

# 基于 FFmpeg+VA-API 的 硬件加速渲染技术

.....  
成/都/虫/洞/奇/迹/科/技/有/限/公/司

2017/11

# 目录

---

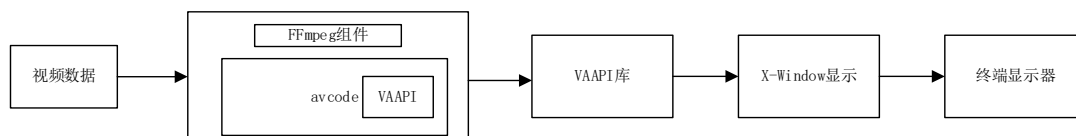
1. 写在前面 .....	3
2. 实现方案 .....	3
2.1 相关环境部署 .....	3
2.2 代码实现 .....	5
3. 测试结果 .....	6
4. 写在最后 .....	7

## 1. 写在前面

硬件加速即利用 GPU 来完成图形相关的操作，将 CPU 空闲出来处理其他事务，特别是在 CPU 性能不足的情况下，此类操作就更必要了。在视频处理上，DXVA 是微软定制的视频加速规范，而在 Linux 平台上则是由 NVIDIA 和 Intel 分别提供的 VDPAU 和 VAAPI 加速规范。灵跃桌面云利用上述 API 可以有效加速视频解码和图像处理，降低系统 CPU 占用，提升系统性能。

## 2. 实现方案

针对 Intel CPU X86 架构平台，可采用 FFmpeg+VAAPI 的硬解方案。其中，FFmpeg 用于处理各种格式的视频解码；VAAPI 则提供对硬件加速视频处理的访问，处理流程图如下。



### 2.1 相关环境部署

#### (一) 安装依赖包

```
# apt-get install autoconf automake debhelper ladspa-sdk libsoundio-dev libbluray-dev libbs2b-dev
libcaca-dev libcdio-paranoia-dev libdvdnav-dev libdvdread-dev libegl1-mesa-dev libenca-dev
libfontconfig1-dev libfreetype6-dev libfribidi-dev libgl1-mesa-dev libgsm-dev libgme-dev libgnutls28-
dev libgsm1-dev libguess-dev libharfbuzz-dev libjack-jackd2-dev libjpeg-dev liblcms2-dev liblircclient-
dev liblua5.3-dev libmodplug-dev libmp3lame-dev libnetcdf-dev libopenal-dev libopus-dev libopencore-
amrnb-dev libopencore-amrwb-dev libpulse-dev librtmp-dev librubberband-dev libschoedinger-dev
libsmbclient-dev libssh-dev libsoxr-dev libspeex-dev libtheora-dev libtool libtwolame-dev libuchardet-
dev libv4l-dev libva-dev libvorbis-dev libvo-amrwbenc-dev libvpx-dev libwaypack-dev libx264-dev
libx265-dev libx11-dev libxext-dev libxinerama-dev libxrandr-dev libxss-dev libxv-dev libxvidcore-dev
pkg-config python python-docutils x11proto-core-dev yasm zlib1g-dev
```

#### (二) 安装 libva

进入当前用户目录，并下载 libva 源码包：

```
# cd ~  
# git clone https://github.com/01org/libva.git  
进入 libva 文件夹中,  
# cd libva  
切换到 1.7.3 版本:  
# git checkout libva-1.7.3  
执行编译安装:  
./autogen.sh && ./configure && make && make install  
运行 vainfo 查看执行结果
```

```
libva info: VA-API version 0.39.4  
libva info: va_getDriverName() returns 0  
libva info: Trying to open /usr/local/lib/dri/i965_drv_video.so  
libva info: Found init function __vaDriverInit_0_39  
libva info: va_openDriver() returns 0  
vainfo: VA-API version: 0.39 (libva 1.7.3)  
vainfo: Driver version: Intel i965 driver for Intel(R) Ivybridge Mobile - 1.8.1  
vainfo: Supported profile and entrypoints  
VAProfileMPEG2Simple : VAEntryPointVLD  
VAProfileMPEG2Main : VAEntryPointVLD  
VAProfileH264ConstrainedBaseline: VAEntryPointVLD  
VAProfileH264Main : VAEntryPointVLD  
VAProfileH264High : VAEntryPointVLD  
VAProfileH264StereoHigh : VAEntryPointVLD  
VAProfileVC1Simple : VAEntryPointVLD  
VAProfileVC1Main : VAEntryPointVLD  
VAProfileVC1Advanced : VAEntryPointVLD  
VAProfileNone : VAEntryPointVideoProc  
VAProfileJPEGBaseline : VAEntryPointVLD
```

