



Mohohan: An On-line Video Transcoding Service via Apache Hadoop

Chun-Han Chen

OgilvyOne Inc.

October 2nd, 2012



Outline

Bio

Introduction

Background

- Hadoop

- AWS S3

- AWS EC2

- AWS EMR

Dataflow

- Overview

- Splitting

- Transcoding

- Merging

Benchmark

Demo

Thanks



Bio

- Chun-Han Chen, Programmer in OgilvyOne inc.,
ringech.chen@ogilvy.com

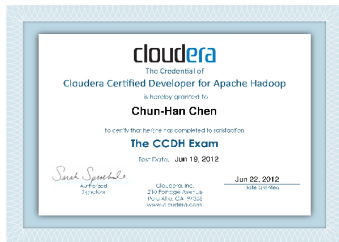


Figure 1 : Certifications



Introduction

- Social apps cause information explosion in **multimedia** way
- Cloud video encoding/transcoding service is needed
- Real-time transcoding is a trade-off between **temporal** and **special** issues



Background: Hadoop

- HDFS: A file system
- MapReduce: A programming framework

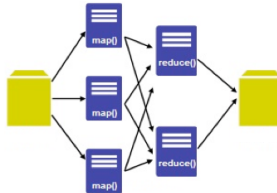


Figure 2 : HDFS and MapReduce



Background: AWS S3

- A distributed **on-line** file system
- A **bucket-item** infrastructure

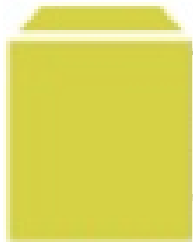


Figure 3 : AWS S3 is another kind of HDFS



Background: AWS EC2

- An on-line **operating system**
- A platform of **physical resources** to run tasks



Figure 4 : AWS EC2 is just a unit in MapReduce



Background: AWS EMR

- A group of EC2s to get/store big data **from S3**
- A group of EC2s to **run MapReduce** with big data

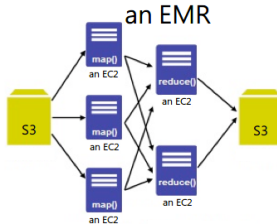


Figure 5 : AWS EMR is an on-line hadoop



Dataflow: Overview

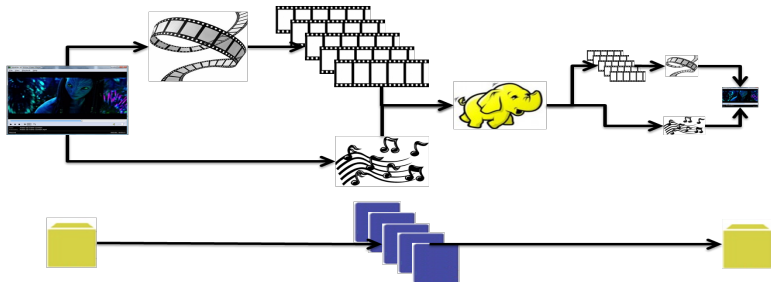


Figure 6 : mohohan's dataflow



Dataflow: Splitting

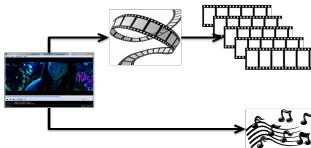


Figure 7 : splitting dataflow

Input: multimedia on AWS S3: $\{\text{mediaons3}\}$

Result: audio/video chunks: $\{\text{multimedia}\}.\text{a.mp4}$, $\{\text{multimedia}\}.\text{v.[0-n].mp4}$

```
s3cmd get  $\{\text{mediaons3}\}$   $\{\text{multimedia}\}$  ;
```

```
ffmpeg -i  $\{\text{multimedia}\}$  -vn -acodec copy  $\{\text{multimedia}\}.\text{a.mp4}$  ;
```

```
ffmpeg -i  $\{\text{multimedia}\}$  -vcodec copy -an  $\{\text{multimedia}\}.\text{v.mp4}$  ;
```

```
mkvmerge -split size:32m  $\{\text{multimedia}\}.\text{v.mp4}$  -o  $\{\text{multimedia}\}.\text{v.}\%1d.\text{mp4}$  ;
```

