

# 英特尔® SVT-AV1 助力腾讯视频云 提升 AV1 编码性能与视频质量



## 目录

1. 前言	1
1.1 背景	1
1.2 目标人群	1
1.3 注意	1
2. 英特尔® SVT 介绍	2
2.1 SVT 技术介绍	2
2.2 英特尔® SVT 的架构	2
2.3 英特尔® SVT 的编码流程	3
2.4 使用英特尔® SVT 的益处	3
3. 互联网厂商率先使用英特尔® SVT	4
3.1 腾讯视频云媒体处理采用 SVT-AV1 编码技术	4
4. 总结和建议	5
5. 参考资料	5
5.1 缩略词	5
5.2 参考文献	5

## 1. 前言

### 1.1 背景

当前, 各种互联网直播、点播及小视频应用的爆炸式发展, 使得视频数据的转码正经历着需求快速增长而驱动的重大变革, 现有的基础设施总是不能满足发展, 迫切需要一种新的视频编码技术方案, 帮助数据中心以有限的资源满足更多应用场景的需求。

英特尔® SVT (Scalable Video Technology, 可扩展性视频技术) 是英特尔基于软件的视频编码架构, 可使编码器在第二代英特尔® 至强® 可扩展处理器上实现性能、延迟和视觉质量之间的最佳平衡, 且允许编码器根据质量和延迟来调整应用程序的性能目标。因为高度并行的软件架构设计, SVT 具有高度的并行性和可扩展性。通过针对指定的第二代英特尔® 至强® 可扩展处理器做的一系列算法上的优化, 英特尔® SVT 具备了很好的性能。目前 SVT-HEVC, SVT-VP9 以及 SVT-AV1 都使用宽松的开源协议 (详细请参见相应 Github 上的协议许可文件), 从而让使用者可以拥有低成本的转码解决方案, 并且减少视频应用的开发强度和开发时间。

### 1.2 目标人群

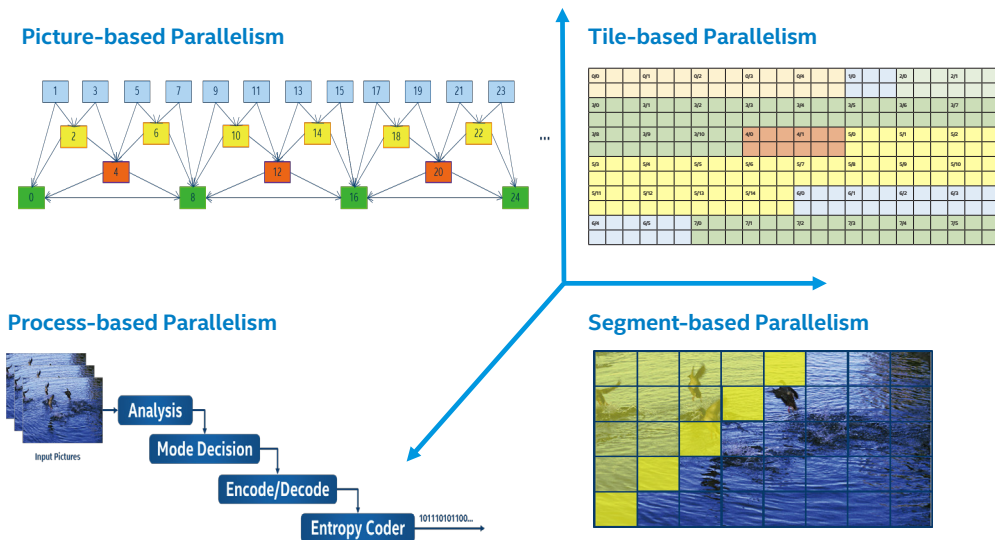
本文档的目标读者是已经熟悉或了解视频业务的开发厂商、开发者和相关运维人员。包括:

