

# 基于 VDMAU 的 GPU 多视频解码<sup>①</sup>

刘其军, 罗明宇, 付燕平, 归 强

(广东粤铁瀚阳科技有限公司, 广州 501630)

**摘要:** 针对多视频解码 CPU 占用率高等问题, 本文研究了基于 VDMAU 的 GPU 视频并行处理方案, 通过集成 VDMAU 的 VdpDevice 和 VdpDecoder, 实现了 FFmpeg 多视频解码的 GPU 并行处理, 极大地降低了多视频解码显示过程中的 CPU 占用率. 测试结果表明, 本文提出的基于 VDMAU 的多视频解码处理方案可极大地提高系统的多视频解码处理能力.

**关键词:** VDMAU; GPU; 多视频解码; FFmpeg

## GPU Video Decoding Base on VDMAU

LIU Qi-Jun, LUO Ming-Yu, FU Yan-Ping, GUI Qiang

(Guangdong Railway & Sun Technology Co. Ltd, GuangZhou 510630, China)

**Abstract:** A GPU multi-video decoding implementation based on VDMAU is presented to solve multi-video display problem. VdpDevice and VdpDecoder is integrated, and the FFmpeg video is decoded with GPU parallel processing. The experimental results show that the CPU utilization is greatly reduced and the multi-video decoding processing scheme based on VDMAU can greatly improve the ability of video decoding.

**Key words:** VDMAU; GPU; multi-video decoding; FFmpeg

随着多媒体和网络技术的迅速发展, 数字视频显示技术已广泛应用于数字电视、网络直播、网络点播、视频会议、视频监控等诸多领域, 而近几年高清视频不断推广与普及, 视频高清化已是必然趋势<sup>[1]</sup>. 通常分辨率大于 720p(1280x720)的视频为高清视频, 当达到 1080p(1920x1080)时即为全高清视频<sup>[2]</sup>. 在高清视频编码方面, H.264 具有低带宽提供优质视频的能力, 其图像质量大大优于传统系统, 比 H.263 和 MPEG-4 提高编码效率约 50%<sup>[3]</sup>. 同等图像质量条件下, 采用 H.264 技术压缩后的数据量只有 MPEG2 的 1/8, MPEG4 的 1/3, 因此, H.264 已成为高清视频主流的编码格式, 但是其算法的复杂性高, 耗费在图形算法上的计算资源大, 目前主流 CPU 上解码单路 H.264 全高清视频的 CPU 占用率为 50%左右, 仅依靠 CPU 已经难以处理高清视频了<sup>[4]</sup>.

于是, GPU 被设计用来专为并行计算提供支持<sup>[5,6]</sup>, 随着 GPU 性能的大幅度提高以及可编程能力的发展,

可以将图形流水线的某些处理阶段以及某些图形算法从 CPU 向 GPU 转移, 从而减轻 CPU 的计算压力<sup>[7,8]</sup>. GPU 的计算优势越来越受到各方关注, 除专业图形应用外, 还广泛应用于通用计算领域, 即 GPGPU (General-purpose computing on graphics processing units)<sup>[9,10]</sup>.

为满足铁路指挥调度系统对多视频显示处理的需求, 本文研究了基于 VDMAU 的 GPU 多视频处理技术, 通过 GPU 并行处理实现对 FFmpeg 多视频的解码显示, 极大地降低了多视频处理的 CPU 占用率. 通过测试, 本文基于 VDMAU 的 GPU 多视频解码处理技术可极大地提高系统的多视频解码显示能力.

## 1 VDMAU 技术

### 1.1 VDMAU 背景

VDMAU (Video Decode and Presentation API for Unix)是由 NVIDIA 开发的基于 UNIX 和类 UNIX 系统

① 基金项目:广东省科技计划项目(2012A080102003);广东省省部产学研结合项目(2012B090500012)

收稿时间:2013-04-28;收到修改稿时间:2013-05-31

(主要包括 Linux、FreeBSD 和 Solaris)下的视频硬件解码技术,它包括视频解码、视频后期处理、图像合成、解码后视频图像的显示等<sup>[1]</sup>.它可以支持的最大视频宽度与高度为 2048 像素,最小视频宽度与高度为 48 像素,并且所有编码限制在最大 8192 个宏区块内(对于 VC-1/WMV 是 8190 个).

VDPAU 支持系统对 MPEG-1、MPEG-2、MPEG-4 ASP、VC-1/WMV9 和 H.264 视频解码的完全加速,但不支持 MPEG-4 ASP 的运动补偿和数据分块.目前,NVIDIA 的 GeForce 8 系列以上的显卡均可支持 VDPAU 对视频的硬件解码.

