

Documentação Do Sistema Distribuído com Comunicação Cliente-Servidor

Orientado pelo professor Felipe Silva da materia Sistemas Distribuídos

Náthaly de Jesus Brito

Santo Antônio de Jesus – BA, dezembro de 2024

Descrição do processo -

Aplicar os conceitos de sistemas distribuídos, modelo cliente-servidor, e middlewares (RPC/gRPC) para implementar um sistema que simula um serviço de consulta a um catálogo de produtos distribuído.

Requisitos -

- desenvolver um sistema distribuído que permita consultar informações de produtos armazenados em um servidor.
- O servidor armazena um catálogo de produtos (nome, preço, e quantidade em estoque).
- O cliente pode realizar as seguintes operações: o Consultar produtos pelo nome; Consultar produtos com preço inferior a um valor fornecido e Atualizar o estoque de um produto específico.
- Utilização de Sockets para estabelecer a comunicação entre cliente e servidor e RPC/gRPC para realizar chamadas remotas.
- O middleware escolhido deve ser configurado adequadamente para garantir a interação entre cliente e servidor.
- O cliente deve ser capaz de se conectar remotamente ao servidor, realizar as operações e receber as respostas.

Descrição da execução -

1. Implementação com Sockets - implementação de um servidor que gerencia catálogo de produtos, e cliente que envia solicitações para consulta ou atualização.
2. Transferência do Sistema em Sockets para gRPC.
3. Middleware utilizado: gRPC (Remote Procedure Call)

Descrição geral do sistema com Socket -

1. Servidor:
 - Inicializar um socket no lado do servidor e vinculá-lo a uma porta.
 - Criação de estrutura de dados para o catálogo.
 - Escutar conexões de cliente.
 - Processar solicitações recebidas (consultar produto pelo nome, consultar produtos com preço inferior a um valor, atualizar a quantidade de estoque de um produto).

- Retornar resposta para o cliente.
2. Cliente:
- Inicializar o socket e conectar ao servidor.
 - Exibir opções de menu para o usuário (consultar produtor por nome, consultar produto com preço inferior a um valor, atualizar estoque).
 - Enviar a solicitação ao servidor e exibir a resposta.

Descrição geral do sistema com gRPC -

1. Criação de arquivo *.proto* com definição do serviço (o arquivo *.proto* descreve as mensagens - requisições/respostas - e os serviços).
2. Utilização do *grpc_tools* (biblioteca do python).
3. Implementação do servidor e cliente com as funções baseadas no *.proto*.
4. Comando utilizado para gerar os arquivos Python do gRPC: `python -m grpc_tools.protoc -I. --python_out=. --grpc_python_out=. catalogo.proto`
5. Esse comando gerará dois arquivos:
 - `catalogo_pb2.py`: define as mensagens.
 - `catalogo_pb2_grpc.py`: define o serviço e a base para implementação.

Dicionário -

1. **Middleware** - são componentes localizados em diferentes dispositivos de uma rede que se comunicam e coordenam suas ações para alcançar um objetivo comum. O objetivo é garantir escalabilidade, alta disponibilidade e desempenho. Como software intermediário, facilita a comunicação e interação entre as aplicações distribuídas.
2. **Modelo Cliente-Servidor** – trata-se de um modelo em que há um cliente (que faz requisições) e um servidor (processa as requisições e retorna resultado). O cliente geralmente é a interface do usuário e o servidor lida com a lógica e armazenamento.
3. **Socket**: é uma conexão orientada a conexão que permite a comunicação bidirecional entre dois dispositivos.
4. **RPC (Remote Procedure Call)**: permite que um programa chame um procedimento em outro sistema, como se fosse uma chamada de função local. É usado para otimizar a comunicação entre processos em uma rede.
5. **GRPC (Google Remote Procedure Call)**: é uma implementação do modelo RPC que utiliza *proto buffers* para definição das funcionalidades e comunicação via o protocolo HTTP/2.