

**Video365**

Project: *video365*

**Transcoder Deployment Guide**

Version: *1.2*

ID: *2*

**Nội Dung**

[1 Phạm vi 3](#_Toc463268832)

[2 Mục đích 3](#_Toc463268833)

[3 Thông tin tài liệu 4](#_Toc463268834)

[3.1 Phê duyệt tài liệu 4](#_Toc463268835)

[3.2 Lịch sử tài liệu 4](#_Toc463268836)

[3.3 Tài liệu liên quan 4](#_Toc463268837)

[4 Cài đặt môi trường 5](#_Toc463268838)

[4.1 Hệ điều hành 5](#_Toc463268839)

[4.2 Các thư viện hỗ trợ 5](#_Toc463268840)

[5 Cài đặt ứng dụng 5](#_Toc463268841)

[5.1 Danh sách thành phần 5](#_Toc463268842)

[5.2 FFMPEG 6](#_Toc463268843)

[5.2.1 Xây dựng gói 6](#_Toc463268844)

[5.2.1 Kiểm tra 7](#_Toc463268845)

[6 Triển khai Transcoder script 7](#_Toc463268846)

[6.1 Danh sách thành phần 7](#_Toc463268847)

[6.2 Nội dung script transcoder 7](#_Toc463268848)

[6.3 Triển khai script 18](#_Toc463268849)

[6.4 Khởi động tiến trình transcoder 18](#_Toc463268850)

[6.5 Theo dõi quá trình transcoder 19](#_Toc463268851)

[6.6 Dừng tiến trình transcoder 19](#_Toc463268852)

# Phạm vi

* Tài liệu liên quan:
  + Thiết kế tổng quan:
  + Hướng dẫn sử dụng:
* Mô tả cách build và cài đặt

# Mục đích

* Tài liệu hướng dẫn build ffmpeg và các lib tương ứng
* Hướng dẫn build NVidia SDK để sử dụng GPU
* Tài được giành cho develop hoắc system admin triển khai hoặc tích hợp hệ thống headend vào ứng dụng xác định

# Thông tin tài liệu

## Phê duyệt tài liệu

<This document was reviewed by the following signatories who have read, understood and agreed the intent, context and outcomes of this document.>

|  |  |  |  |
| --- | --- | --- | --- |
| *Tên* | *Chức vụ* | *Email* | *Phương thức phê duyệt*  (Email/Signature) |
|  |  |  |  |

## Lịch sử tài liệu

<When using this template remove the template history details and replace with the history details of the new document>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *(\*) Ngày* | *Phiên bản* | *Mục thay đổi* | *Chi tiết thay đổi* | *(\*\*) A, D, M* | *Người viết* |
| 03/06/2016 | 1.0 |  | Khởi tạo | A | Dương Quốc Huy |
| 28/09/2016 | 1.0 |  | Qui trình cài đặt | A | Phan Viet Anh |
| 29/09/2016 | 1.1 |  | Thêm nội dung script | A | Phan Viet Anh |
| 03/10/2016 | 1.2 |  | Sửa lại cách cài đặt | M | Phan Viet Anh |

*(\*): Date format: dd/mm/yyyy*

*(\*\*): A – Added; M – Modified; D – Deleted*

## Tài liệu liên quan

|  |  |  |
| --- | --- | --- |
| *Tiêu đề* | *ID* | *Phiên bản* |
| Video365 - Model Deployment Guide | 1 | 1.2 |

# Transcode Architecture



- Main.sh: là script để khởi động transcode\_api.sh, có tùy chọn gọi 1 hoặc nhiều tiến trình transcoder.

- Transcode\_api.sh: làm nhiệm vụ gọi đến api cms để lấy thông tin raw và điều khiển ffmpeg xử lý video, sau khi transcode xong script sẽ trả về các thông tin của file thành phẩm cho cho CMS.

- FFmpeg: là phần mềm chuyên xử lý video, dùng để tạo ra các video thành phẩm theo các định dạng khác nhau.

- Store Raw: nơi chưa các video cần được transcode

- Store File: nơi chứa các video thành phẩm sau khi được transcode.

- Store log: bao gồm error\_trans\_process.log, update\_fail.log, update\_success.log. Nơi chứa các log các thông tin success, fail của tiến trình transcode. Dùng để debug khi có lỗi xảy ra.

