

Descrição do Projeto para 4º Semestre – 2019

Considere um servidor de live streaming, enviado conteúdo para clientes que se conectam a ele. O servidor implementado simula uma transmissão de TV ao vivo, transmitindo blocos de um filme.

Característica do servidor:

- O conteúdo (vídeo) será previamente quebrado em blocos de 2 segundos: por exemplo
<filme>_00000.mkv <filme>_00001.mkv <filme>_00002.mkv <filme>_00003.mkv
<filme>_000040.mkv ... <filme>_10000.mkv
 - para testes serão disponibilizados arquivos com extensão mkv e mp4, selecione uma delas para implementar o player.
- Porta do servidor: 6060
- Canais: numerados de 0 a N (sempre iniciando em 0)
- Servidor invertido: após receber a requisição de um cliente (para adicionar em um canal), o servidor inicia o envio de blocos ao cliente no mesmo canal ou em outra conexão estabelecida pelo cliente.
- Para executar o servidor:
 - descompactar arquivo Servidor.zip
 - os arquivos de conteúdo estão na pasta filme:
 - se desejar utilizar conteúdo com outra extensão, obtenha um novo arquivo com a extensão desejada (exemplo: clipe ou filme), ajuste o arquivo separa.py para “quebrar” seu novo conteúdo.
 - Para executar o servidor: com python 3 ou superior execute o arquivo server.py
 - Como exemplo de cliente: na pasta “client” execute o arquivo client.py

Operações implementadas pelo servidor:

1. Conectar cliente a um canal: o cliente solicita a inserção em um canal e passa a receber a transmissão do canal (iniciando do momento atual de transmissão);
2. Requisição da lista de Clientes conectados em um canal;
3. Desconectar um cliente de um canal;
4. Número de cliente conectados: envia a quantidade de cliente conectados em um determinado canal;

Detalhes de Implementação

O Servidor e cliente teste foram implementado em Python na versão 3.

Cada canal será atendido por uma thread, que percorre uma lista de clientes enviando o bloco atual de vídeo {a transmissão para todos os elementos da lista não deve demorar mais que 2 segundos}.

Se conexão para transmissão com cliente falhar, o servidor passa para o próximo cliente da lista, se falhar novamente na próxima tentativa (quando chegar sua vez novamente) será removido da lista de transmissão.

Operações implementadas pelo servidor:

1. Recebe requisição do cliente;
2. Processa requisição do cliente de acordo com o código recebido, por exemplo:
 - Requisição para conexão em um dos canais e é possível atender, o cliente passa a receber a transmissão do canal (iniciando do momento atual de transmissão);
 - Requisição da lista de Clientes conectados em um canal;
 - Desconectar um cliente de um canal
 - Número de cliente conectados: envia a quantidade de cliente conectados em um determinado canal;

Mensagens processadas pelo servidor

Todas as mensagens enviadas pelo cliente devem ser no formato: <Codigo><Canal>
Com Código em 2 dígitos e canal com um dígito.

Requisição para conexão em um dos canais

Elementos da mensagem:

Código (2 caracteres)	Canal (1 caractere)
10	X

Exemplo: mensagem 100: incluir cliente no canal “0”

Resposta (servidor para cliente): código “10” se requisição atendida com sucesso – código “00” quando a quantidade máxima de clientes por canal é atingida, nesse caso o cliente não será atendido pelo servidor.

Requisição da lista de Clientes conectados em um canal

O cliente solicita a lista de cliente que recebem a transmissão do canal (lista utilizada pela thread)

Código (2 caracteres)	Canal (1 caractere)
11	x

Resposta: lista de endereços IP conectados no canal, testar com a geração de resposta através de uma thread específica para essa atividade (abre conexão com o cliente e envia a lista)

Desconectar um cliente de um canal

O cliente solicita a desconexão de um canal.

Código (2 caracteres)	Canal (1 caractere)
12	x

Número de clientes conectados

Envia a quantidade de cliente conectados em um determinado canal

Código (2 caracteres)	Canal (1 caractere)
13	x

Resposta: Quantidade de cliente conectados no canal.

Sua tarefa

Implementar o módulo cliente para conexão com o servidor de streaming, com as seguintes características:

- Retransmissão de Conteúdo:** após contato com o servidor, os clientes devem se configurar atendendo a característica (c) e a quantidade máxima de retransmissão possível do servidor (restrição 1), assim um novo cliente deve receber o conteúdo de um cliente ao qual se conecta.
- Tolerância a Falhas:** um módulo cliente A pode retransmitir o conteúdo para outros clientes, na ocorrência de falhas de A, os módulos clientes dependentes devem: identificar a falha, reorganizar a sequência de transmissão-recepção para manter todos os clientes restantes recebendo o conteúdo. Se um cliente apresentar falha, o elemento que atua como servidor desse cliente deve removê-lo de sua lista de transmissão.

- c) **Balanceamento de Carga:** cada elemento pode atuar como servidor para um número máximo de clientes (vamos fixar em 2). Para selecionar um servidor, verifique qual deles pode atender e utilize uma métrica (previlegiando o menor atraso). O servidor principal pode atender no máximo um cliente por canal.

Restrições:

1. O servidor apresenta uma quantidade limitada de conexões que pode atender simultaneamente, ou seja, a quantidade de clientes conectada diretamente ao servidor é limitada (somente 1 cliente por canal).
2. Os clientes podem pertencer a redes distintas (sem possibilidade de broadcast), portanto a comunicação em broadcast não será possível entre clientes.

O conteúdo recebido pelo cliente (blocos do vídeo) deverá ser armazenado em disco.

Player de conteúdo: implemente (ou utilize algum player já implementado – como por exemplo VLC), ele será executado no mesmo computador do cliente, sua função será exibir de forma contínua o conteúdo dos arquivos recebidos pelo cliente. O player deve remover (apagar o arquivo após exibição), caso opte por utilizar o VLC (ou outra implementação) essa opção deve ser realizada por outro processo.

O módulo cliente deve oferecer as seguintes opções (para gerenciamento) ao usuário:

- Listar Conexões: listar o IP+Porta do elemento que envia conteúdo e de todos os elementos que recebem conteúdo do cliente em análise;
- Listar arquivos recebidos: uma opção do cliente ou do player que exibe a lista de arquivos aguardando para exibição no cliente.

O trabalho entregue deve conter:

- código fonte;
- documentação gerada pelo doxygen.
- Relatório sobre a implementação: indicando a arquitetura adotada, mensagens transmitidas/recebidas, diagrama de estados (sequenciamento de ações), algoritmos adotados para cada uma dos requisitos solicitados;

Os testes serão realizados com o código do servidor enviado para teste.

Agradeço ao Matheus Cernelutt pela implementação do Servidor.