1. 视频直播、点播、转码系统工程师
2. 视频业务售前架构师、运维工程师
3. 互联网视频业务相关读者

### 1.3 注意

本资料如有更改, 恕不另行通知。

英特尔® SVT 编码器的性能取决于系统配置, 可能需要与您相关的系统硬件、系统软件特性相关。更多信息请访问 [intel.com](https://www.intel.com)。



图一 高度并行的英特尔® SVT 架构

## 2. 英特尔® SVT 介绍

### 2.1 SVT 技术介绍

视频编码标准发展了三十年，大的框架基本没有变过，编码的流程也类似，都是针对时域空域的相关性做有损压缩，区别就是会越来越复杂，用算力来换压缩率。目前视频编码已经成为大多数视频云应用程序的关键部分。将源视频内容压缩为大小尽可能少的字节，且需要编码在最短的时间内完成，还不会显著影响视频质量，是目前视频应用的普遍需求。许多视频压缩技术和标准（例如 MPEG-2、AVC、HEVC、VP9 和 AV1）已经被广泛地应用在各种视频业务中，但当前的问题是，标准兼容编码器可能非常复杂，需要大量的计算和内存资源，且对于某些高质量和高分辨率的视频应用，硬件编码器（基于 ASIC 或 SoCs）还可能力不从心。而解决这些问题的有效途径，就是应用高性能和高质量的软件（如基于第二代英特尔® 至强® 可扩展处理器）编码器。

为了实现这一目标，英特尔提供了一个基于软件的高效视频编码器 SVT，有望在性价比、质量和可扩展视频技术方面实现充分平衡，这将使基于软件的视频编码器能够有效地应对不同视频应用程序的各种需求。

### 2.2 英特尔® SVT 的架构

大家如果有使用过开源的 SVT-HEVC、SVT-VP9 或者 SVT-AV1 会发现它们的速度非常快，无论快速档慢速档都可以有效利用多个 CPU 核。这得益于英特尔® SVT 的高度并行化的框架，主要体现在以下几点：

#### 1 时域上的并行

最简单的方法是根据 GOP 在时域上并行编码，不同线程处理不同的 GOP，但是这种方法比较粗糙，不够有效，同时并行度也不够高。英特尔® SVT 采用分层编码的方式，同时维护了一套帧之间依赖关系的逻辑，所有满足依赖关系的帧都可以同时进行编码。如图一所示分为四层，最开始先编第 0 帧，然后编第 8 帧；而后第 4 帧和第 16 帧就可以同时开始编码，以此类推。在实际应用中，一些客户是允许有一到两秒钟的延迟的，如果是 60fps 也就意味在编码器端我们最多能缓存 120 帧，这样会有大量的满足依赖关系的帧可以同时开始编码流程。

#### 2 空域上的并行

从 H.265 开始，引入了 tile 的概念，tile 之间可以并行处理，没有依赖关系。但是引入 tile 会对视频质量造成一定的影响。英特尔® SVT 在内部实现时提出一个 Segment 的概念，简单来讲，它相当于 H.265 中 WPP 的扩展。每个 Segment 可以是若干个 LCU，也可以是一个 LCU，也就是 H.265 里的 WPP。每一个 Segment 分段会有相应的依赖关系（左方和上方）。在处理的时候，所有满足依赖关系的 Segment 都会从线程池里取出线程来处理。每个 segment 内部是以 LCU 为单位，用光栅扫描顺序处理。这种 Segment 的处理方式是无损的，同时也是可以跨tile 的，可以和 tile 结合起来使用，从而有效地满足某些大分辨率低延迟的编码需求。

