

遨游"视"界 做你所想 Explore World, Do What You Want

哔哩哔哩H.265编码器 在直播和点播的实践和应用

> 叶天晓 技术专家 视频云 Bilibili





遨游"视"界 做你所想 Explore World, Do What You Want

LiveVideoStackCon 2019 深圳

2019.12.13-14



成为讲师: speaker@livevideostack.com

成为志愿者: volunteer@livevideostack.com

赞助、商务合作: kathy@livevideostack.com







B站 哔哩哔哩自研H.265编码器

PPTV 视频算法组负责人

七牛 七牛直播初始成员

虹软(Arcsoft) 视频算法组

新加坡南洋理工大学 多媒体技术实验室

1. B站为何要做自有视频编码器

- 2. 怎样做视频编码器
- 3. 自有编码器现状
- 4. 视频编码器针对B站直播业务的优化
- 5. 视频编码器针对B站点播业务的优化

B站点播业务



2018.01: UP主激励计划

2018.02: B站清晰度升级, 解锁60fps,

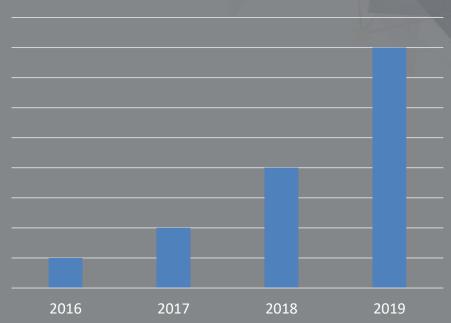
支持最高码率6000kbps

2018.08: 支持DASH来多分辨率切换

2018.12: 最大上传大小从4GB增加到8GB

2019.06: B站4K画质上线





B站直播业务



北京 2019 遨游"视"界 做你所想 Explore World, Do What You Want





时长 更长 带宽成本

存储成本

算力成本

画幅 更大 SCHOOL STATE OF THE SECOND SEC

种类

更多

码率即成本

开源编码器

x265性能

云厂商

转码费用

转码

数据挖掘

- 1. B站为何要做自有视频编码器
- 2. 怎样做视频编码器
- 3. 自有编码器现状
- 4. 视频编码器针对B站直播业务的优化
- 5. 视频编码器针对B站点播业务的优化

画质 Quality



码率 Rate



复杂度 Complexity • 画质最好的编码器?

• 速度最快的编码器?

• 高实时性的编码器?

• 超低码率的编码器?

视频编码器

视频 解码 器

测试 架构

数据 分析 工具 ■ 正确编码

■高效编码

■ 与业务相结合编码

yhevc正确编码测试方法论



编码器 vs 解码器

- 熵编码层正确性
- 重建层正确性
- 预测层正确性

编码器 vs 编码器

- C vs ASM
- Debug vs release
- x86 vs Linux
- 单线程 vs 多线程
- First time vs second time

yhevc正确编码测试举例: CUSize, TUSize

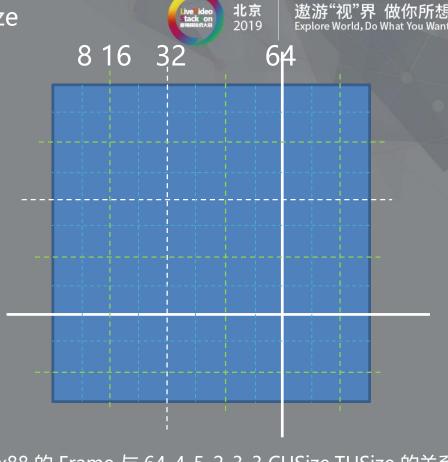
设定	MaxCUSize	MaxPartitionDepth (MinCUSize)	TULog2MaxSize (TUMaxSize)	TULog2MinSize (TUMinSize)	TUMaxDepthInter	TUMaxDepthIntra
64_4_5_2_3_3	64	4 (8)	5 (32)	2 (4)	3	3
32_3_5_2_3_3	32	3 (8)	5 (32)	2 (4)	3	3
16_2_4_2_3_3	16	2 (8)	4 (32)	2 (4)	3	3
64_1_5_4_1_1	64	1 (64)	5 (32)	4 (16)	1	1

yhevc正确编码测试举例: FrameSize

遨游"视"界 做你所想 Explore World, Do What You Want

width	height		
8	8		
16	16		
16	24		
24	16		
24	32		
32	24		
32	32		
352	8		
8	352		

width	height
64	64
72	72
80	80
88	88
88	128
96	96
104	104
112	112
120	120



88x88 的 Frame 与 64_4_5_2_3_3 CUSize, TUSize 的关系



遨游"视"界 做你所想 Explore World, Do What You Want







复杂度不变 画质提高

- 预分析算法
- 码控算法
- •基于内容改变拉格朗日常数表格

复杂度提高 画质提高

- Weighted motion estimation
- Rate distortion optimized quantization (rdoq)
- Large range motion estimation