# Mô tả API

# Transcode Flow



# Cài đặt môi trường

Để đảm bảo hệ thống headed build thành công và hoạt động ổn định thì cần đáp ứng các yêu cầu sau

## Hệ điều hành

Để có thể chạy nén H264 và sử dụng GPU để transcode thì cẩn cài đặt OS Centos 6

## Các thư viện hỗ trợ

Cài đặt các thư viện hệ thống và tạo biến môi trường

# export PKG\_CONFIG\_PATH=/usr/local/lib/pkgconfig

# yum install –y git gcc.x86\_64 gcc-c++.x86\_64 \

zlib-devel.x86\_64 freetype-devel.x86\_64 expat-devel.x86\_64

# Cài đặt ứng dụng

Transcoder thành phần chính là FFMPG, nhưng để ffmpeg có thể decode và encode ra các định dạng khác nhau thì cần phải cài đặt thêm các lib hỗ trợ. Các thành phần cài đăt sẽ được mô tả ở dưới đây.

## Danh sách thành phần

|  |  |  |  |
| --- | --- | --- | --- |
| *Component* | *Version* | *Reference Documnet* | *Installation Package Source* |
| Yasm |  |  | #git clone git://github.com/yasm/yasm.git |
| H.265/HEVC video encoder |  |  | #wget <https://bitbucket.org/multicoreware/x265> |
| AAC audio encoder. |  |  | #git clone git://git.code.sf.net/p/opencore-amr/fdk-aac |
| MP3 audio encoder. |  |  | #wget <http://downloads.sourceforge.net/project/lame/lame/3.99/lame-3.99.5.tar.gz> |
| VP8/VP9 video encoder |  |  | #wget <https://chromium.googlesource.com/webm/libvpx.git> |
| FFMPEG |  |  | #git clone git://source.ffmpeg.org/ffmpeg |

1. Cài đặt yasm (assembler)

# cd /root && tar -xzvf yasm-1.2.0.tar.gz

# cd yasm-1.2.0 && ./configure && make && make install

1. Cài đặt thư viện Video

# cd /root && tar -xjvf x264-snapshot-20140522-2245-stable.tar.bz2

# cd x264-snapshot-20140522-2245-stable

# ./configure --enable-shared && make && make install

# cd /root && tar -xzvf xvidcore-1.3.3.tar.gz

# cd xvidcore/build/generic && ./configure && make && make install

1. Cài đặt thư viện Audio

# cd /root && tar -xzvf faac-1.28.tar.gz

# cd faac-1.28 && ./configure && make && make install

# cd /root && tar -xzvf lame-3.99.5.tar.gz

# cd lame-3.99.5 && ./configure && make && make install

# cd /root && tar -xzvf opencore-amr-0.1.3.tar.gz

# cd opencore-amr-0.1.3 && ./configure && make && make install

1. Cài đặt thư viện Subtitle

# cd /root && tar -xjvf fribidi-0.19.6.tar.bz2

# cd fribidi-0.19.6

##edit source code to fix error of PAGE\_SIZE

# [ "${CROSS\_PREFIX}" == "arm-marvell-linux-gnueabi" ] || \

sed -i "s|include <asm/page.h>|include <unistd.h>\n# \

define PAGE\_SIZE sysconf(\_SC\_PAGESIZE)|" lib/common.h

# ./configure && make && make install

# cd /root && tar -xzvf fontconfig-2.11.1.tar.gz

# cd fontconfig-2.11.1 && ./configure && make && make install

# cd /root && tar -xzvf libass-0.10.2.tar.gz

# cd libass-0.10.2 && ./configure && make && make install

## FFMPEG

### Xây dựng gói

# cd /root && tar -xzvf ffmpeg-2.2.2.tar.gz

# cd ffmpeg-2.2.2

# ./configure --prefix=/opt/ffmpeg-2.2.2 \

--enable-gpl --enable-version3 --enable-nonfree \

--disable-ffplay --disable-ffserver \

--enable-postproc --enable-avfilter --enable-pthreads \

--enable-libx264 --enable-libxvid \

--enable-libfaac --enable-libmp3lame \

--enable-libopencore-amrnb --enable-libopencore-amrwb \

--enable-libass --enable-pic

# make && make install

Tạo shortcut (thuận tiện cho việc nâng cấp version)

# ln -s /opt/ffmpeg-2.2.2 /opt/ffmpeg

Thiết lập shortcut tới các file chạy

# echo 'export PATH=$PATH:/opt/ffmpeg/bin' >> ~/.bashrc

# source ~/.bashrc

Cài đặt font

**# rpm -ivh http://dl.atrpms.net/el6.6-x86\_64/atrpms/stable/atrpms-repo-6-6.el6.x86\_64.rpm**

### Kiểm tra

Khi file ffmpeg có, chạy file để show version

# ffmpeg -version

ffmpeg version 2.2.2

built on Jun 3 2014 11:25:36 with gcc 4.1.2 (GCC) 20080704 (Red Hat 4.1.2-52)