#### 3 编码流程上的并行

在整个编码过程中，有些分析可以在原始帧上做，没有什么依赖关系。有些分析处理需要知道参考帧，而有些处理需要在

重构帧上进行。从预处理分析到模式选择，再到编码、熵编码，整个编码的并行度会进一步下降。英特尔® SVT 针对不同编码阶段的并行做了最大限度的优化。从线程的角度来说，有的模块是负责处理数据的，有的线程是用来控制负责同步编码器内部状态的，不同的模块可以运行在不同的线程上 (绑定在不同的 CPU core 上) 处理满足依赖关系的 frame 或者 block，这种灵活的设计能够保证英特尔® SVT 有效地利用所有的 CPU 运算性能。

### 2.3 英特尔® SVT 的编码流程

英特尔® SVT 数据流的处理过程如图二所示。左边部分叫做 Open-Loop，可以在原图像上做，不需要等重构帧，只要输入图像满足参考关系就可以处理。右边部分叫做 Closed-Loop，需要在重构图像上做。图中有多个框的 kernel 代表多个线程。如图第一个 kernel 是负责资源协调的工作，只有一个线程；第二个 Picture Analysis Kernel 是多线程，会做很多图像静态分析例如计算方差等工作。在处理的时候会将一帧图像从空域上分成多个区域，交由不同的线程处理；后面的 Picture Decision Process 中会决定参考帧依赖关系，同时会做场景切换检测 (因为场景切换检测可能会打乱原来的 GOP)。在确定参考帧前后关系后，就开始做运动估计。和 Picture Analysis Kernel 类似，也是空域上切成多个区域，分给不同的线程并行处理。之后会经过 BRC 模块和一个时域分析的模块，检测是否有运动、背景等等。接下来便来到一个叫 Picture Manager Kernel 的重要部分，它会解决重构帧的依赖关系。所有满足重构帧依赖关系的帧会进行 MD (Mode Decision)、编码、熵编码、packetization 的过程。中间的 filter 对 H.265、AV1、AVS3 会

略有不同。在进行 MD 和 Encode-Decode 的时候，会利用前面介绍的 Segment 来提高并行度。可以看出，在整个编码流程中，英特尔® SVT 在各个阶段都最大化的进行并行处理。

