SUPINFO Academic Dept.

SUPINFO International University

INSTITUTE OF INFORMATION TECHNOLOGY

Group Projects
M.Sc.1 - Transcode

Contents

2015-2016

Version 1.0
Last update: 8/12/2015
Use: Students/Staff
Author: Samuel CUELLA

TABLE OF CONTENTS

1. Project Overview

Transcode Inc business model is to sell CPU power across the network. They operate a pool of workers that will undertake various tasks submitted by users through a website.

Users submit tasks over the web interface and get the result once completed. Users are charged for each task they submit.

During the launching process, the first tasks available will be audio and video transcoding.

You've selected as the main subcontractor that will undertake the development of the POC.

2. Functional Expression

Users primarily work with transcode through a website. The website allow users to submit transcoding tasks, either by uploading content or providing a URL that directly points to the content. Once the user has selected to task to be performed (encoding raw material to H.264 for example) he gets a quote and has to pay to have his request processed.

The engine will then take care of the task and notify the user when his job is complete.

2.1. Software development

The product is split in two parts: A core that processes conversion jobs and a website that processes and delivers orders.

2.1.1. Core

The application core maintains a queue of jobs to perform. The core can simultaneously process more than one job. Each job is subdivided is smaller pieces, each piece is sent to a worker to be processed.

When the last worker is done, the core put all the pieces together and notifies the user by mail that the result is ready.

2.1.2. Web application

Users use the web application to submit jobs, pay the fee, and get access to the result.

2.1.2.1. User accounts

Users need an account to make use of the website features. However, if they already have a Facebook or Google account, they can automatically link it to their transcode account. If they don't they can create an account with their email address.

User accounts allow to keep up to 10GB of previous conversion results. Those files are only available when the user is logged into his account.

2.1.2.2. Submitting a task

Anyone can begin a task submission. If the user is not logged into his account or doesn't have an account with trasncode, he'll be prompted to create one after the file to convert has been uploaded and the price computed. He'll have to go through the account creation process (or use a Google/Facebook account) before continuing and paying the conversion fee.

Users can ask Transcode to convert a file either by uploading it or by giving a URL (i.e http://my-web-site.com/uploads/vacations.avi) where the resource is. Afterwise, the file format auto-detected and the user just has to select the format he wants transcode to convert the file to.

Note

Transcode can also be used to extract soundtracks from movies. Converting a movie container to a sound only format (avi to mp3 for example) will extract the audio.

The user is then prompted with the price to pay to get the conversion done. He can pay using paypal. The task will be enqueued as soon as the payment has been received.

Once the task is done, the user gets an email and can find the conversion result in his account.

2.1.2.3. Prices

To remain attractive, prices should stay low. The base price is 1 EUR to process 1 hour of video/sound. Feel free to come up with alternative pricing schemes.

2.2. Supporting architecture

The supporting architecture must be reliable and efficient. It has to provide two main features: High availability of the application components, and storage. It must also have an efficient network between the core that orchestrates tasks and the workers that will undertake actual conversion tasks.

It should also be easy to add new workers to the pool to scale up if necessary.

2.2.1. Core cluster

The core application logic orchestrates all submitted tasks. It should be powered by a high-availability cluster that always make the service available. If one node goes down, the service is still available on another node.

2.2.2. Web client cluster

The web client must also run on an high-availability cluster.

2.2.3. Storage

The solution must make use of a SAN. In your POC you can create an iSCSI SAN using heartbeat. The network design should be done accordingly. The storage is a critical part of the setup.

2.2.4. Workers

Workers are doing the actual conversion task. It should be easy to add/remove workers from the pool. They must have a correctly-sized access to the SAN.

3. Deliverables

Students should include the following elements in their final delivery:

3.1. Software development

- A zip archive with the project source code. The source code must also come with the build system used (Project file, autotools...), if any.
- Project documentation, based on the template.
  - Technical documentation explaining your choices and/or implementation choices/details on the following items (at least):
    - Workers orchestration
    - Map/Reduce
- Deployment manual

The first document is an academic document. Address the reader as a teacher, not a client. The last one (deployment manual) should address the reader as a user. These documents can be in French or in English, at your option.

3.2. Supporting architecture

- Network architecture schema, with all components.
- All server configuration files.
- Anything you find relevant.
- Project documentation, based on the template.
  - Technical documentation explaining your choices and/or implementation choices/details on the following items (at least):
    - Networking
    - Clustering
- Step-by-step deployment manual

The first document is an academic document. Address the reader as a teacher, not a client. The last one (deployment manual) should address the reader as a user. These documents can be in French or in English, at your option.

4. Graded Items

The project will be graded as follows, on a 220/200 scale:

- Software development (110 points)
  - Core (60 points)
    - The core has a list of tasks. (5 points)
    - The core can process more than one task at once. (10 points)
    - The core splits a task and send pieces to workers. (20 points)
    - The core concatenates results from workers to create the final result. (20 points)
    - The core notifies/signals when a task has been processed. (5 points)
  - Webclient (40 points)
    - Users can create an account with their email address. (5 points)
    - Users can use their Facebook to create an account. (3 points)
    - Users can use their Google to create an account. (3 points)
    - Users can access their tasks results. (5 points)
    - Users have access to 10GB of previous results. (3 points)
    - Users can submit tasks. (8 points)
    - The webclient pushes the task as soon as it's paid for. (13 points)
  - Bonus (10 points)
    - Bonus features done by the students. (10 points)
- Supporting architecture (110 points)
  - Engine cluster (25 points)
    - The application core is high available and can survive losing nodes. (25 points)
  - Web cluster (25 points)
    - The web client is high available and can survive losing nodes. (25 points)
  - SAN (10 points)

- All storage is SAN-based. (30 points)

- Workers (20 points)
  - It's easy to add/remove workers to the pool. (10 points)
  - Workers have sufficient I/O with the SAN. (10 points)

- Bonus (10 points)
  - Bonus features done by the students. (10 points)