configuration: --prefix=/opt/ffmpeg-2.2.2 --enable-gpl --enable-version3 --enable-nonfree --disable-ffplay --disable-ffserver --enable-postproc --enable-avfilter --enable-pthreads --enable-libx264 --enable-libxvid --enable-libfaac --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libass --enable-pic

libavutil 52. 66.100 / 52. 66.100

libavcodec 55. 52.102 / 55. 52.102

libavformat 55. 33.100 / 55. 33.100

libavdevice 55. 10.100 / 55. 10.100

libavfilter 4. 2.100 / 4. 2.100

libswscale 2. 5.102 / 2. 5.102

libswresample 0. 18.100 / 0. 18.100

libpostproc 52. 3.100 / 52. 3.100

# Triển khai Transcoder script

## Danh sách thành phần

|  |  |  |  |
| --- | --- | --- | --- |
| *Component* | *Version* | *Reference Documnet* | *Chức năng* |
| main.sh |  |  | Tạo các tiến trình để xử lý transcode |
| transcode\_api.sh |  |  | Tiến hành transcode |

## Nội dung script transcoder

#### main.sh

dùng để khởi động hoặc thêm tiến trình transcoder

# vi /opt/script\_trans/main.sh

#!/bin/bash

#######################################################

# MAIN SCRIPT TRANSCODE API - mAloTV

#######################################################

###Description

# This script to control transcode process. It has some function:

# - Initialize a/some transcode process(es) if it has not any transcode process before

# - Add more a/some transcode process(es) if it had some running transcode process

###

echo -e "Ban co muon khoi tao hay them tien trinh transcode?\n"

echo -e "1.Khoi tao\n"

echo -e "2.Them\n"

read answer

base\_dir=`pwd`

if [ $answer -eq 2 ];then

echo -ne "Nhap vao so tien trinh muon them: "

read m

num\_log=`ps -aux 2>/dev/null | grep '/bin/bash' | grep "$base\_dir/transcode\_api.sh" | wc -l`

for (( i = 1; i <= $m; i++ ))

do

nohup $base\_dir/transcode\_api.sh $[$num\_log + $i] &

done

elif [ $answer -eq 1 ]; then

echo -ne "Nhap vao so tien trinh muon chay: "

read n

list\_process=`ps aux | grep '/bin/bash' | grep "$base\_dir/transcode\_api.sh" | awk -F ' ' '{print$2}'`

for k in $list\_process; do

kill -9 $k

done

for (( i = 1; i <= $n; i++ ))

do

nohup $base\_dir/transcode\_api.sh $i &

done

else

echo -e "Ban phai an so tien trinh ???\n"

fi

#### transcode\_api.sh

Kết nối đến cms và xử lý nghiệp vu transcoder.

# vi /opt/script\_trans/transcode\_api.sh

#!/bin/bash

#################################################

# SCRIPT TRANSCODE API - MSP

#################################################

###Description

#Input: $transcode\_id: is transcode process number. The main process count number of transcode process to generate a/some new transcode process(es)

#Ouptput: log\_files: include

# - duplicate.log : store infor about error file which be duplicated with a previous raw

# - error\_trans\_process.log : store general error information of all transcode fuctions. It is useful to troubleshoot

# - check\_duplicate.log : store hash code of raw files. It is used to check duplicate file

###

###Configuration parameters

cms='<Domain api cms>'

cms\_port='<port cms>'

disk\_id='disk1' ###if change this, must also change $file\_path of update\_cms function##

file\_dir="/opt/transcode/$disk\_id/file"

log\_dir='/opt/logs'

error\_dir="$log\_dir/error"

#fix ratio

fix\_ratio=1.55

#temp folder store subtitle srt & ass

tmp\_dir='/opt/temp'

temp\_srt="$tmp\_dir/temp$1.srt"

temp\_ass="$tmp\_dir/temp$1.ass"

###########################

#check & make folder

mkdir -p $error\_dir

mkdir -p $file\_dir

mkdir -p $tmp\_dir

###FUNCTION

###########################

#function check\_mount

#Desc: check mount connection

#input: $raw\_path

#0 is ok, 1 is nok

function check\_mount {

# mount\_point=`echo $raw\_path | awk -F'/' '{print$1,$2,$3}' | tr ' ' '/'`

if mount | grep "$file\_dir" > /dev/null; then

return 0

else

echo -e "$(date) Error: Can't mount to Origin.....Check immediately,plz " >> $error\_dir/error\_trans\_process.log

return 1

fi

}

#function check cms connection

#input $cms $cms\_port

function check\_cms {

nc -z -w2 $1 $2

if [ $? -ne 0 ]; then

echo -e "$(date) FATAL Error: Oooop.....connection to cms $cms error\n" >> $error\_dir/error\_trans\_process.log