Algorithm 1: bash splitting script



Dataflow: Transcoding

Input: original audio/video chunks: $\${input} = \${multimedia}.a.mp4, \${multimedia}.v.[0-n].mp4$

Result: transcoded audio/video chunks: $\${multimedia}.a.ok.mp4, \${multimedia}.v.[0-n].ok.mp4$

```
filenameA='basename  $\${input}$ ';  
filenameB='basename  $\${input}$  .a.mp4';  
if [  $\${filenameA}$  !=  $\${filenameB}$  ]; then ;  
  for i in {1..2}; do ;  
    ffmpeg -y -i  $\${filename}$  -vn -acodec libfaac -ab  $\${abr}$ k -pass  $\${i}$  -threads 0  $\${filename}.ok.mp4$  ;  
  done ;  
else ;  
  for i in {1..2}; do ;  
    ffmpeg -y -i  $\${filename}$  -vcodec libx264 -b  $\${vbr}$ k -an -pass  $\${i}$  -threads 0  $\${filename}.ok.mp4$  ;  
  done ;  
fi ;
```

Algorithm 2: bash transcoding script



Dataflow: Merging

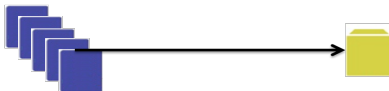


Figure 8 : merging dataflow

Input: transcoded audio/video chunks: `${multimedia}.a.ok.mp4`, `${multimedia}.v.[0-n].ok.mp4`

Result: transcoded multimedia on AWS S3: `${mediaons3}.ok.mp4`

```
mergestr="mkvmerge -o ${multimedia}.v.ok.mp4 " ;
for ((i=1;i<=${splits};i++)) do ;
  _mergestr="+ ${mergestr} ${multimedia}.v.${i}.ok.mp4" ;
done ;
'${mergestr}' ;
ffmpeg -i ${multimedia}.a.ok.mp4 -i ${multimedia}.v.ok.mp4 -vcodec copy -acodec copy ${multimedia}.ok.mp4 ;
s3cmd put ${multimedia}.ok.mp4 ${mediaons3}.ok.mp4 ;
```

Algorithm 3: bash merging script



Benchmark

- Relevant services: zencoder, encoding, sorenson, and panda

Table 1 : The details of test data: Avatar movie trailer

Video Bitrate (kbps.)	near 10000
Audio Bitrate (kbps.)	near 100
Duration (secs.)	209

Table 2 : Transcode settings for benchmark

Video Bitrate (kbps.)	3000
Video Codec	H.264
Audio Bitrate (kbps.)	96
Audio Codec	AAC
Encode Passes	2

- mohohan settings: 10 c1.xlarge EC2 instances



Benchmark cont.

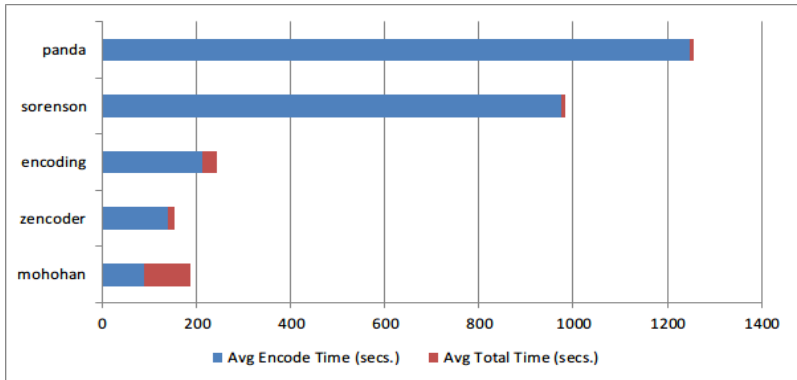


Figure 9 : Benchmark performance among the services



Demo

- <http://mohohan.no-ip.org>



Thanks

- Thank you for your attention :)
- Here are some useful links:
 - AWS: <http://aws.amazon.com>
 - Benchmark: <http://goo.gl/rPXZz>
 - mohohan: <http://mohohan.no-ip.org>
 - zencoder: <http://zencoder.com>
 - encoding: <http://www.encoding.com>
 - sorenson: <http://www.sorensonmedia.com>
 - panda: <http://www.pandastream.com>