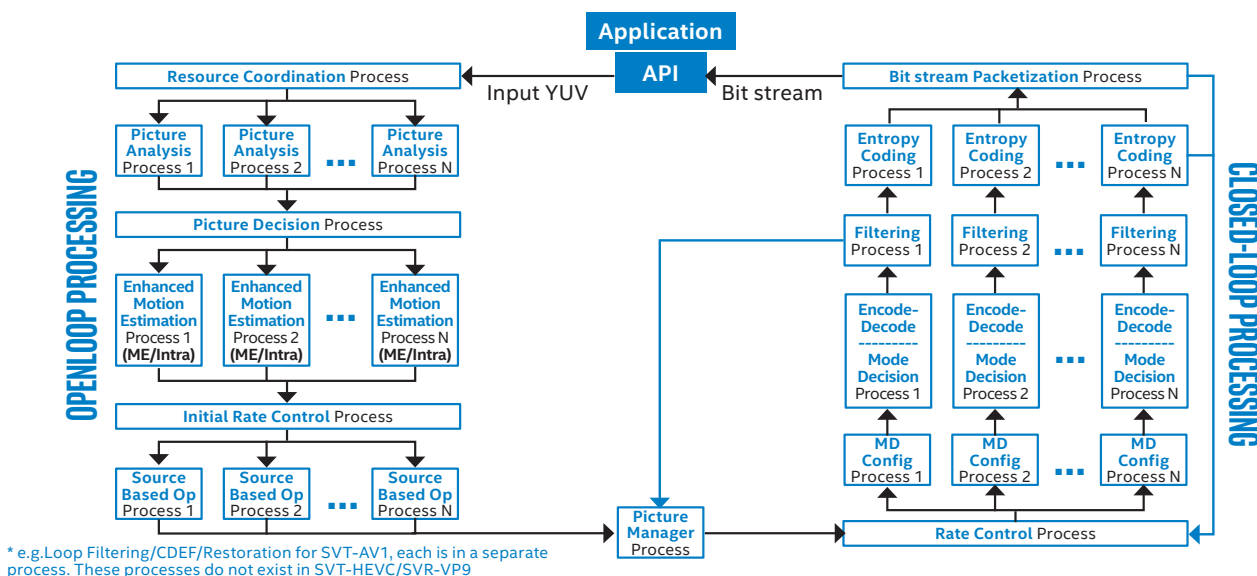
### 2.4 使用英特尔® SVT 的益处

开发 SVT 是为了在第二代英特尔® 至强® 可扩展处理器上高效处理和编码多分辨率视频内容，并提升视频转码解决方案的可扩展性能，包括：

英特尔® SVT 在性能、延迟和视频质量之间实现了极好的平衡。事实上，SVT 编码器具有多档性能和质量的预设值，能够满足各种质量需求下的视频云应用程序，包括视频点播 (VOD)、广播、流媒体、监视、云图形和视频会议等。

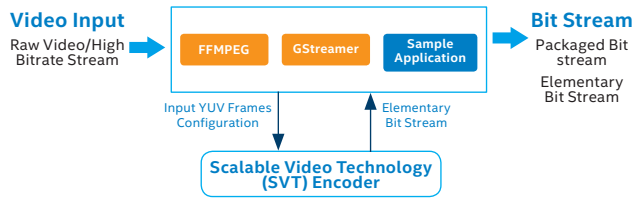
SVT 针对第二代英特尔® 至强® 可扩展处理器和英特尔® 至强® D 处理器进行了高度优化，特别是基于英特尔® 高级矢量扩展 512 (英特尔® AVX-512) 指令进行了优化，以提高性能。基于大多数数据中心都使用第二代英特尔® 至强® 可扩展处理器，SVT 针对第二代英特尔® 至强® 可扩展处理器的优化提升了视频编码工作负载的效率。SVT 编码器的高度并行化特性可充分利用第二代英特尔® 至强® 可扩展处理器的多核特性。因此，视频云服务提供商可以利用其现有的基础设施来交付优化的视频编码工作。

英特尔® SVT 提供了基于通用计算的实时编码解决方案。基于 CPU 的编码方案提供了更好的灵活性，包括易于集成升级，灵活地创建各种视频云应用，以及作为其他视频处理工具和组件提供接口，简化开发完整的端到端视频应用的过程。



\* e.g. Loop Filtering/CDEF/Restoration for SVT-AV1, each is in a separate process. These processes do not exist in SVT-HEVC/SVR-VP9

图二 SVT 的数据流框图



图三 英特尔® SVT 与应用程序的接口示意

### 3. 互联网厂商率先使用英特尔® SVT

#### 3.1 腾讯视频云媒体处理采用 SVT-AV1 编码技术

AV1 (Alliance for Open Media Video 1) 是由 AOM (Alliance for Open Media, 开放媒体联盟) 制定的一个开源、免版权费的视频编码格式, 目标是解决 H.265 昂贵的专利费用和复杂的专利授权问题, 并成为新一代领先的免版权费的编码标准。腾讯已于 2019 年 10 月正式加入开放媒体联盟 (Alliance for Open Media, AOMedia), 推进视频 AV1 标准商业化, 并成为了董事会成员之一。

SVT-AV1 在 V0.8 版本之后, 引入了更多的 AVX2 和 AVX-512 优化, 同时还添加了更多的档位, 并对单核执行做了优化。在算法上, 通过对分区划分以及模式选择的优化, 能够在保证质量的情况下进一步提高编码的速度。

