b. Threads

- Responsabilidade: Permitir o processamento simultâneo de múltiplas conexões de clientes.
- Tecnologia: Threading em Python.

c. Sistema de Arquivos

- Responsabilidade: Armazenar e gerenciar informações sobre vídeos e datanodes.
- Arquivos: database.txt (lista de datanodes), index.txt (índice de vídeos).

2. Serviços Definidos

a. Upload de Arquivos

- Descrição: Receber arquivos de clientes e distribuí-los entre datanodes.
- Implementação: Seleção aleatória de datanodes e encaminhamento do arquivo.

b. Streaming de Vídeo

- Descrição: Transmitir vídeo de um datanode selecionado para o cliente.
- Implementação: Busca por datanode contendo o vídeo e encaminhamento do stream.

c. Listagem de Arquivos

- Descrição: Fornecer uma lista de vídeos disponíveis.
- Implementação: Ler o arquivo index.txt e enviar nomes de arquivos para o cliente.

d. Busca de Arquivos

- Descrição: Buscar vídeos com base em um termo de pesquisa.
- Implementação: Filtrar o índice de vídeos para encontrar correspondências.

3. Protocolos Implementados

a. TCP/IP

- Descrição: Usado para comunicação de rede entre o servidor e os clientes, incluindo datanodes.

4. Funcionamento Detalhado

a. Upload de Arquivos

- Recebe arquivo do cliente.
- Seleciona aleatoriamente datanodes para replicação (fator de replicação = 3).
- Envia arquivo para os datanodes selecionados.
- Atualiza index.txt com informações do arquivo.

b. Streaming de Vídeo

- Recebe pedido de streaming.
- Seleciona um datanode contendo o vídeo.
- Estabelece conexão com o datanode e inicia o streaming de volta para o cliente.

c. Listagem de Arquivos

- Ao receber pedido de listagem, lê index.txt.
- Extrai e envia nomes únicos de arquivos para o cliente.

d. Busca de Arquivos

- Recebe termo de busca.
- Filtra index.txt para encontrar correspondências.
- Envia resultados para o cliente.

1. Componentes Arquiteturais de um banco de dados

a. Servidor de Rede

- Responsabilidade: Gerenciar conexões de rede e processar requisições de clientes.
- Tecnologia: Socket em Python, utilizando TCP/IP.

b. Threads

- Responsabilidade: Permitir o processamento simultâneo de múltiplas conexões de clientes.
- Tecnologia: Threading em Python.

c. Sistema de Arquivos

- Responsabilidade: Armazenar e gerenciar arquivos de vídeo recebidos.
- Localização: Diretório uploads{PORT}.

2. Serviços Definidos

a. Upload de Arquivos

- Descrição: Receber e armazenar arquivos de vídeo enviados pelos clientes.
- Implementação: Escrita de arquivos no sistema de arquivos local.

b. Streaming de Vídeo

- Descrição: Transmitir vídeo armazenado para o cliente.
- Implementação: Leitura e envio de arquivos de vídeo em pedaços.

3. Protocolos Implementados

a. TCP/IP

- Descrição: Usado para comunicação de rede entre o datanode e os clientes, incluindo outros servidores.

4. Funcionamento Detalhado

a. Upload de Arquivos

- Recebe um pedido de upload de arquivo.
- Abre um novo arquivo no diretório especificado.
- Recebe e escreve dados do arquivo até receber a indicação de conclusão.
- Fecha a conexão após salvar o arquivo.

b. Streaming de Vídeo

- Recebe um pedido de streaming.
- Abre o arquivo de vídeo solicitado.
- Envia o arquivo em pedaços para o cliente.
- Fecha a conexão após enviar o arquivo completo.

