Aluno: Rafael Eiki Matheus Imamura

RA: 176127

Matéria: MC833 – Programação de Redes de Computadores

Professor: Edmundo Roberto Mauro Madeira

Projeto 2 – TCP VS UDP

# Introdução

Neste projeto, foi desenvolvido um sistema que representa um gerenciador de perfis de usuários, contendo informações como habilidades e experiência. O sistema é composto por um servidor e um cliente TCP. Foram realizados testes em computadores distintos na mesma rede para conseguir avaliar o tempo gasto na rede, sendo esses valores comparados com o tempo de análise.

# Sistema: descrição geral e casos de uso

O sistema possui 6 funcionalidades que lidam com o gerenciamento de usuários com dados de um currículo e uma de controle. Os números a seguir indicam o identificador da operação.

1. Dado um nome de um curso, retornar todas as pessoas que se formaram naquele curso;
2. Dada uma cidade, retornar todas as habilidades das pessoas daquela cidade;
3. Dado um email e uma experiência, adicionar a experiência no perfil do dono do email;
4. Dado um email, as experiências de um perfil específico;
5. Retornar todas as informações de todos os perfis;
6. Dado um email, retornar todas as informações de um perfil;
7. O cliente se desliga do servidor.

Para rodar o sistema, o usuário deve:

1. Entrar no diretório “server”;
2. “make”;
3. “./server”;
4. Entrar no diretório “client”;
5. “make”;
6. “./client IP\_DO\_SERVIDOR” => O IP do servidor é visível no terminal do servidor.

# Armazenamento e estrutura de dados do servidor

No servidor, os dados são armazenados em binário em um arquivo. O arquivo guarda os registros de forma sequencial no arquivo “database.bin”. Este arquivo é criado ao subir o servidor. Os dados iniciais lá colocados estão no arquivo “database\_seed.c”.

Detalhes da implementação do servidor TCP

O tempo todo que um cliente se conecta, o servidor permanece com um socket aberto para ele. O limite do servidor é de 10 conexões simultâneas. Para cada conexão aberta, um novo processo filho é criado para cuidar dele.

Ao enviar uma requisição para o servidor, o cliente envia um código que representa uma das operações possíveis. Seguido disso, o cliente pode enviar informações adicionais necessárias para a operação (tal como email ou nova experiência do usuário). Ao retornar, o servidor envia a quantidade de caracteres que a mensagem vai ter em um inteiro, seguido da mensagem em si. No caso de enviar uma imagem, o servidor envia o tamanho da string do path da image, o path, o tamanho do arquivo da imagem e então a imagem em si (dados).

Cada vez que um perfil completo é enviado, seja em listas ou individualmente, o servidor envia o arquivo de foto do perfil. O cliente recebe os arquivos lendo os bytes repetidamente até se esgotar o tamanho do arquivo, enviado anteriormente.

# Tempos de comunicação

Para cada operação do servidor, foram realizadas 20 chamadas de testes. Os tempos resultantes das respostas do servidor e da rede podem ser visualizados nos Gráficos 1 a 6, abaixo. O cenário era de 2 computadores em LAN. As médias apresentadas têm o intervalo de confiança de 95%. O sistema estava com 16 registros.

**Gráfico 1 – Tempo da operação 1 em μs**

Tempo médio do servidor: 1,38 ± 0,15 ms

Tempo médio da rede: 30,25 ± 0,47 ms

**Gráfico 2 – Tempo da operação 2 em μs**

Tempo médio do servidor: 1,23 ± 0,13 ms

Tempo médio da rede: 1,61 ± 0,17 ms

**Gráfico 3 – Tempo da operação 3 em μs**

Tempo médio do servidor: 4,27 ± 0,39 ms

Tempo médio da rede: 1,37 ± 0,12 ms

**Gráfico 4 – Tempo da operação 4 em μs**

Tempo médio do servidor: 1,27 ± 0,08 ms

Tempo médio da rede: 1,48 ± 0,02 ms

**Gráfico 5 – Tempo da operação 5 em μs**

Tempo médio do servidor: 3,45 ± 0,65 ms

Tempo médio da rede: 172,83 ± 7,23 ms

**Gráfico 6 – Tempo da operação 6 em μs**

Tempo médio do servidor: 1,49905 ± 0,16 ms

Tempo médio da rede: 1,00925 ± 0,05 ms

A quantidade de informações retornada por operação foi:

1. 2 perfis (inclui imagens);
2. 4 experiências de perfis;
3. Frase indicando o estado da operação;
4. Experiência de um perfil;
5. Todos os perfis (16, incluindo imagens);
6. Um perfil (inclui imagens).

As imagens tinham 4 pesos: 36KB, 37KB, 55KB e 139KB.

Dado esse tempo de comunicação, é notável como o tempo na rede é muito superior ao tempo de processamento no servidor, especialmente em casos de maior volume de dados, como a operação 5. A operação 3 é a única que demora mais no servidor do que na rede, por retornar pouca informação para o cliente e ter que fazer escrita (atualização de registro).

# Conclusão

O desenvolvimento de sistemas distribuídos, mesmo que simples, envolvem diversas dificuldades. A definição do protocolo, possíveis erros na rede são alguns deles. O uso do protocolo TCP traz vantagens por dar garantia de ordem dos pacotes e de sua chegada, fazendo com que o programador precise se preocupar menos.

No cenário dado, a perda de pacotes é baixa, fazendo com que o TCP seja uma escolha fácil. É notável como a maior parte do tempo é gasta em rede e não em processamento, outra questão que deve ser levada em conta ao criar um servidor TCP.

# Referências

MADEIRA, E. R. M. “Socket TCP / MC833”. Notas de aula. Acesso em: mar. 2019.