目前, 国内常见的直播协议有 RTMP、HLS/DASH、HTTP-FLV 等, 其中, FLV (Flash Video) 格式简单轻量, 不需要大的媒体头部信息。整个 FLV 由 FLV Header, FLV Body 以及其它 Tag 组成。一般而言, CDN 下行是 HTTP 长连接并且加载速度快; 相比 RTMP 协议, HTTP-FLV 因为基于 HTTP/80 传输, 所以它能够很好地穿透防火墙, 有效避免被防火墙拦截, 可利用 HTTP 302 跳转支持灵活调度/负载均衡, 并且可以使用 HTTPS 加密传输。而国外主流视频协议是 HLS/DASH, 最大的不同在于这并不是一个请求完整的数据流, 其会在服务器端将流媒体数据切割成连续且时长较短的 ts/fmp4 小文件, 并通过 M3U8/MPD 索引文件按序访问 ts/fmp4 文件。由于播放器缓存以及 ts/fmp4 分段生成最小 GOP 等限制, 导致直播时延大, 平均时延在 10s 以上。最新的 CMAF/LHLS 通过 chunk 传输、将切片粒度减小到帧级别等技术以减小延迟, 其原理和 FLV 长连接 tag 类似。FLV 规范不支持 HEVC(H.265)/AV1, FFmpeg 社区对 FLV 的新视频编码算法 CodecID 也没有新增定义支持, 腾讯视频云 T-FFmpeg (腾讯视频云的 FFmpeg 维护版本) 正在推动社区支持封装/解

封 H.265/AV1 的 FLV 的补丁。目前, 国内各直播 CDN 厂商基本支持 H.265 的封装和解封, 而腾讯视频云则针对 FLV 支持 AV1 进行了一系列优化。

FLV 新增 CodecID: 0xD 为 AV1 定义, 新增 FLV AV1 sequence head 定义:

```

{
    uint8_t obuheader; //自定义 obu header 0x02, forbidd(0), obutype(0), extern(0), size field(0), reserved(0)
    uint16_t width; //画面宽
    uint16_t height; //画面高
    uint8_t profile; //编码 profile id 0 reserved
    uint8_t frame_rate; // 编码帧率
    uint8_t marker:1; //always 1
    uint8_t version:7; //1
    uint8_t seq_profile:3; // FF_PROFILE_AVI_MAIN/FF_PROFILE_AVI_HIGH/FF_PROFILE_AVI_PROFESSIONAL
    uint8_t seq_level_idx_0:5; // codec level 0-31
    uint8_t seq_tier_0:1; // codec tier
    uint8_t high_bitdepth:1; //1 10bits 0 8bits
    uint8_t twelve_bit:1; //1 12bits 0 10bits
    uint8_t monochrome:1; //1 gray 0 with uv
    uint8_t chroma_subsampling_x:1; //字段指示序列头 OBU 的子采样 x 值
    uint8_t chroma_subsampling_y:1; //字段指示序列头 OBU 的子采样 y 值
    uint8_t chroma_sample_position:2; //指定子采样流的采样位置
    uint8_t padding; //always 0
    {
        Sequence head obu buf; //sequence head obu buf
    }
}
    
```

FFmpeg FLV 封装和解封主要修改点:

封装 (flv\_write\_codec\_header/flv\_write\_packet):

```

else if(par->codec_id == AV_CODEC_ID_AVI) {
    avio_w8(pb, 0x02);
    avio_w16(pb, par->width);
    avio_w16(pb, par->height);
    avio_w8(pb, par->profile);
    if(par->sample_aspect_ratio.den == 0)
    {
        avio_w8(pb, 0);
    }
    else
    {
        avio_w8(pb, par->sample_aspect_ratio.num/par->sample_aspect_ratio.den);
    }
    ff_isom_write_avic(pb, par->extradata, par->extradata_size);
}
    
```

```

if(par->codec_id == AV_CODEC_ID_AVI)
{
    int32_t try_size = (pkt->data[0] << 24) | (pkt->data[1] << 16) | (pkt->data[2] << 8) | (pkt->data[3]);
    if(try_size != (pkt->size - 4))
    {
        uint8_t *size_buf = av_malloc(pkt->size - 4);
        int32_t size_len = pkt->size - 4;
        size_buf[0] = (pkt->size >> 24)&0xFF;
        size_buf[1] = (pkt->size >> 16)&0xFF;
        size_buf[2] = (pkt->size >> 8)&0xFF;
        size_buf[3] = (pkt->size)&0xFF;
        memcpy(size_buf + 4, pkt->data, pkt->size);
        if(pkt->buf)
        {
            av_buffer_unref(&pkt->buf);
        }
        pkt->data = NULL;
        pkt->size = 0;
        av_packet_from_data(pkt, (uint8_t*)size_buf, size_len);
    }
}
}
    
```

