

Projeto 2

Distributed Object Detection

Diogo Gomes, Nuno Lau

Este documento complementa o guião já publicado do Projeto 2 de Computação Distribuída. Não pretende ser uma lista exaustiva dos requisitos deste trabalho, mas pretende chamar a atenção para alguns pontos importantes (e que serão avaliados) e também informar sobre algumas funcionalidades que podem usar sem qualquer penalização.

Os trabalhos submetidos devem cumprir os seguintes requisitos:

- O *upload* do ficheiro de vídeo para o servidor deve ser realizado através de um *endpoint* na raiz do servidor.
- O cliente que faz o *upload* do vídeo não deve esperar que o vídeo seja totalmente processado, basta esperar que o *upload* do vídeo tenha sido concretizado para o servidor.
- O servidor pode gravar ficheiros no disco mas não pode passar informação ou dados para os *workers* através do disco. O servidor e os *workers* não podem partilhar ficheiros no sistema de ficheiros.
- Deve ser possível fazer upload de dois (ou mais) vídeos distintos mas que têm o mesmo nome. Ambos os vídeos devem ser processados nesse caso.
- O processamento das *frames* de um vídeo pode ser realizado por qualquer ordem. Os alarmes não têm de ser emitidos por ordem crescente do id da *frame*. No entanto, todas as *frames* têm de ser processadas e todos os alarmes correspondentes impressos na consola.
- Os *requests* HTTP são tratados pelo Flask através de *threads* distintas. O código deve garantir que não existem condições de corrida durante a execução das diferentes *threads*. É de notar que a simples atualização de um contador partilhado por várias *threads* pode levar a condições de corrida.
- O servidor deve ser resistente a *omission failures*, i.e. situações em que um (ou mais) *worker(s)* não responde(m) ao pedido de tratamento de uma *frame*, mas continua(m) ativo(s)
- Serão permitidas situações de *busy waiting* enquanto um vídeo é processado, embora sejam preferidas soluções que não entrem em *busy waiting*.*
- A avaliação será realizada usando o *development server*. No entanto, devem saber que, numa versão de produção, o uso de variáveis globais pode trazer vários problemas pois, por exemplo, o serviço pode ser providenciado por vários processos independentes que não partilham essas variáveis.*
- Para facilitar a avaliação independente do projecto devem criar um ficheiro “run.sh” que execute o servidor, workers e qualquer *daemon* necessário e realize o processamento

* Estes pontos apenas poderão penalizar trabalhos com notas superiores a 16.

do vídeo `moliceiro.m4v`. O script deverá executar sem qualquer parâmetro para um cenário com 2 workers.