Licenciatura em Engenharia Informática - Computação Distribuída

Docentes:
Prof. Diogo Gomes & Prof. Nuno Lau

Projeto 2
Distributed Object Detection

Rui Fernandes, 92952

DETI
Universidade de Aveiro

04 – 2020

1 Introduction

The goal of this project was to develop a distributed system that handles the detection of objects in a vídeo. To accomplish this the following technologies were used: Flask, Redis and Celery.

2 Technologies

Flask is used to establish the server and allow comunications to occur betwen it and the workers created, it also allows the client to send vídeos he inteds to have processed to the server.

Celery is an asynchronous task queue that allows creation of worker nodes that consequently can execute tasks in a concurrent manner. The server sends these tasks to the queue and celery distributes the tasks among workers so that they can process them in the background.

Redis serves as the message broker that decides which worker should get each task, it allows efficient distribution of a large amount of tasks in the queue.

3 Solution Walk through

1) The server is turned on using the server.py script.
2) A number of celery worker processes are initialized based on the worker.py script.
3) The client requests a video to be processed by using the *curl -F 'video=video_name http://localhost:5000* comand.
4) The server saves some information of the video in its main dictionary struture (that serves as information storage).
5) The server separates the video in image frames and immediately calls the processing celery task for each one.
6) The server is free for new interactions as the requested tasks run in the background.
7) As a worker receives a frame, it processes it and compiles the required information, then send a post request to the server with the information attached in json format.
8) When the server receives a post from a worker it further processes the data received and updates the information storage.
9) After all frames of a video have been processed, the server compiles all the video information and displays it.

4 Problem

Currently processing a frame takes approximately 40 seconds, and even having multiple workers processing different frames at once, the processing of a whole video is extremely cumbersome and unreliably slow. As such the current code is only analyzing the first 32 frames of videos.

The code could definitely be improved a lot to allow better processing times and overall a reliable program.

5 Testing

To test the solution, open a command box, make sure redis has a running server using the *redis-server* command, and then simply use the *./run.sh* command to run the bash script. This will trigger 5 command boxes to appear, 4 celery workers and the server script, a few seconds after the curl command will trigger, sending the video to the server.