图中红框内表示当前所支持硬解的视频压缩格式。

### (三) 安装 FFmpeg

进入灵跃桌面云当前用户目录，并下载 FFmpeg 源码包

```
# cd ~  
# git clone https://git.ffmpeg.org/ffmpeg.git ffmpeg  
进入 FFmpeg 文件夹中,  
# cd ffmpeg  
执行编译安装  
./configure --disable-static --enable-shared --enable-gpl --enable-version3 --disable-debug  
--enable-vaapi &&make && make install  
最后执行 ffmpeg 查看执行结果。
```

```

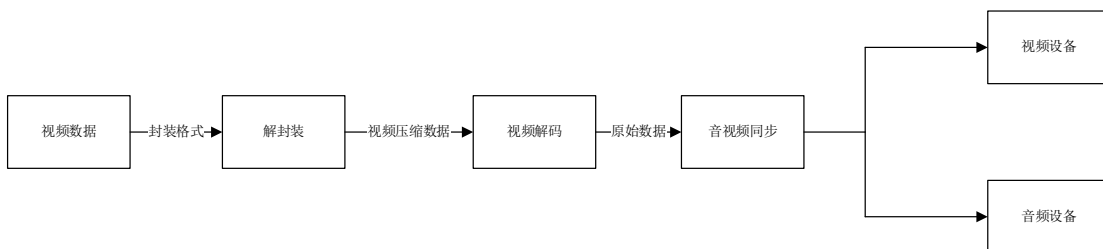
ffmpeg version 3.3.1 Copyright (c) 2000-2017 the FFmpeg developers
built with gcc 5.4.0 (Ubuntu 5.4.0-6ubuntu1~16.04.5) 20160609
configuration: --disable-static --enable-shared --enable-gpl --enable-version3 --disable-debug --enable-vaapi
WARNING: library configuration mismatch
avutil      configuration: --disable-static --enable-shared --enable-gpl --enable-version3 --disable-debug
avcodec     configuration: --disable-static --enable-shared --enable-gpl --enable-version3 --disable-debug
avformat    configuration: --disable-static --enable-shared --enable-gpl --enable-version3 --disable-debug
avdevice    configuration: --disable-static --enable-shared --enable-gpl --enable-version3 --disable-debug
avfilter    configuration: --disable-static --enable-shared --enable-gpl --enable-version3 --disable-debug
swresample  configuration: --disable-static --enable-shared --enable-gpl --enable-version3 --disable-debug
postproc    configuration: --disable-static --enable-shared --enable-gpl --enable-version3 --disable-debug
libavutil   55. 58.100 / 55. 58.100
libavcodec  57. 89.100 / 57. 89.100
libavformat 57. 71.100 / 57. 71.100
libavdevice 57.  6.100 / 57.  6.100
libavfilter  6. 82.100 /  6. 82.100
libswscale  4.  6.100 /  4.  6.100
libswresample 2.  7.100 /  2.  7.100
libpostproc 54.  5.100 / 54.  5.100
Hyper fast Audio and Video encoder