### 1.2 组成模块

VDPAU 主要由 Core API 和 Window System Integration Layer 两个模块组成,其中 Core API 模块主要负责 VDPAU 库的初始化、视频解码、解码成功后的处理及显示(如图像颜色空间的转化、图像合成噪声处理、去隔行、OSD 显示等),Window System Integration Layer 模块主要是为了快速与 X Window 集成显示.VDPAU 主要的数据类型有 Device Types、Surface Types 和 Transfer Types,对应的主要数据结构如下:

#### ① Device Types

VdpDevice: VDPAU 的根对象,所有应用必须建立在其基础之上.必须先创建它,才能进行后面的一切处理.

#### ② Surface Types

VdpVideoSurface: 存储解码成功后的视频帧,格式为 YcbCr. VdpOutputSurface: 存储 RGB4:4:4 数据,主要用于视频解码后的处理、合成、渲染等操作.

VdpBitmapSurface: 存储只读的位图数据,用于 OSD 的显示.

#### ③ Transfer Types

VdpDecoder: 解码器,处理压缩的视频数据,生成解码后的图像. VdpOutputSurface: 用于视频解码后的处理、合成、图像渲染. VdpVideoMixer: 用于视频处理、去隔行、图像合并等. VdpPresentationQueue: 用于生成基于显示面的时间戳,类似于视频中 PTS 的概念.

### 1.3 视频解码

VDPAU 实现视频解码的过程如下:应用程序从视频文件或网络视频流中读取视频数据,根据视频数据的格式(如 MPEG-1、MPEG-2 和 H.264 等)创建对应

的 VdpDecoder,然后把视频数据送入 VdpDecoder 进行解码,如果需要进行后期效果处理(如图像的缩放等)则调用 VdpVideoMixer 进行处理.需要 OSD 信息时,可通过 VdpBitmapSurface 来实现.VDPAU 实现视频解码的流程如图 1 所示.

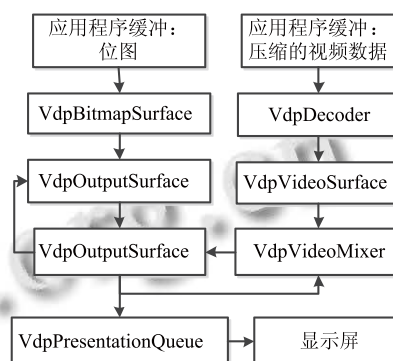


图 1 VDPAU 实现视频解码的流程

## 2 基于VDPAU的GPU视频解码

### 2.1 方案设计

虽然 VDPAU 提供了视频解码的硬件加速技术,但是它未能对所支持的视频编码格式进行解码集成,也就是说,它只提供了一个关于 GPU 视频解码的技术平台,对于具体的编码格式(如 MPEG-1、MPEG-2 和 H.264 等)的解码处理,需要根据解码算法自定义代码实现.很显然逐一实现 MPEG-1、MPEG-2 和 H.264 等解码算法在时间、稳定性、成本上是不明智的.因此,寻找一个稳定可靠的视频解码框架来代替全部自定义实现是非常有必要的.这里将采用集音视频解码于一身、且支持几乎所有格式的开源音视频解决方案 FFmpeg.

FFmpeg 是目前流行的音视频编码转换方案,集成了音视频录制、转换以及流化等功能,主要应用于 Windows、Linux、MAC OS X、BSD 等操作系统<sup>[12]</sup>,由 avutil、avcodec、avformat、avdevice、avfilter、swscale 和 swresample 等源码库组成<sup>[13]</sup>,其中 avutil 是 FFmpeg 跨平台移植的基础,它定义了安全的可移植的字符串函数、随机数据发生器及数据结构,同时提供了额外的数据函数、密码技术和多媒体相关的功能(如像素和样本格式枚举等);avcodec 提供了通用的编解码器;avformat 提供了音频、视频和字幕流多路复用和多路分解的通用框架,包括多个多媒体容器格式的解复用器;avdevice 提供了渲染和捕获通用的多媒体输入、

输出设备的通用框架, 如 Video4Linux2、VfW、Dshow、ALSA 等; avfilter 提供了通用的音频/视频过滤框架; swscale 支持高度优化过的图像缩放、色彩和像素格式转换等; swresample 支持高度优化的音频重采样和采样格式转换等。

本文提出的基于 VDPAU 的 GPU 多视频解码技术将融合 FFmpeg 与 VDPAU, 充分利用 FFmpeg 强大的解码处理能力来解析视频数据的编码格式, 并根据编码格式选择相应的解码模块, 在解码处理时与 VDPAU 相结合, 从而快速而高效地实现了基于 VDPAU 的 GPU 多视频解码, 其视频解码流程如图 2 所示。

