



UNIVERSIDADE
ESTADUAL DE LONDRINA

LUCAS BENOSSI MAGALHÃES
NICOLAS OMORI
RAFAELLA BATISTA DA SILVA

PROJETO SISTEMAS OPERACIONAIS

LONDRINA - PR
2019

Projeto de Sistemas Operacionais - 2018

1. Resumo

Conceitos básicos de Socket e padrões de Internet foram discutidos no segundo semestre do ano de 2018, bem como diversos outros dentro da disciplina de Sistemas Operacionais. Como forma de avaliação, um projeto envolvendo comunicação em rede foi desenvolvido. Neste trabalho, uma apresentação de uma aplicação em Java será apresentada, para transmissão de vídeo, utilizando diversos clientes, com comunicação em redes feita em Socket TCP.

2. Introdução

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. O servidor pode conectar um cliente a um canal, a partir da solicitação de inserção em um canal e passa a receber a transmissão do canal (iniciando do momento atual de transmissão); Pode requisição da lista de clientes conectados em um canal; Pode desconectar um cliente de um canal e enviar a quantidade de cliente conectados em um determinado canal.

Cada canal será atendido por uma thread, que percorre uma lista de clientes enviando o bloco atual de vídeo, não demorando mais que 2 segundos para a transmissão para todos os elementos na lista. Caso a 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.

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

a) 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.

b) 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.

c) Balanceamento de Carga: cada elemento do sistema (clientes) deve manter taxas de transmissão balanceada, ou seja, a quantidade de bytes enviados devem ser aproximadas.

E possui as seguintes 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.

2. Os clientes podem pertencer a redes distintas (sem possibilidade de broadcast), portanto a comunicação em broadcast não será possível entre clientes.

3. O cliente deve apresentar uma função que permita a configuração (no início de sua execução) da quantidade de elementos (outros clientes) que podem se conectar a este cliente (configuração realizada uma única vez, no início da execução do cliente).

O conteúdo recebido pelo cliente (blocos do vídeo) deverá ser armazenado em disco. O player de vídeo utilizado foi o VLC, e é executado no mesmo computador do cliente, exibindo de forma contínua o conteúdo dos arquivos recebidos pelo cliente.

3. Metodologia

O desenvolvimento deste projeto pode ser compreendido pelas seguintes etapas: estudo da tecnologia de socket, estudo da linguagem Java e Python, análise e definição da arquitetura da aplicação no modelo cliente/servidor, elaboração do protocolo de comunicação entre cliente/servidor, desenvolvimento da programação do projeto.

4. Modelo Cliente/Servidor

Considerado o modelo mais utilizado pelas aplicações distribuídas em redes de computadores (Tanenbaum, 1997), a tecnologia cliente/servidor é uma arquitetura na qual o processamento da informação é dividido em módulos ou processos distintos. Um processo é responsável pela manutenção da informação (servidores) e os outros responsáveis pela obtenção dos dados (clientes). Por definição, o processo cliente envia uma solicitação para o servidor, que por sua vez processa e envia os resultados para o cliente. (Tanenbaum, 1997)

5. Socket

Visando possibilitar a comunicação entre computadores utilizando rede, a tecnologia de socket foi utilizada como base para o desenvolvimento do jogo. Um socket é um ponto final de um fluxo de comunicação entre processos

através de uma rede de computadores. Atualmente, essa comunicação é baseada no Protocolo de Internet ou IP (Internet Protocol).

Cada socket possui um número que consiste nesse endereço de IP do host mais um número de 16 bits local para esse host, ou seja, o número de uma porta do protocolo de transporte (neste caso, TCP). A comunicação é estabelecida entre a conexão de um socket da máquina transmissora e um socket da máquina receptora. (Tanenbaum, 1997)

6. Linguagem Python

Entre as linguagens dinâmicas, o Python pode ser destacado como uma das mais populares e poderosas (Borges, 2010). Python foi a linguagem selecionada para a realização da parte de programação do projeto, por ser uma linguagem simples e oferecer um ambiente novo para a utilização do socket TCP.

7. Linguagem Java

Java é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems (Deitel, 2005). Diferente das linguagens de programação convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é interpretado por uma máquina virtual (Java Virtual Machine, mais conhecida pela sua abreviação JVM).

8. Conclusão

Através do desenvolvimento do projeto, foi possível aprofundar mais o conhecimento sobre sistemas operacionais, incluindo uma melhor visão sobre seu vasto conteúdo, bem como o aprendizado da linguagem python, utilizando a IDE PyCharm, e Java, utilizando a IDE Eclipse.

9. Referências

Tanenbaum, Andrew S. (1997) “Redes de computadores”. Rio de Janeiro, Campus;

Borges, Luiz Eduardo. Python para Desenvolvedores. 2. ed. Rio de Janeiro, 2010;

Harvey M. Deitel (2005). *Java: Como Programar* 6 ed. São Paulo: Pearson education do Brasil.