sleep $sleep\_time && continue

fi

}

#function fault\_func

#Desc: This function will be called when process be fail

#Input: $1

# 1: update cms to inform that raw is error (can not transcode) and set this file to inactive status

# 2: update cms to inform that transcode process is error while raw file still transcode normally and resume status to active

function fault\_func {

if [ $1 -eq 1 ]; then

status\_code=7

else

status\_code=6

fi

mv "$temp\_path" "$raw\_path" 2>/dev/null

# Remove hash in check\_duplicate file

local hash=`cat $log\_dir/hash.tmp | grep $profile\_id | awk -F' ' '{print$1}'`

sed -i "/$hash/ s/^.\*$//;/^\s\*$/d" $log\_dir/check\_duplicate.log

curl --data "profile\_id=$profile\_id&status=$status\_code" "http://$cms:$cms\_port/index.php/converter/update-status-content"

#curl --data "profile\_id=$profile\_id&status=$status\_code" "http://$cms/index.php/converter/unlock-content"

sleep $sleep\_time && continue

}

#function standardize\_name

#Input: $raw\_path (video's name, subtitle's name)

#Output: $temp\_path

function standardize\_name {

raw\_dir=`dirname "$1"`

old\_name=`basename "$1"`

stand\_name=`echo ${old\_name%.\*} | tr '[:upper:]' '[:lower:]' | tr -c [:alnum:][:cntrl:] - |tr -s - - | sed 's/^-//' | sed 's/-$//'`

ext=`echo ${old\_name##\*.}`

temp\_path="$raw\_dir/$stand\_name.$ext"

mv "$1" "$temp\_path" 2>/dev/null

}

#function check\_raw

# - check\_raw

# - check\_format

# - check\_duplicate

#Check existence and duration video raw

#input: $raw\_path

#output: 0 is ok, 1 is nok

function check\_raw {

if [ -f $1 ]; then

local duration\_raw=`ffprobe -loglevel quiet -print\_format default=nokey=1:noprint\_wrappers=1 -show\_entries format=duration $1 | cut -d '.' -f1`

if [ $duration\_raw -ne 0 ]; then

echo 0

else

echo -e "$(date) Error: video raw $raw\_path is not a video" >> $error\_dir/error\_trans\_process.log

echo 1

fi

else

echo -e "$(date) Error: video raw $raw\_path is not existed" >> $error\_dir/error\_trans\_process.log

echo 1

fi

}

function check\_format {

local list="mp4 wav avi mkv flv mp3 3gp wmv m4v ts mov vob mpg divx webm tp mpeg"

local file=$(basename $1)

local ext=`echo ${file##\*.}`

if [[ $list =~ $ext ]]; then

echo 0

else

echo "$(date) Error: This extension is not support for $raw\_path" >> $error\_dir/error\_trans\_process.log

echo 1

fi

}

function check\_duplicate {

#Module hash file MD5

local filesize=`stat -c %s "$1"`

local skipamount="$((${filesize} / 2048))"

local MD5FILE=`dd if="$1" bs=1024 count=2 skip=$skipamount 2>/dev/null |md5sum |cut -d ' ' -f1`

#Check in check\_duplicate.log; if existed, skip and write to duplicate.log. Otherise, add hash code to check\_duplicate.log file

grep -m 1 $MD5FILE $log\_dir/check\_duplicate.log >/dev/null

if [ $? -eq 0 ]; then

echo -e "$(date) $profile\_id - $raw\_path - $MD5FILE" >> $error\_dir/duplicate.log

echo -e "$(date) Error: $raw\_path is duplicate video, check again,plz" >> $error\_dir/error\_trans\_process.log

echo -e " If you still want to transcode this video, you should remove $MD5FILE in $log\_dir/check\_duplicate.log" >> $error\_dir/error\_trans\_process.log

echo 1

else

echo "$MD5FILE" >> $log\_dir/check\_duplicate.log

echo "$MD5FILE $profile\_id" >> $log\_dir/hash.tmp

echo 0

fi

}

#function check\_sub

#input: $raw\_path $temp\_srt $temp\_ass

#output: 0 is ok, 1 is no subtitle

function check\_sub {

if [ -z $sub\_path ]; then

sub\_path=`echo $1 | sed 's/\..\*$/\.srt/'`

fi

if [ -f $sub\_path ]; then

file -bi $sub\_path |grep -i utf-16 >/dev/null 2>&1

if [ $? == 0 ]; then

iconv -f UTF-16 -t UTF-8 $sub\_path > $2

else

cp -f $sub\_path $2

fi

ffmpeg -f srt -i $2 -f ass -y $3 >/dev/null 2>&1

# rm -rf $sub\_path

return 0;