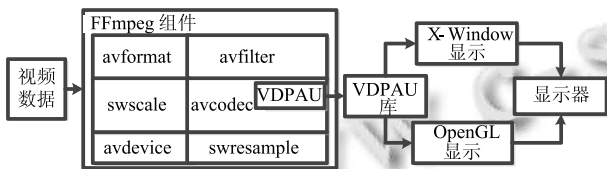


图 2 基于 VDPAU 的 GPU 多视频解码流程

### 2.2 方案实现

虽然 FFmpeg 的 libavcodec/vdpau.c 中集成了 VDPAU 对 MPEG-1, MPEG-2, VC-1/WMV3 和 H.264 等视频的 GPU 处理解码, 但是它没有集成 VDPAU 库的初始化, 没有实现用 VDPAU 库来进行 GPU 解码, 无法把 VDPAU 解码后的图像显示出来。因此, 关于 VDPAU 的 GPU 解码过程, 必须自定义代码实现。结合整个过程, 具体步骤归纳如下:

#### ① 编译支持 VDPAU 的 FFmpeg

由于在默认情况下, FFmpeg 并不支持 VDPAU, 因此需要重新编译支持 VDPAU 的版本。具体是通过 git clone git://source.ffmpeg.org/ffmpeg.git ffmpeg 下载最新的 FFmpeg 源码, 在编译配置时加上 VDPAU 选项: ./configure --enable-vdpau 即可。

#### ② 获取 VDPAU 库

VDPAU 库可由源码编译和直接下载两种方式获取。对于源码编译方式, 可以访问 <http://cgib.freedesktop.org/~aplattner/libvdpau/> 下载源码, 然后按照 configure, make, make install 三步进行编译即可<sup>[14,15]</sup>。

对于直接下载方式, 与使用的 linux 发行版本有关, 对于 Centos 5.8, 可直接用 yum install libvdpau-devel.x86\_64 下载安装即可。

#### ③ FFmpeg 中集成 VDPAU 解码

要在 FFmpeg 中体现 VDPAU 的 GPU 解码处理, 需要自定义实现并替换 FFmpeg 解码处理方法。具体是自定义实现 FFmpeg 视频解码中 AVCodecContext 对象必须用到的 get\_buffer、release\_buffer 和 draw\_horiz\_band 等三个方法, 其中在 get\_buffer 中把解析后的数据存储于 VDPAU 的 VdpVideoSurface 对象中, 在 draw\_horiz\_band 中实现了 VdpDecode 和 VdpVideoMixer 的创建及 VdpDecode 的解码和 VdpVideoMixer 解码后的后期处理, 最后, 通过 release\_buffer 释放了创建的 VdpVideoSurface 等相关对象。

由于 VDPAU 与 FFmpeg 都是基于 C 语言开发的库, 并没有体现面向对象的优点。所以为了简化它的使用, 建议用面向对象的方法进行了二次封装。新建类 VDPAUConext 类封装实现 VDPAU 的初始化, 如: VdpDevice 对象的创建、释放, 同时整合所有 VDPAU 中的公有方法, 方便调用。新建类 VDPAUDecode 类封装实现 VdpDecoder 的创建与销毁、视频解码和 VdpVideoMixer 的图像变换等处理。新建类 FFmpegDecode 类封装 FFmpeg 视频解码流程, 如视频数据的输入和视频解码的基本设置等。利用类 FFmpegDecode 与 VDPAUDecode 的结合来实现 FFmpeg 与 VDPAU 的集成, 视频数据经 VDPAUConext、VDPAUDecode 的 GPU 并行处理后, 完成视频的解码显示。因此, 基于 VDPAU 的 GPU 多视频解码实现流程如图 3 所示, 这里并未画出默认解码方式, 这里关注的是 VDPAU 解码。

