

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 UDP. 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 e com os tempos do projeto 1, que representa o servidor com as mesmas funcionalidades, mas usando TCP.

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 UDP, o usuário deve:

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

As especificações para a utilização do servidor TCP são as mesmas do trabalho 1.

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. As conexões são respondidas serialmente, uma por vez.

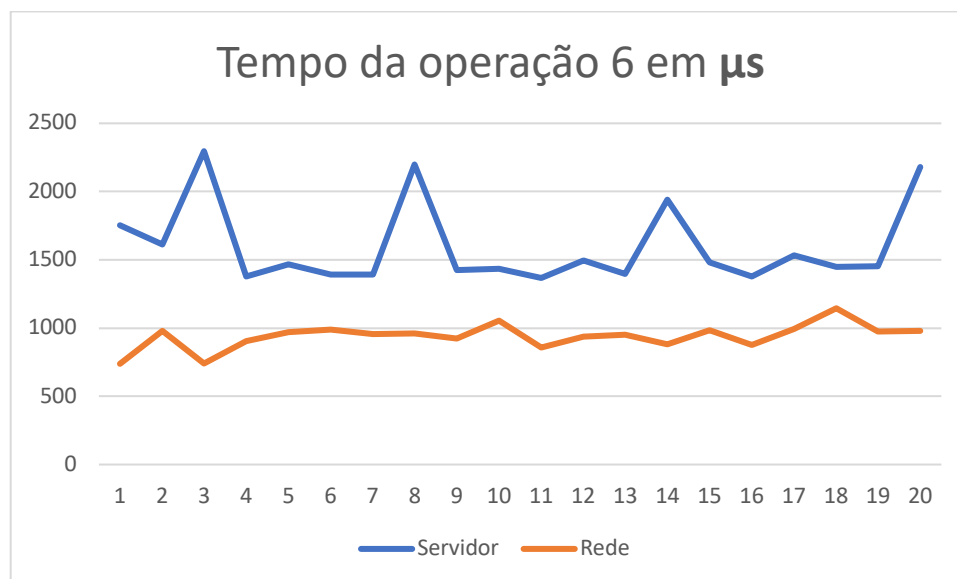
Ao enviar uma requisição para o servidor, o cliente envia uma mensagem com um código da operação e informações adicionais necessárias para a operação (tal como email ou nova experiência do usuário). Ao retornar, o servidor envia a quantidade de caracteres que a mensagem vai ter em um inteiro, seguido 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

Foram realizadas 20 chamadas de testes da operação 6 – informações do perfil e um arquivo de foto -, como foi especificado no enunciado do trabalho. Os tempos resultantes das respostas do servidor e da rede podem ser visualizados no Gráficos 1, que foram medidos calculando o tempo total do cliente e subtraindo-se o tempo de resposta no servidor. O cenário era de 2 computadores em LAN. As médias apresentadas têm o intervalo de confiança de 95%. O sistema estava com 16 registros.

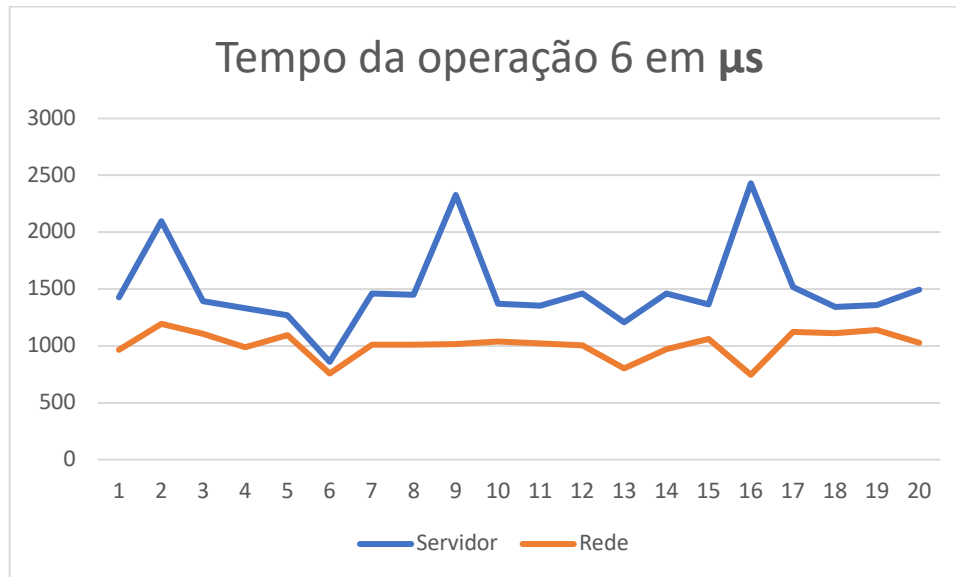
Gráfico 1 – Tempo da operação 6 em μ s (servidor UDP)



Tempo médio do servidor: $1,60 \pm 0,13$ ms

Tempo médio da rede: $0,93 \pm 0,04$ ms

Gráfico 2 – Tempo da operação 6 em μ s (servidor TCP)



Tempo médio do servidor: $1,50 \pm 0,16$ ms

Tempo médio da rede: $1,01 \pm 0,05$ ms

As imagens dos perfis enviados tinham 4 pesos: 36KB, 37KB, 55KB e 139KB. O tempo de operação mostrado nessa operação não inclui a imagem de 139KB, que é significativamente maior.

O tempo da rede e servidor se aproximam em ambos os casos, sendo o tempo no servidor provavelmente maior pois é necessário procurar dentro do arquivo. O tempo médio da rede no UDP foi ligeiramente inferior ao TCP, o que era esperado, mas o tempo médio do servidor foi maior. A diferença pequena se deve ao tamanho pequeno dos dados comparado a casos reais de uso de aplicações UDP. Isso pode ter ocorrido pela forma como os pacotes foram quebrados (1300 bytes cada, no máximo) no caso do UDP. O TCP já realiza a segmentação automaticamente.

Para a imagem maior (139KB), o servidor UDP teve problemas em enviar os pacotes na rede, faltando 1 pacote em vários casos. Esses casos não foram representados na amostra de dados para fazer uma comparação mais igualitária do TCP e UDP.

Conclusão

Desenvolver o mesmo sistema, mesmo tendo o TCP como base, não foi uma tarefa simples como era esperado. O UDP necessita de alguns cuidados a mais do que o TCP, para gerar os pacotes, tomar cuidados com os tamanhos das mensagens, falta de conexão e perda de dados. No caso deste projeto, este último não era um requisito, mas para determinadas aplicações pode ser necessário maior tratamento para a perda de pacotes.

De forma geral, o TCP parece ser uma melhor escolha para esse problema. Visto que não é necessária uma atualização em tempo real e a confiabilidade dos dados é importante, especialmente para o arquivo (caso falte uma parte ele não faz sentido), o UDP não traz grandes vantagens no contexto.

Referências

MADEIRA, E. R. M. "Sockets UDP / MC833". Notas de aula. Acesso em: mai. 2019.