

Sistema de Processamento de Imagem em Preto e Branco

Descrição | Objetivo | Estrutura do Projeto | Requisitos | Como usar
| Configuração | Licença

Info:

Professor: Dr. Aldo Henrique Dias Mendes

Participantes: Lucas R Silva, Luan Medrado

Disciplina: PROGRAMACAO CONCORRENTE E DISTRIBUIDA

Semestre: 2024/1

Descrição:

- Este sistema permite enviar imagens para um servidor via WebSocket, convertê-las para preto e branco usando XML-RPC e receber as imagens processadas de volta.
- Ele foi projetado para lidar com várias imagens simultaneamente, aproveitando o paralelismo para otimizar o desempenho.

****todos os servidores foram criados no repl.it**

Objetivos:

Entender o melhor funcionamento de um projeto distribuído usando threads e RPC e melhorar o desempenho final.

Estrutura do Projeto:

O sistema consiste em três componentes principais:

- Cliente (cliente.py): Responsável por enviar as imagens para o servidor WebSocket. Ele usa uma abordagem assíncrona e multithreading para enviar várias imagens em paralelo, acelerando o processo.
- Servidor RPC (rpc_img.py): Implementa uma função RPC (preto_branco) que recebe uma imagem codificada em Base64, converte-a para preto e branco e retorna a imagem processada, também em Base64.
- Servidor WebSocket (websocket.py): Recebe as imagens do cliente, encaminha-as para o servidor RPC para processamento e envia as imagens em preto e branco de volta para o cliente.

Requisitos:

- Python 3.x

- Bibliotecas: `asyncio`, `websockets`, `base64`, `PIL`, `io`, `time`, `concurrent.futures`, `xmlrpc.server`, `xmlrpc.client`, `logging` (já incluídas no código)

Como usar:

- Inicie o servidor RPC: Execute o arquivo `rpc_img.py`. Ele iniciará um servidor RPC na porta 8000.
- Inicie o servidor WebSocket: Execute o arquivo `websocket.py`. Ele iniciará um servidor WebSocket na porta 8001.
- Execute o cliente: Execute o arquivo `cliente.py`. Ele enviará as imagens listadas em `imagens_paths` para o servidor, que as processará e retornará as versões em preto e branco. As imagens processadas serão salvas na pasta `out`.

**** Serve no repl.it ****

Configuração:

- Caminhos das imagens: A lista `imagens_paths` em `cliente.py` contém os caminhos das imagens a serem enviadas. Modifique-a para incluir as imagens que você deseja processar.
- URL do servidor WebSocket: A variável `url_servidor` em `cliente.py` deve apontar para o endereço do servidor WebSocket.
- Certifique-se de que ele esteja correto.
- Endereços dos servidores: Os scripts `rpc_img.py` e `websocket.py` estão configurados para rodar em "0.0.0.0". Se os servidores estiverem em endereços diferentes, ajuste as configurações nos scripts.

© Licença:

Este repositório está licenciado sob a licença MIT.

License