else

echo '' > $3

echo -e "$(date) Warning: No subtitle for $1" >> $error\_dir/error\_trans\_process.log

return 1;

fi

}

###########################

# FUNCTIONS BUILD VIDEO FILTER, VIDEO OPTION AND AUDIO OPTION

###########################

#Build video filter

#Input: $widthmax $heightmax $temp\_ass

function build\_vf {

if [ $compare\_ratio -eq 1 ]; then

vf="scale=$1:trunc'(ow/a/2)'\*2"

else

vf="scale=trunc'(oh/a/2)'\*2:$2"

fi

if [ "$3" != "" ]; then

vf="ass=$3,$vf"

fi

echo $vf

}

#Build video option (codec, bitrate, filter, preset)

#Input: $codec $bitrate $vf $preset

function build\_video {

if [ "$4" == "" ]; then

echo -c:v $1 -b:v $2 -vf $3

else

echo -c:v $1 -b:v $2 -vf $3 -vpre $4

fi

}

#Build\_audio (codec, bitrate, channel, rate, volume)

#Input: $codec $bitrate $channel $rate $volume

function build\_audio {

echo -c:a $1 -b:a $2 -ac $3 -ar $4 -af volume=$5

}

#Build output

#Input: $temp\_path $video\_version

function output {

raw\_name=$(basename $1)

raw\_name\_no\_ext=$(echo ${raw\_name%.\*})

#output

mkdir -p "$destination\_folder/${raw\_name\_no\_ext}"

echo "$destination\_folder/${raw\_name\_no\_ext}/${raw\_name\_no\_ext}\_$2.mp4"

}

#Build file m3u8

#Input: $output\_version ($output\_SU, $output\_SD,...)

function build\_m3u8 {

mp4split -o `echo $1 | sed 's/ismv/ism/'` $1

mp4split -o `echo $1 | sed 's/ismv/m3u8/'` `echo $1 | sed 's/ismv/ism/'`

}

#Function check ouput file

#file after transcode completely should compare duration with raw file to know whether transcode successful or not

#Input: $transcoded\_file

#Output: 0 is ok, 1 not ok

function check\_output {

duration\_file=`ffprobe -loglevel quiet -print\_format default=nokey=1:noprint\_wrappers=1 -show\_entries format=duration $1 | cut -d '.' -f1`

if [ `expr $duration\_file \\* 100 / $duration\_raw` -lt 95 ]; then

echo "$(date) Error: File after transcode has too short duration than original raw file. Check raw $raw\_path, plz" >> $error\_dir/error\_trans\_process.log

fault\_func 1

fi

}

#Function update\_cms

#Update outputs info to cms

#Input: $profile\_id $content\_id $output\_SU

#Output: update\_success.log update\_fail.log

function update\_cms {

echo "##Start Update CMS##" >> $error\_dir/error\_trans\_process.log

local profile\_id=$1

local content\_id=$2

local file\_path=`echo $3 | awk -F '/opt/transcode/disk1/file/' '{print$2}'`

local bitrate=`ffprobe -loglevel quiet -print\_format default=nokey=1:noprint\_wrappers=1 -show\_entries format=bit\_rate $3`

local width=`ffprobe -loglevel quiet -print\_format default=nokey=1:noprint\_wrappers=1 -show\_entries stream=width $3`

local height=`ffprobe -loglevel quiet -print\_format default=nokey=1:noprint\_wrappers=1 -show\_entries stream=height $3`

local duration=`ffprobe -loglevel quiet -print\_format default=nokey=1:noprint\_wrappers=1 -show\_entries format=duration $3 | cut -d '.' -f1`

local output=`curl --data "profile\_id=$profile\_id&content\_id=$content\_id&content\_type=$content\_type&url=$file\_path&bitrate=$bitrate&width=$width&height=$height&duration=$duration" "http://$cms:$cms\_port/index.php/converter/content-converted" | sed 's/\({\|"\|}\)//g'`

local result=`echo $output | awk -F',' '{print$1}' | awk -F':' '{print$2}'`

local content\_type=$4

if [ "$result" != 'true' ]; then

echo "curl --data \"profile\_id=$profile\_id&content\_id=$content\_id&content\_type=$content\_type&url=$file\_path&bitrate=$bitrate&width=$width&height=$height\" \"http://$cms:$cms\_port/index.php/converter/content-converted\"" >> $error\_dir/update\_fail.log

echo "$(date) Error: Video $file\_path update fail: $output" >> $error\_dir/error\_trans\_process.log

return 1

else

echo "curl --data \"profile\_id=$profile\_id&content\_id=$content\_id&content\_type=$content\_type&url=$file\_path&bitrate=$bitrate&width=$width&height=$height\" \"http://$cms:$cms\_port/index.php/converter/content-converted\"" >> $error\_dir/update\_success.log

return 0

fi

echo "##Done\n##" >> $error\_dir/error\_trans\_process.log

}

#function update status content

#Input: $content\_id

#status\_code:

# const STATUS\_ACTIVE = 1;

# const STATUS\_INACTIVE = 0;

# const STATUS\_TEST = 3;

# const STATUS\_TRANCODED = 4; // DA TRANSCOE

# const STATUS\_TRANCODE\_PENDING = 5; // DANG TRANSCOE, Khoa

# const STATUS\_RAW=6; // raw chua transcode

# const STATUS\_RAW\_ERROR=7; // raw error

# const STATUS\_UPLOADING=8; // raw error

function update\_status {

local profile\_id=$1

curl --data "profile\_id=$profile\_id&status=4" "http://$cms:$cms\_port/index.php/converter/update-status-content" >> $error\_dir/error\_trans\_process.log

}

#Function: perform transcoding

function transcode {

for i in "$@"; do

case "$i" in

SU)

video\_SU=$(build\_video libx264 2000k $(build\_vf 1280 720 $sub))

audio\_SU=$(build\_audio libfaac 128k 2 44100 2)

output\_SU=$(output $temp\_path $i)

;;

HD)

video\_HD=$(build\_video libx264 1200k $(build\_vf 1280 720 $sub))

audio\_HD=$(build\_audio libfaac 128k 2 44100 2)

output\_HD=$(output $temp\_path $i)

;;

SD)

video\_SD=$(build\_video libx264 800k $(build\_vf 848 480 $sub))

audio\_SD=$(build\_audio libfaac 64k 2 44100 2)

output\_SD=$(output $temp\_path $i)

;;

MB)

video\_MB=$(build\_video libx264 500k $(build\_vf 640 360 $sub))

audio\_MB=$(build\_audio libfaac 64k 2 44100 2)

output\_MB=$(output $temp\_path $i)

;;

FP)

video\_FP=$(build\_video libx264 300k $(build\_vf 424 240 $sub))

audio\_FP=$(build\_audio libfaac 64k 2 44100 2)

output\_FP=$(output $temp\_path $i)

;;

esac

done

case "$#" in

1)

echo "ffmpeg $main\_opt $video\_FP $audio\_FP $output\_FP" >> $error\_dir/error\_trans\_process.log

ffmpeg $main\_opt $video\_FP $audio\_FP $output\_FP

;;

2)

echo "ffmpeg $main\_opt $video\_MB $audio\_MB $output\_MB $video\_FP $audio\_FP $output\_FP" >> $error\_dir/error\_trans\_process.log

ffmpeg $main\_opt $video\_MB $audio\_MB $output\_MB $video\_FP $audio\_FP $output\_FP

;;

3)

echo "ffmpeg $main\_opt $video\_SD $audio\_SD $output\_SD $video\_MB $audio\_MB $output\_MB $video\_FP $audio\_FP $output\_FP" >> $error\_dir/error\_trans\_process.log

ffmpeg $main\_opt $video\_SD $audio\_SD $output\_SD $video\_MB $audio\_MB $output\_MB $video\_FP $audio\_FP $output\_FP

;;

4)

echo "ffmpeg $main\_opt $video\_HD $audio\_HD $output\_HD $video\_SD $audio\_SD $output\_SD $video\_MB $audio\_MB $output\_MB $video\_FP $audio\_FP $output\_FP" >> $error\_dir/error\_trans\_process.log

ffmpeg $main\_opt $video\_HD $audio\_HD $output\_HD $video\_SD $audio\_SD $output\_SD $video\_MB $audio\_MB $output\_MB $video\_FP $audio\_FP $output\_FP

;;

5)

echo "ffmpeg $main\_opt $video\_SU $audio\_SU $output\_SU $video\_HD $audio\_HD $output\_HD $video\_SD $audio\_SD $output\_SD $video\_MB $audio\_MB $output\_MB $video\_FP $audio\_FP $output\_FP" >> $error\_dir/error\_trans\_process.log

ffmpeg $main\_opt $video\_SU $audio\_SU $output\_SU $video\_HD $audio\_HD $output\_HD $video\_SD $audio\_SD $output\_SD $video\_MB $audio\_MB $output\_MB $video\_FP $audio\_FP $output\_FP

