
Projecto Computação Distribuída

Licenciatura em Engenharia Informática - Computação Distribuída

Distributed Photo Collage

Docentes:

- Diogo Gomes (dgomes@ua.pt)
- Nuno Lau (nunolau@ua.pt)
- João Rodrigues (jmr@ua.pt)

Prazo:

22 de Julho – 10h00

Conceitos a abordar:

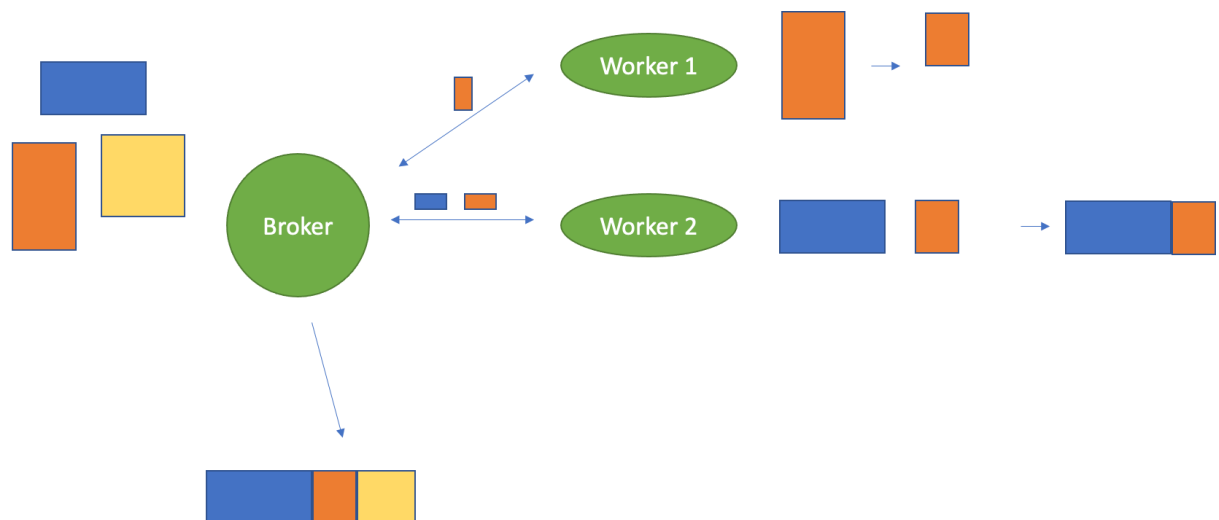
- Sockets
- Marshalling
- Broker based Architecture
- Fault Tolerance

Introdução

Um dos amigos aveirenses do projecto da época normal decidiu fazer uma colagem com todas as fotos dos amigos. Começou por fazer download de todas as fotos para o seu computador e pretende agora colar todas as fotos criando uma grande imagem em que as fotos estão todas numa única tira. Esta tarefa é muito intensiva do ponto de vista computacional pelo que encomendou aos alunos de CD um sistema distribuído para realizar a tarefa nuns computadores velhos que têm na garagem e que volta e meia se desligam sozinhos.

O sistema distribuído a realizar neste projecto irá assentar numa arquitectura baseada no algoritmo Map-Reduce [1]. Deverá ser utilizado um broker centralizado e um conjunto de processos independentes (workers), distribuídos eventualmente por diversos computadores, capazes de: reduzir todas as imagens a uma altura comum, reduzir o número de fotos de forma iterativa colando 2 fotos de cada vez até ficar apenas 1 única foto que é uma tira gigante.

Para esta tarefa deverão desenvolver todo o código necessário para analisar uma pasta que contém ficheiros de imagem em vários formatos e tamanhos, distribuir por diversos processos dois tipos de tarefas (reduzir imagem e colar 2 imagens) e, finalmente, gravar para disco e apresentar uma imagem final composta por todas as imagens que residiam na pasta (ver diagrama).



Para esta tarefa é fornecido este guião e um conjunto de imagens disponíveis em [3]. Devem criar o vosso projecto no GitHub Classroom usando <https://classroom.github.com/a/iqU6003t>.

