PPLIVE Developer Design Document开发部程序员设计文档

Transcli转码核心设计文档

**PPLive研发部**

|  |  |  |
| --- | --- | --- |
| *Developers* | | 朱明亮 (zoominla) |
|  | |  |
| *Status* | | Ready for Review (August 4, 2011)  Update (March 16, 2015) |
| Spec产品需求文档 | http://sharepoint/tech/Shared%20Documents/Design%20Docs/client/压制系统内核需求规格说明书.doc | |
| PM产品经理 |  | |
| QA测试人员 | 张玉霞/曾斌 | |
| OPS运维人员 | 胡昊 | |

**Transcli 转码核心 设计说明**

1. **基本流程**

首先解码器将多媒体文件解码成两路原始数据流：音频流(格式：PCM)和视频流(格式：YV12)，然后对这两路原始流进行处理，分别传给音频/视频编码器编码成音视频ES(Elemental Stream)文件，ES是不带封装的中间文件。最后用混流器把两个ES文件封装起来，生成最终的目标文件，比如Mp4文件。

不同阶段的数据流通过管道串联起来。

视频编码

解码器

音频编码

视频处理

音频处理

混流器

MP4

PCM

YUV

PCM

YUV

ES

ES

1. **编解码器及处理工具**
   1. 解码器

解码由ffmpeg进程负责，delogo/scale/fps/crop/deinterlace/subtitle等filter都用ffmpeg处理，根据配置生成音视频filter 字符串。

加Logo及截图由watermark 库负责。

* 1. 编码器

Libx264用于编码H.264视频

Libxvid 用于编码MPEG4视频

FFMPeg用于编码H.263/WMV8/MPEG2等视频

Libfaac 用于编码 AAC-LC音频

Libfdk-aac 用于编码 HE-AAC音频

Liblame 用于编码 Mp3音频

FFMPeg 用于编码 Mp2/WMA8/AC-3等音频

* 1. 混流器

Mp4Box用于封装Mp4/3GP/3G2/F4V等格式

FFMPeg 用于封装 AVI/FLV/WMV/MPG/TS/MKV等格式

* 1. 媒体信息分析

用ffprobe来解析片源的音视频信息，主要解析的视频信息如下：宽/高/宽高比/视频格式/码率/FPS/FPS类型/Interlace

主要解析的音频信息如下：格式/采样率/声道数/码率

* 1. 版本

FFMpeg: 2.3

Libx264: x264 core 129

Libx265: 9340

Mp4Box: 0.4.6-DEV (internal rev. 5)

1. **主要的类及其关系**
   1. 解码器类

目前解码器用的都是进程调用的方式，这些解码器实现的关键就是这些解码器的命令行使用。解码器的主要功能有画面缩放/裁剪/deinterlace/降噪/改变帧率/Resample/声道选择或混合等。

基类：CDecoder

派生类： CDecoderFFMpeg，CDecoderCopy。

* 1. 编码器类

转码系统支持常见的H.264 和 Xvid(MPEG4 Visual)编码，音频支持AAC-LC/HE-AAC/MP3等编码。

视频编码器基类：CVideoEncoder

视频编码器派生类：CX264Encode, CX265Encode, CXvidEncode

音频编码器基类：CAudioEncoder

音频编码器派生类：CFaacEncoder, CFdkAacEncoder, CMp3Encoder

* 1. 混流器类

混流器的作用是把音视频ES流封装起来，形成最终的媒体文件，转码系统有MP4Box，可封装mp4/3GP/3G2/F4V等格式；有FFMpeg，可封装Flv/AVI/WMV/MOV/MPG/TS/MKV等格式；CFlvMuxer，用于封装FLV格式。

基类：CMuxer

派生类：CMP4box, CFFMpegMuxer

* 1. 编解码器及混流器之间的关系

上述的基础类聚合于CTransworker类中，该类封装了整个编解码的过程，(1) CTransworker首先解析配置文件，提取相应的编解码器及混流器的配置信息，然后用ffprobe分析源视频信息，结合配置信息生成相应的解码器类/编码器类/混流器类的实例，根据配置信息可能生成多个编解码器，比如多音轨支持时，会有多个音频编码器及多个音频解码器，对应到不同的音轨。

(2) CTransworker启动解码器，解码器开始分别向音视频管道写数据，然后创建音视频编码线程，在这两个线程内，分别从音视频管道读取数据送到相应的编码器进行编码，编码结果输出到音视频ES文件。在编码的过程中会生成相应的统计信息：转码速度/进度/时间等。

(3) 当编码器解码完毕，编码线程可以从管道获取结束状态，清理编码器并推出线程。主线程等待编码线程结束之后，就启动混流器，生成最终的媒体文件，整个过程结束。

1. **开源代码库的使用**
   1. H.264编码器x264

转码核心是用动态连接库的方式使用x264，用到了x264.h, libx264.lib, libx264.dll三个文件，由于x264的代码不能用MS的编译器，要用Mingw环境编译，但不能生成libx264.lib文件，这里介绍一下libx264.lib的制作方法。

1. 在Mingw下编译x264，生成的文件用libx264-xxx.dll(其中xxx是x264的版本号)。
2. 去掉版本号，用VC自带工具dumpbin处理该dll 产生def文件：

Dumpbin libx264.dll /exports > libx264.def

1. 编辑libx264.def，参照x264.h文件，只保留头文件列出的函数，其余的删掉，开头换成：LIBRARY libx264 EXPORTS
2. 用VC自带工具lib生成.lib文件

Lib /DEF:libx264.def /MAHCINE:i386 /OUT:libx264.lib

* 1. 转码码参数配置文件参考pplive\_config.xml

1. **流程图**