;;

esac

if [ $? ne 0 ]; then

#if transcode fail, log && recover raw\_path

echo "$(date) Error: Transcode process fail with $raw\_path" >> $error\_dir/error\_trans\_process.log

fault\_func 2

fi

check\_output $output\_FP

#Remove raw file

# rm -rf $temp\_path

# rm -rf $sub\_path

sed -i "/$profile\_id/ s/^.\*$//;/^\s\*$/d" $log\_dir/hash.tmp

}

######################################################

#MAIN BODY - TRANSCODE PROCESS

######################################################

while [ 1 ]; do

sleep\_time=`awk -vmin=60 -vmax=100 'BEGIN{srand(); print int(min+rand()\*(max-min+1))}'`

#get & process raw infor

check\_cms $cms $cms\_port

get\_link="http://$cms:$cms\_port/converter/contents-raw"

output=`curl -m 5 "$get\_link" | sed 's/\({\|"\|}\)//g'`

echo "$(date) Info: Get link result: $output" >> $error\_dir/error\_trans\_process.log

result=`echo $output | awk -F',' '{print$1}' | awk -F':' '{print$2}'`

# raw\_path='/opt/transcode/raw/test.mkv'

if [ "$result" != 'true' ]; then

echo "$(date) Warning: no raw file .....go to bed...zzz...zzz" >> $error\_dir/error\_trans\_process.log

sleep $sleep\_time && continue

else

profile\_id=`echo $output | awk -F',' '{print$3}' | awk -F':' '{print$3}'`

content\_id=`echo $output | awk -F',' '{print$4}' | awk -F':' '{print$2}'`

cp\_id=`echo $output | awk -F',' '{print$5}' | awk -F':' '{print$2}'`

raw\_path=`echo $output | awk -F',' '{print$6}' | awk -F':' '{print$2}'`

sub\_path=`echo $output | awk -F',' '{print$7}' | awk -F':' '{print$2}'`

content\_type=`echo $output | awk -F',' '{print$8}' | awk -F':' '{print$2}'`

# check\_mount $raw\_path

# if [ $? -eq 1 ]; then

# sleep $sleep\_time && continue

# fi

SAVEIFS=$IFS

IFS=$'\n'

standardize\_name "$raw\_path"

check\_sub $raw\_path $temp\_srt $temp\_ass

if [ $? -eq 1 ]; then

sub=''

else

sub=$temp\_ass

fi

IFS=$SAVEIFS

# if [[ $(check\_raw $temp\_path) -eq 0 && $(check\_format $temp\_path) -eq 0 && $(check\_duplicate $temp\_path) -eq 0 ]]; then

if [[ $(check\_raw $temp\_path) -eq 0 && $(check\_format $temp\_path) -eq 0 ]]; then

#Check vs cut folder destination

if [ `ls $file\_dir/$cp\_id | grep video-file- | wc -l` -eq 0 ]; then

mkdir -p $file\_dir/$cp\_id/video-file-0

fi

exist\_num=`ls -dt $file\_dir/$cp\_id/video-file-\* | head -n 1 | cut -d '-' -f3`

destination\_folder="$file\_dir/$cp\_id/video-file-$exist\_num"

file\_total=`ls $destination\_folder | wc -l`

if [ "$file\_total" -ge 1000 ]; then

next\_num=$[exist\_num + 1]

destination\_folder=$file\_dir/$cp\_id/video-file-$next\_num

mkdir -p $destination\_folder

fi

#/opt/transcode/file/$disk\_id/video-file-$number/$stand\_name\_HD.ssm/$stand\_name\_HD.ismv

main\_opt="-loglevel warning -i $temp\_path -y -map\_chapters -1"

#Check raw quality: revolution of video is base on height of video.

#So we'll check height of raw video to decide the number of output versions

#For example:

#240p (424x240, 0.10 megapixels)

#360p (640x360, 0.23 megapixels)

#480p (848x480, 0.41 megapixels, "SD" or "NTSC widescreen")

#720p (1280x720, 0.92 megapixels, "HD")

#1080p (1920x1080, 2.07 megapixels, "Full HD")

duration\_raw=`ffprobe -loglevel quiet -print\_format default=nokey=1:noprint\_wrappers=1 -show\_entries format=duration $temp\_path | cut -d '.' -f1`

raw\_width=$(ffprobe -loglevel quiet -print\_format default=nokey=1:noprint\_wrappers=1 -show\_entries stream=width $temp\_path | head -n 1)

raw\_height=$(ffprobe -loglevel quiet -print\_format default=nokey=1:noprint\_wrappers=1 -show\_entries stream=height $temp\_path | head -n 1)

raw\_ratio=$(echo "scale=2;$raw\_width/$raw\_height"|bc)

compare\_ratio=$(echo "$raw\_ratio > $fix\_ratio"|bc)