在云点播方面, AV1 在容器格式 MKV/TS/MP4 都有标准定义, FFmpeg 社区在 4.1.x 以上对 AV1 标准容器格式 MKV/MP4/TS 支持了封装/解封, 腾讯视频云点播已经支持了 AV1 包括转码、编辑、识别、审核等全路径的媒体处理。

具体来说, AV1 可以让用户在较低的带宽条件下享受较好的画质体验。对比现在广泛使用的 AVC 标准, 在相同画质下可以大大节省带宽成本。



目前, 腾讯云也是国内首家直播+点播同时支持 AV1 视频处理业务的公有云厂商。腾讯云一直以来都潜心深耕音视频领域, 不断完善自身产品体系, 并结合用户业务场景不断创新, 为用户带来极速高清、画质增强、老片修复等技术产品。

## 4. 总结和建议

通过充分利用多核处理器架构的特性, 以及通过参数化在性能和视频质量之间实现的更佳平衡, 英特尔® SVT 已成为各种视频应用程序的优选编码器。

SVT 编码器经过英特尔软件团队的大力优化, 能够基于第二代英特尔® 至强® 可扩展处理器高效运行, 并利用多核的并行性, 大幅提升处理器编码的性能。

在中国领先的互联网服务提供商腾讯实例上运行的英特尔® SVT 编码器, 已经证明了其性能卓越且开发高效, 可在实现与其他编码器类似或更好的视频质量的同时, 成本更低, 效率更佳。

## 5. 缩略词和参考文献

### 5.1 缩略词

Term	Definition
SVT	Scalable Video Technology
CLX-SP	Cascade Lake – Scalable Performance (Intel® Xeon® Processors with standard/advanced RAS used in main stream two socket server)
CSP	Cloud Service Provider
CDN	Content Delivery Network
AVC	Advanced Video Coding
AVX	Advanced Vector Extensions
OTT	Over The Top
MPEG	Moving Picture Experts Group
ASIC	Application Specific Integrated Circuit
GOP	Group of Pictures

### 5.2 参考文献

1. SVT-HEVC下载地址: <https://github.com/OpenVisualCloud/SVT-HEVC>
2. SVT-HEVC问题反馈地址: <https://github.com/OpenVisualCloud/SVT-HEVC/issues>
3. SVT-AV1下载地址: <https://github.com/OpenVisualCloud/SVT-AV1>
4. SVT-AV1问题反馈地址: <https://github.com/OpenVisualCloud/SVT-AV1/issues>

#### 法律声明:

英特尔技术特性和优势取决于系统配置, 并可能需要支持的硬件、软件或服务得以激活。产品性能会基于系统配置有所变化。没有任何产品或组件是绝对安全的。更多信息请从原始设备制造商或零售商处获得, 或请见 [www.intel.com](http://www.intel.com)。

本文并未(明示或默示、或通过禁止反言或以其他方式)授予任何知识产权许可。

英特尔未做出任何明示和默示的保证, 包括但不限于, 关于适销性、适合特定目的及不侵权的默示保证, 以及在履约过程、交易过程或贸易惯例中引起的任何保证。

英特尔、英特尔标识以及其他英特尔商标是英特尔公司或其子公司在美国和/或其他国家的商标。

©英特尔公司版权所有