AnyVision Dev. Interview Task

You should spend 1-2 hours on this task. Code should be written in python. You may use 3rd party libraries as needed.

**Scenario:** You need to provide a service that can resize images. By service, we mean that independent software client processes should be able can ‘connect’ to the service in some way. These clients may be local (on the same machine), or (for extra points), your service could allow connections from a mixture of both local and remote clients. Note that the service should process images (single video frames), not complete videos.

The service needs to scale so that many clients can use it concurrently. You might consider using message/task system architectures scale your service via distributed processing. It must be possible for multiple requests to be processed concurrently and all computational resources available should be utilised. You should try to make use of the system’s GPUs if possible.

To test your service you should create a test program that uses your service to concurrently resize N instances of a video, where N is passed as a command line argument to the test application. Note we are using N instances of the same video for simplicity, in reality it would be N different videos. The scaling of each video instance should start M seconds after the previous. For example, if N = 2 and M = 10, you test should start using your service to process one instance of the video immediately, and 10 seconds later it should start using your service to process the second instance. The values of N and M will be passed as command line parameters, as will the video path. Split the video into frames and make multiple calls to your service.

The test program should create N sets of output frames, saved to disk as images, e.g.

/tmp/vid-instance1/Frame00001.jpg

/tmp/vid-instance1/Frame00002.jpg

/tmp/vid-instance1/Frame00003.jpg

/tmp/vid-instance2/Frame00001.jpg

/tmp/vid-instance2/Frame00002.jpg

/tmp/vid-instance2/Frame00003.jpg

… and so on.

You should write the re-sized images to disk as soon as they have been processed (as opposed to converting all frames and then writing them to disk at the end). Make your test report runtime metrics including the frame-rate of the conversion (a running average) for each video instance, report when each video instance has been completely converted, and display the total amount of time taken for each instance. Remember, a key requirement is to process all data as quickly as possible.

You will be expected to talk through your approach, explain it’s advantages and limitations, discuss what other options you considered, and what other approaches could be considered in the future.

**Tip:** Focus on all elements of the problem – try not to get bogged down on one particular bit. It is more important to show how you tackled the problem and what solutions you have come up with rather than having 100% running code covering all aspects – although that obviously helps :). Manage your time. Please ask for any clarifications as needed.