if (( 0 <= $raw\_height && $raw\_height <= 240 )); then

transcode FP

update\_cms $profile\_id $content\_id $output\_FP $content\_type

update\_status $profile\_id

elif (( 241 <= $raw\_height && $raw\_height <= 360 )); then

transcode FP MB

update\_cms $profile\_id $content\_id $output\_MB $content\_type

update\_cms $profile\_id $content\_id $output\_FP $content\_type

update\_status $profile\_id

elif (( 361 <= $raw\_height && $raw\_height <= 480)); then

transcode FP MB SD

update\_cms $profile\_id $content\_id $output\_SD $content\_type

update\_cms $profile\_id $content\_id $output\_MB $content\_type

update\_cms $profile\_id $content\_id $output\_FP $content\_type

update\_status $profile\_id

elif (( 481 <= $raw\_height && $raw\_height <= 720 )); then

transcode FP MB SD HD

update\_cms $profile\_id $content\_id $output\_HD $content\_type

update\_cms $profile\_id $content\_id $output\_SD $content\_type

update\_cms $profile\_id $content\_id $output\_MB $content\_type

update\_cms $profile\_id $content\_id $output\_FP $content\_type

update\_status $profile\_id

else

transcode FP MB SD HD SU

update\_cms $profile\_id $content\_id $output\_SU $content\_type

update\_cms $profile\_id $content\_id $output\_HD $content\_type

update\_cms $profile\_id $content\_id $output\_SD $content\_type

update\_cms $profile\_id $content\_id $output\_MB $content\_type

update\_cms $profile\_id $content\_id $output\_FP $content\_type

update\_status $profile\_id

fi

else

fault\_func 1

fi

fi

done

## Triển khai script

(2 script được đính kèm theo tài liệu này)

Thực hiện việc copy thư mục script vào /opt/script\_trans/

Khai báo API của CMS để làm việc cùng transcoder:

###Configuration parameters

cms='<Domain api của cms>'

cms\_port='<port cms>'

Thực hiện khai báo thêm domain và IP:

# vi /etc/host

<IPcuacms> <Domain api của cms>

Tạo thư mục chứ transcode

# mkdir -p /opt/transcode/disk1

Mount ổ đĩa có dung lượng lớn để chưa file video.

# mount /dev/mapper/vg\_streamingorig-LogVol02 /opt/transcode/disk1

Cấu hình cho volume tư động mount khi hệ thông reboot

# vi /etc/fstat

/dev/mapper/vg\_streamingorig-LogVol02 /opt/transcode/disk1 ext4 defaults 1 2

Tạo thư mục file raw va file thành phẩm

# mkdir -p /opt/transcode/disk1/{file,raw}

Khai báo đường dẫn chứa file thành phẩm trong script

# vi /opt/script\_trans/transcode\_api.sh

disk\_id='disk1'

file\_dir="/opt/transcode/$disk\_id/file"

## Khởi động tiến trình transcoder

#cd /opt/script\_trans

# ./main\_transcode\_api.sh

Ban co muon khoi tao hay them tien trinh transcode?

1.Khoi tao

2.Them

Kiểm tra tiến trình

# ps ux|grep transcode

root 4615 0.0 0.0 106760 2040 ? S Jun10 6:30 /bin/bash /opt/script\_trans/transcode\_api.sh 1

## Theo dõi quá trình transcoder

Để xem transcoder đang transcode nôi dung nào, có thể tail log như sau:

# tailf /opt/logs/error/error\_trans\_process.log

Thu Jun 9 17:57:36 ICT 2016 Warning: No subtitle for /opt/transcode/disk1/raw/1/bigbuckbunny\_1100.mp4

ffmpeg -loglevel warning -i /opt/transcode/disk1/raw/1/bigbuckbunny-1100.mp4 -y -map\_chapters -1 -c:v libx264 -b:v 896k -vf scale=trunc'(oh/a/2)'\*2:360 -c:a libfaac -b:a 64k -ac 2 -ar 44100 -af volume=2 /opt/transcode/disk1/file/1/video-file-0/bigbuckbunny-1100/bigbuckbunny-1100\_MB.mp4 -c:v libx264 -b:v 576k -vf scale=trunc'(oh/a/2)'\*2:240 -c:a libfaac -b:a 64k -ac 2 -ar 44100 -af volume=2 /opt/transcode/disk1/file/1/video-file-0/bigbuckbunny-1100/bigbuckbunny-1100\_FP.mp4

## Dừng tiến trình transcoder

Tìm tiến trình transcoder và kill theo pid

# ps aux

…

root 29272 0.0 0.0 106248 1484 ? S 12:07 0:00 /bin/bash /opt/script\_trans/transcode\_api.sh 1

…

# kill -9 29272