

TP2 - Computação Distribuída

Grupo 16

- Diogo Carichas - 49473
 - Alexandre Severino - 49457
-

Introdução

Este documento descreve o funcionamento de um sistema distribuído desenvolvido para a gestão de imagens, permitindo o upload e download de ficheiros de imagem com a funcionalidade de marcação de imagens com palavras-chave. O sistema é composto por vários componentes interligados que utilizam comunicação gRPC, armazenamento partilhado (GlusterFS), filas RabbitMQ para processamento assíncrono e ordenado de mensagens, e o grupo Spread para coordenação e consistência entre os servidores.

Objetivo

O sistema foi projetado para:

- Gerir a distribuição de cargas de clientes entre servidores;
 - Permitir o *upload* de imagens que são armazenadas no GlusterFS;
 - Processar imagens carregadas com palavras-chave através de um programa (Worker) conectado ao RabbitMQ;
 - Coordenar servidores e manter estados consistentes utilizando o grupo Spread.
-

Arquitetura do Sistema

O sistema é composto pelos seguintes módulos principais:

1. Registry Server

- Atua como ponto de entrada para os clientes;
- Mantém o estado global dos servidores, incluindo o número de clientes ativos em cada servidor;
- Ao receber uma solicitação de um cliente, redireciona para o servidor com menor carga.

2. Servidores

- Responsáveis por receber *uploads* e *downloads* de imagens;
- Quando recebem um *upload*, armazenam a imagem no GlusterFS e geram um ID associado;
- Envia mensagens para o grupo Spread para atualizar o estado global;
- Publicam mensagens na exchange RabbitMQ com o ID da imagem e palavras-chave associadas;
- O *download* é feito com acesso ao armazenamento partilhado do GlusterFS, através do ID da imagem.

3. GlusterFS

- Sistema de ficheiros distribuído utilizado para armazenar imagens de forma partilhada entre servidores.

4. RabbitMQ

- Utilizado para gestão assíncrona do processamento das imagens.
- Inclui:
 - **Configurador:** Responsável por configurar a exchange, fila de trabalhadores e suas ligações;
 - **Exchange e Filas:** A exchange entrega mensagens para a fila dos trabalhadores, que contém informações sobre as imagens a processar;

5. Worker (Trabalhador)

- Consumidor de mensagens RabbitMQ;
- Processa imagens armazenadas no GlusterFS com base nos IDs recebidos e associa as palavras-chave especificadas;
- Armazena as imagens processadas novamente no GlusterFS e notifica o grupo Spread sobre a conclusão do trabalho.

6. Grupo Spread

- Coordena os servidores do sistema;
- O primeiro servidor (não incluindo o Registry Server) que se junta ao grupo torna-se o líder, mantendo um registo de todas as mensagens recebidas;
- Quando um novo servidor se junta ao grupo, o líder replica o histórico de mensagens para o novo membro para garantir consistência;
- Monitora o número de clientes ativos em cada servidor.

7. Cliente

- Executa pedidos de *upload* e *download* a um servidor obtido através do Registry Server, fornecendo ou uma imagem para carregar, ou o ID da imagem para transferir e a sua respetiva diretoria de armazenamento;

Fluxo de Operação

1. Upload de Imagem

1. O cliente contacta o Registry Server para obter o IP de um servidor disponível;
2. O cliente envia a imagem para o servidor designado;
3. O servidor:
 - Notifica o grupo Spread sobre o estado do cliente;
 - Gera um ID único para a imagem;
 - Armazena a imagem no GlusterFS;
 - Publica uma mensagem na exchange RabbitMQ com o ID da imagem e palavras-chave.
4. O servidor devolve o ID da imagem ao cliente.

2. Processamento da Imagem

1. A mensagem publicada na exchange RabbitMQ é consumida por um Worker;
2. O Worker:
 - Obtém a imagem correspondente ao ID no GlusterFS;
 - Processa a imagem (marcação com palavras-chave);
 - Armazena a imagem processada no GlusterFS;
 - Notifica o grupo Spread sobre a conclusão do processamento.

3. Download de Imagem

1. O cliente contacta o Registry Server para obter o IP de um servidor disponível;
2. O cliente solicita ao servidor a imagem correspondente a um ID;
3. O servidor recupera a imagem do GlusterFS e a envia ao cliente.

Manual de Instalação

Pull das Imagens Docker

Execute os seguintes comandos para obter as imagens Docker necessárias:

```
# Registry Server
docker pull cdg16/registry

# Servidores svc
docker pull cdg16/svc

# RabbitMQ Configurator
docker pull cdg16/rabbitconfig
```

Execução dos Containers

1. Registry Server

```
docker run -d -p <port>:<port> --name registry cdg16/registry <port>
<daemonIP> <spreadGroup>
```

- `port` : Porta do servidor (ex.: 8000);
- `daemonIP` : Endereço IP do daemon Spread;
- `spreadGroup` : Nome do grupo Spread.

2. Servidor svc

```
docker run -d -p <svcPort>:<svcPort> --name <svcName> -v
/var/sharedfiles:/mnt/sharedfs cdg16/svc
<svcPort> <daemonIP> <spreadUserName> <spreadGroup>
<IP_BROKER> <exchangeName> <routingKey>
```

- `svcPort` : Porta do servidor (ex.: 9001);
- `daemonIP` : Endereço IP do daemon Spread;
- `spreadUserName` : Nome de utilizador para o grupo Spread. **Nota:** Todos os nomes de utilizador devem começar com "svc" devido à lógica de *parsing* JSON

- aplicacional;
- `spreadGroup` : Nome do grupo Spread;
- `IP_BROKER` : Endereço IP do RabbitMQ;
- `exchangeName` : Nome da exchange RabbitMQ;
- `routingKey` : Chave de encaminhamento para a fila RabbitMQ.

3. RabbitMQ Configurator

```
docker run -d --name rabbitconfig cdg16/rabbitconfig  
<IP_BROKER> <RABBIT_HOST> <EXCHANGE_NAME>  
<QUEUE_NAME> <ROUTING_KEY>
```

- `IP_BROKER` : Endereço IP do RabbitMQ;
 - `RABBIT_HOST` : Endereço do host RabbitMQ;
 - `EXCHANGE_NAME` : Nome da exchange RabbitMQ;
 - `QUEUE_NAME` : Nome da fila RabbitMQ;
 - `ROUTING_KEY` : Chave de encaminhamento para a fila RabbitMQ.
-

4. Worker

```
java -jar /var/sharedfiles/worker.jar <WORKER_NAME> <daemonIP> <IP_BROKER>  
<QUEUE_NAME> <spreadGroup>
```

- `WORKER_NAME` : Nome de utilizador para o grupo Spread;
 - `daemonIP` : Endereço IP do daemon Spread;
 - `IP_BROKER` : Endereço IP do RabbitMQ;
 - `QUEUE_NAME` : Nome da fila RabbitMQ;
 - `spreadGroup` : Nome do grupo Spread.
-

A aplicação Worker assume a execução dentro de uma *VM Google Cloud Console* com acesso à diretoria partilhada de GlusterFS

5. Client

```
java -jar client.jar <registryPort> <registryAddress>
```

- `registryPort` : Porta do servidor Register;
 - `registryAddress` : Endereço IP do servidor Register.
-