```

## 2.2 代码实现

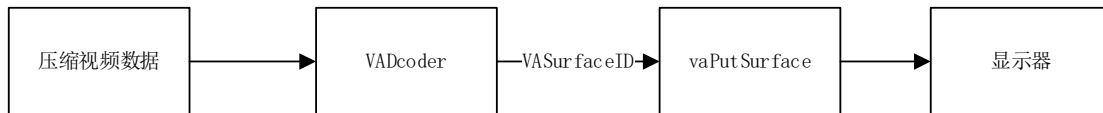
### (一) 视频播放原理

简单来说，视频播放器在播放本地视频文件需要经历以下步骤：解封装，解码，音视频帧同步，最终会把数据投递到对应的驱动设备，具体过程如下图所示：



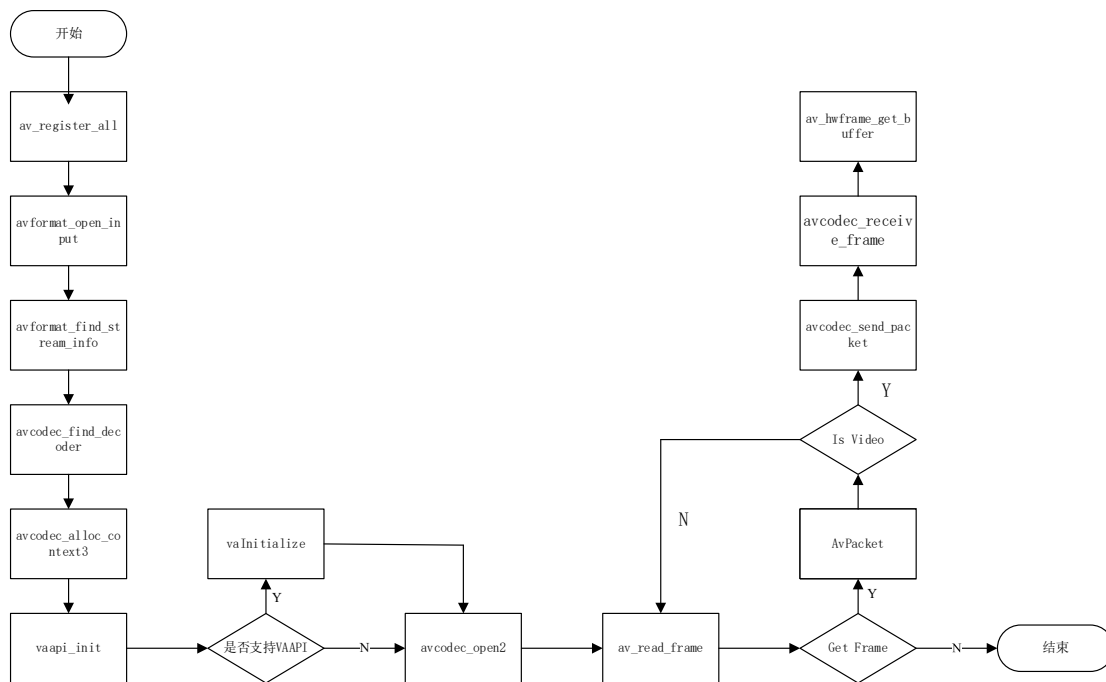
### (二) VA-API 解码流程

当程序把视频的压缩数据传递到 VA-API 解码模块时，会根据视频的压缩格式(如 H.264.MPEG-2 等)创建对应的 VADecoder，然后把数据送入 VADecoder 进行解码，最后由 vaPutSurface 来处理视频解码后的合成和渲染，具体过程如下：

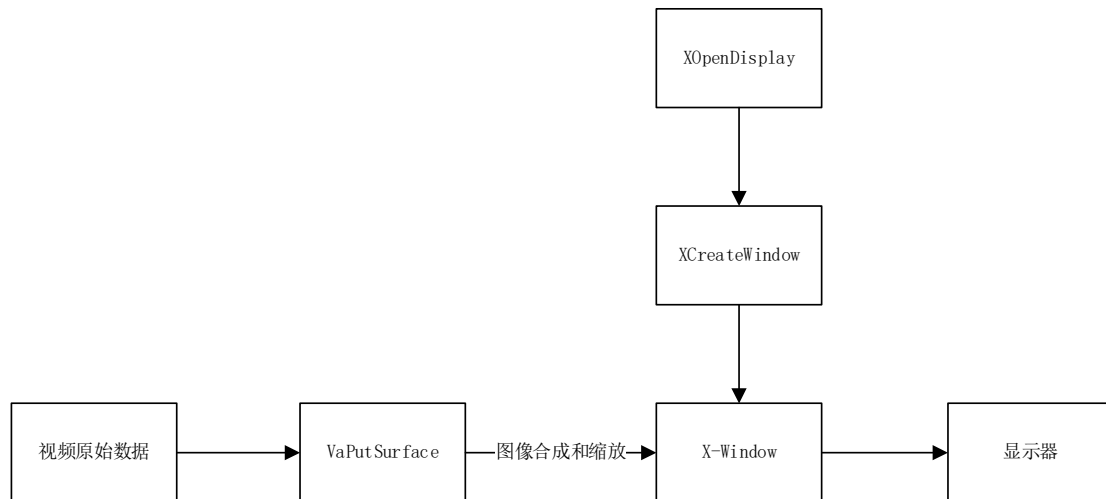


### (三) 具体实现

基于 VA-API 和 FFmpeg 的解码流程图如下：



由于灵跃桌面云使用的 FFmpeg 的版本是 3.3 版本，里面集成了 VA-API 中对视频的细节处理，所以只需要在 AVCodecContext 的 get\_format 回调函数中指定硬解参数，解码后的视频数据会存储在 GPU 内存中，通过 vaPutSrfuce 函数来处理后期的合成和图像渲染，下图为窗口创建和图像处理的过程图：



### 3. 测试结果

本次测试为对比使用 GPU 硬解和 CPU 软解的 CPU 占用率，视频参数：分辨率

1920\*1080、帧率 25FPS、编码格式 H.264。灵跃桌面云依据测试结果可以看出，使用 GPU 硬解能大大降低 CPU 使用率，同时画面输出更加清晰流畅。

解码类型	CPU 占用百分比
基于 VAAPI 的 GPU 硬解	4%
基于 CPU 软解	50%

#### 4. 写在最后

至此，灵跃桌面云基于 VAAPI 的硬件加速技术介绍完毕。在实际的应用场景中，还可进一步与视频重定向结合，关于视频重定向的结合应用将在后续更新，请持续关注灵跃桌面云公众号更新。