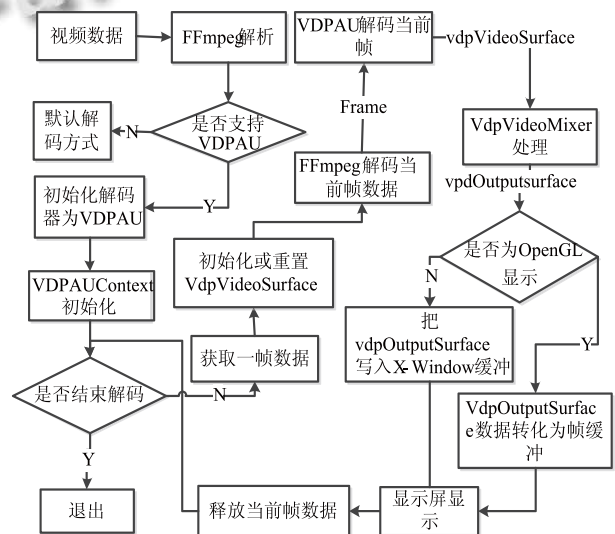


图 3 基于 VDPAU 的 GPU 多视频解码的流程

### 2.3 测试结果

为了验证基于 VDP AU 的 GPU 多视频解码的有效性,通过普通 CPU 解码和基于 VDP AU 的 GPU 解码进行了多视频解码对比测试,同时,为了验证 VDP AU 能对 MPEG-1、MPEG-2、WMV3 和 H.264 等编码格式视频进行硬件加速,分别选择 MPEG-1、MPEG-2、WMV3 和 H.264 等编码格式的视频文件进行了解码测试。在单核 CPU 的情况下,测试结果如图 4 所示:

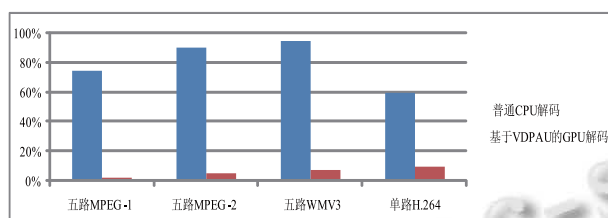


图4 CPU占用率测试结果

从图4可以明显地看出:基于VDP AU的GPU视频解码技术与普通CPU解码相比,大大降低了系统的CPU占用率。

值得一提的是,由于测试中的H.264编码是1080p视频,普通CPU无法软解码五路1080p视频,因此,仅列出了单路视频进行比较,但采用基于VDP AU的GPU多视频解码后,可以支持五路1080p视频的解码,CPU总体占用率大约为50%左右。

### 3 结语

本文研究了基于VDP AU的GPU视频解码技术,同时提供了FFmpeg与VDP AU结合的视频解码实现方案,既充分利用了FFmpeg对主流视频格式的支持,又实现了基于VDP AU的GPU视频硬件加速技术,充分发挥了GPU等硬件对视频解码的处理效能。针对不同编码格式的多视频解码和1080p全高清视频解码的测试结果进一步证明基于VDP AU的GPU多视频解码方案能有效地降低多视频解码的CPU占用率,提高系统的多视频解码处理能力,为多路视频的解码处理提供了有效的解决方案。

### 参考文献

- 1 余和初,聂蓉.高清视频监控技术的现状及发展趋势.中国公共安全(综合版),2009,(12).
- 2 任广杰,朱启东,曹宁.高清视频监控中的若干关键技术分析.电信科学,2011,(2):119-123.
- 3 李超,柴文磊,刘劲松.高清视频会议系统技术浅析.信息安全与技术,2010,(10):50-51.
- 4 苏俊峰.基于GPU的高效视频解码技术研究.南京:南京邮电大学.2012.
- 5 张朝晖,刘俊起,徐勤建.GPU并行计算技术分析与应用.信息技术,2009,(11):86-89.
- 6 吴恩华,柳有权.基于图形处理器(GPU)的通用计算.计算机辅助设计与图形学学报,2004,16(5):601-612.
- 7 侯兴松,刘大齐,盛凯,顿玉洁.H.264并行编码中负载平衡方法.中国图象图形学报,2012,17(8):911-918.
- 8 杨鑫,许端清,赵磊.基于多核架构的大图像实时浏览技术.中国图象图形学报,2011,16(2):152-160.
- 9 王成良,谢克家,刘昕.多核图像处理并行设计范式的研究与应用.计算机工程,2011,37(14):220-222.
- 10 卢风顺,宋君强,银福康,张理论.CPU/GPU协同并行计算研究综述.计算机科学,2011(3):5-9.
- 11 Dong M, Qiu RC, Wei RP, Bi S, Cai WY, Xin JY. New high-definition video player method based on GPU technology. Cyber Technology in Automation, Control, and Intelligent Systems(CYBER). 27-31 May 2012
- 12 尹璐洁,吴哲夫,覃亚丽.基于Blackfin DSP的FFmpeg音视频应用 2008年中国高校通信类院系学术研讨会论文集(上册).2009-03-07.
- 13 Major A, Ying Y, Nousias I, Milward M, Khawam S, Arslan T. H.264 decoder implementation on a dynamically reconfigurable instruction cell based architecture. SOC Conference, 2006 IEEE International 24-27 Sept. 2006.
- 14 余伯山.Linux系统管理与网络管理.北京:清华大学出版社.2010.
- 15 Matthew N, Stones R.陈健,宋健建译.Linux程序设计.北京:人民邮电出版社.2010.