

Relatório Distributed Photo Collage

Objetivo

O projeto consiste na construção de uma rede distribuída, onde o objetivo é a construção de uma imagem constituída pelas imagens de um dado conjunto onde a ordem destas é determinada pela data de modificação. Para isso, existe o *broker* que vai receber todas as imagens colocando-as pela ordem correta e começa a distribuir o trabalho pelos *workers*. Os *workers* estão encarregues do redimensionamento das imagens para cumprirem a altura pretendida, que foi definida ao iniciar o *broker* e da colagem de duas imagens.

Abordagem

Para começar, foram definidos os scripts de arranque tanto do *broker* como dos *workers*, que permitem a passagem de argumentos essenciais para o funcionamento dos mesmos. De seguida, iniciou-se o desenvolvimento das classes **Protocol**, **Broker** e **Worker**.

Classes

Protocolo

Um dos requisitos definidos era o uso de sockets UDP. Devido à falta de controlo da informação enviada por este tipo de sockets, foi necessário criar um protocolo, capaz de lidar com esta imprevisibilidade. Para isso foi definido a estrutura do corpo das mensagens enviadas dentro deste sistema, onde cada mensagem era constituída por tamanho variável, obviamente havendo limitação de tamanho máximo. Cada mensagem contém 8 bytes para definir o tamanho da mensagem, 1 byte para definir o tipo de mensagem que vai ser enviada, onde 1 é *acknowledgment*, 2 *acknowledgment response* e 5 de dados, 1 byte de uma hash para identificar o endereço que enviou, 2 bytes de counter para saber a ordem dos pacotes, visto que estes podem chegar pela ordem errada e por fim a mensagem em si a ser enviada.

Broker

O Broker começa por inicializar todas as variáveis necessárias, por carregar as imagens da pasta e por limpar a pasta *temp*, pasta onde são guardadas o resultado das operações intermédias.

Posteriormente, o broker fica à espera de mensagens dos *workers*, ao receber uma mensagem, analisa se a mensagem ainda é válida ou se já passou o tempo de tolerância, 10 segundos. Caso passe do tempo assume-se que o *worker* morreu e é removido da lista de *workers*. Depois de validar a mensagem, este verifica se algum *worker* quer entrar com a mensagem JW. Depois, começa a tentar delegar trabalhos ao verificar se existem imagens na lista de imagens carregadas e se houver começa a delegar trabalho para um *worker* que esteja disponível (Ready) enviando uma mensagem RSZ. Ao receber uma mensagem RDONE, significa que um redimensionamento foi efetuado com sucesso e a imagem recebida tem de ser adicionada às imagens editadas para serem posteriormente coladas. Quando não houver mais nenhuma imagem para ser redimensionada dá-se início ao processo de colagem das imagens. Para isso são enviadas duas imagens para um *worker* através da mensagem STC.

Um problema que surge com a colagem das imagens com vários *workers* a funcionar em simultâneo, é a perda de noção de ordem. Para resolver este problema nas mensagens de colagem também é enviado o tempo menor entre as duas imagens de modo a garantir que a ordem é mantida. Ao receber uma mensagem SDONE o *broker* vai verificar se ainda existem imagens para serem coladas e caso isso se verifique, envia uma mensagem STC, se só houver uma imagem quer dizer que a imagem final está pronta e envia a todos os *workers* uma mensagem para se desligarem e por fim mostra as estatísticas adquiridas durante a execução do mesmo.

Worker

O Worker começa por receber o seu endereço e o endereço do broker e envia uma mensagem para o broker uma mensagem JW, para se poder ligar à rede. Posteriormente, começa a ler mensagens e dependendo do tipo de mensagens irá realizar a tarefa atribuída.

No caso de a mensagem ser de RSZ, a imagem é lida da mensagem e é calculado o rácio da imagem para poder efetuar o redimensionamento da imagem e enviar para o *broker*.

Caso seja de colagem, STC, as duas imagens são lidas e é criada uma imagem com o comprimento igual à soma do comprimento das duas imagens e

as duas são coladas, a primeira a partir do início da nova imagem e a segunda a partir do pixel a seguir ao último usado pela primeira imagem, enviando o resultado para o *broker*.

Por último, se for QUIT ele termina tudo e desliga-se.

Técnicas usadas e Redundância

Para guardar as imagens, começou-se por tentar guardar as imagens em JPEG, devido à elevada compressão do formato, mas devido às limitações de comprimento da imagem, este formato foi trocado pelo PNG que apesar de mais lento, permite um comprimento muito maior.

Para resolver o problema de um dos *workers* se apoderar do trabalho todo, estes passaram a ter estados (*dead*, *ready*, *working*) de maneira a conseguir distribuir o trabalho de maneira mais justa.

Para tratar da redundância, foi preciso criar duas novas listas de maneira a perceber quais eram as imagens que estão a ser trabalhadas e caso um dos *workers* falham o trabalho é atribuído a outro de maneira a prevenir a perda de informação.

Conclusões

Com este trabalho foi possível consolidar conhecimentos sobre as sockets UDP e sobre a necessidade de tratamento das mensagens e das falhas advindas. Deu para perceber o funcionamento e as vantagens e desvantagens da arquitetura baseada num Broker.