Objectivos

Este é um trabalho aberto em que terão que desenvolver um **protocolo de mensagens** que permita ao broker distribuir o trabalho pelos diversos workers que podem residir noutros computadores. O broker deve iniciar sem qualquer worker ligado. À medida que novos workers se ligam, o broker deve distribuir trabalho pelos mesmos. Não existem workers especializados. Um worker tanto pode redimensionar a imagem como fazer colagens (e retornar a nova imagem). O broker é responsável por garantir que no final existe apenas 1 imagem que é a colagem de todas as fotos inicialmente disponibilizadas, ordenadas por data de última modificação [4].

Para o utilizador deste sistema, os workers deverão incluir funcionalidades de:

- Redimensionar imagens
- Criar uma nova imagem a partir de 2 imagens (cada worker só pode colar 2 imagens/ficheiros de cada vez)

Deverão existir 2 scripts "executáveis": broker.py e o worker.py que permite executar estas operações.

Como os workers vão correr em computadores velhos pouco fiáveis, o sistema tem que tolerar a falha dos workers que podem "morrer" a qualquer instante ou levar tempos infinitos a realizar uma operação.

Funcionamento

O broker.py deve iniciar com 2 argumentos: nome de pasta com as imagens iniciais do peer e altura da imagem da colagem.

O broker.py deve analisar a pasta e aguardar por workers disponíveis para realizar as operações de redimensionamento e colagem.

O worker.py deve se ligar ao broker (endereço fornecido por argumento) e receber a operação e o(s) ficheiro(s) necessário(s) para realizar a operação via **socket UDP**.

Podem existir um número indeterminado de workers e não podem ser especializados numa única operação (ao longo do seu ciclo de vida, pelo menos 1 worker deve realizar ambas as operações: redimensionamento e colagem)

Tanto o broker como os workers devem imprimir para o ecrã a tarefa comunicada entre si, Exemplos:

[BROKER] Enviar capela.jpg para redimensionar no worker 127.0.0.1:32145
[BROKER] Recepcionado capela.jpg após redimensionamento no worker 127.0.0.1:32145
[BROKER] Enviar capela.jpg e rossio.jpg para colagem no worker 127.0.0.1:32145
[BROKER] Recepcionado 1.jpg após colagem no worker 127.0.0.1:32145

— // —

[WORKER] Recepcionado capela.jpg
[WORKER] Redimensionamento completado
[WORKER] Recepcionado capela.jpg e rossio.jpg para colagem
[WORKER] Colagem completada

No final (quando a colagem é terminada) o Broker deve desligar os workers e imprimir as seguintes estatísticas:

- número de redimensionamentos totais
- número de colagens totais
- número de redimensionamentos por worker (média)
- número de colagens por worker (média)
- tempo mínimo, médio e máximo para fazer um redimensionamento (tempo é medido pelo broker, medindo tempo decorrido entre envio e recepção)
- tempo mínimo, médio e máximo para fazer uma colagem (tempo é medido pelo broker, medindo tempo decorrido entre envio e recepção)
- lista de workers e operações realizadas por cada um dos workers
- tempo total desde a primeira atribuição de tarefa a 1 worker até colagem final

Para simular os computadores velhos deve matar aleatoriamente processos worker e colocar sleeps random de vários minutos após o código do worker que realiza a operação.

Avaliação

A avaliação deste trabalho será feita através da submissão de código na plataforma GitHub classroom e de um relatório em formato PDF com não mais de 5 páginas colocado no mesmo repositório junto com o código e com o nome **relatorio.pdf**.

Está em avaliação o protocolo definido e documentado, assim como as *features* implementadas de acordo com os objetivos incrementais:

- Comunicação cliente/servidor
- Transferência de imagem entre nós
- Distribuição balanceada de operações
- Eficiência de processamento
 - Redução efectiva do tempo necessário a fazer uma colagem em função do número de workers
 - Todos workers devem poder realizar todas as operações
- Resiliência a falhas
 - Caso um dos workers se desligue, o sistema deve assegurar que não há perda de fotos

Referências

- [1] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1 (January 2008), 107–113. <https://doi.org/10.1145/1327452.1327492>
- [2] <https://pillow.readthedocs.io/en/stable/>
- [3] <https://filesender.fccn.pt/?s=download&token=39d83675-2f3c-4d2a-b65c-cf36099132b0>
- [4] <https://docs.python.org/3/library/os.html#os.stat>