复杂度降低 画质降低

- 快速CU分割策略
- 快速PU分割策略
- 快速TU分割策略
- 快速运动搜索
- 提前终止策略

复杂度降低 画质不变

- C语言函数优化
- 汇编函数优化
- 数据结构优化
- 多线程优化
- Cache命中优化

- 1. B站为何要做自有视频编码器
- 2. 怎样做视频编码器

3. 自有编码器现状

- 4. 视频编码器针对B站直播业务的优化
- 5. 视频编码器针对B站点播业务的优化



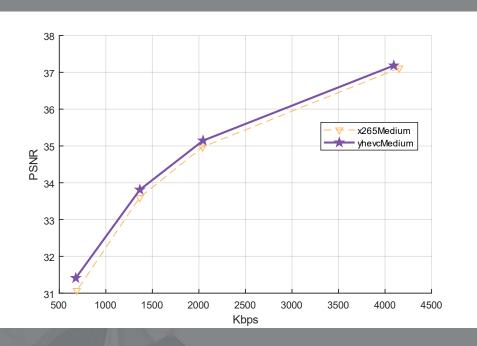
- 从零编写
- 6万行C++/C, 10万行汇编
- 40多个快速算法,80多个配置参数,400多个测试脚本
- 支持主流的编码工具集
- 多个preset档位
- 支持two-pass编码, 可用于点播及直播业务
- 与x265比, 在相同画质下能达到大约3倍的编码速度

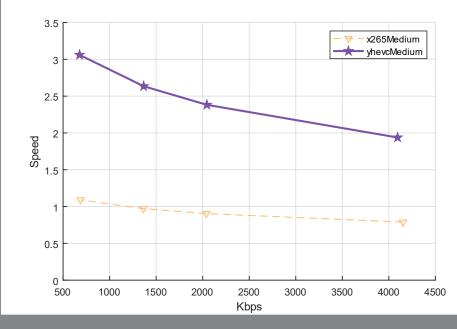


遨游"视"界 做你所想 Explore World, Do What You Want

Class B basketball_drive_1920x1080_50fps_500frames 的测试结果

- x265: tune psnr, gop 250, two pass abr, 700kbps, 1400kbps, 2100kbps, 4200kbps
- yhevc: tune psnr, gop 250, two pass abr, 700kbps, 1400kbps, 2100kbps, 4200kbps



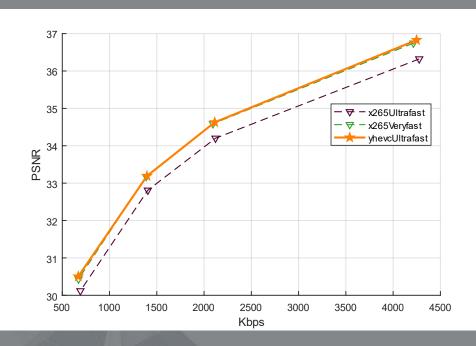


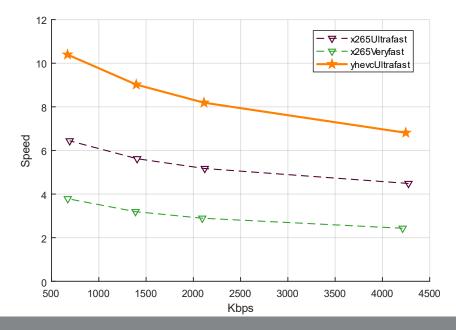


遨游"视"界 做你所想 Explore World, Do What You Want

Class B basketball_drive_1920x1080_50fps_500frames 的测试结果

- x265: tune psnr, gop 250, abr, 700kbps, 1400kbps, 2100kbps, 4200kbps
- yhevc: tune psnr, gop 250, abr, 700kbps, 1400kbps, 2100kbps, 4200kbps





yhevc编码器上线B站历程



2018.11 点播转码系统 试上线yhevc **2018.12** 直播转码系统 试上线yhevc 2019.01 大量铺开转码 为H.265格式

2019年8月 HEVC流量占比超过50%



- 2. 怎样做视频编码器
- 3. 自有编码器现状
- 4. 视频编码器针对B站直播业务的优化
- 5. 视频编码器针对B站点播业务的优化

软件视频编码器在直播中的痛点



北京 2019 遨游"视"界 做你所想 Explore World, Do What You Want





复杂场景

简单场景



编码速度

60fps

软件视频编码器在直播中的痛点



高复杂度的 足球比赛, 低 复杂度的视 频会议

分辨率 1080p, 720p

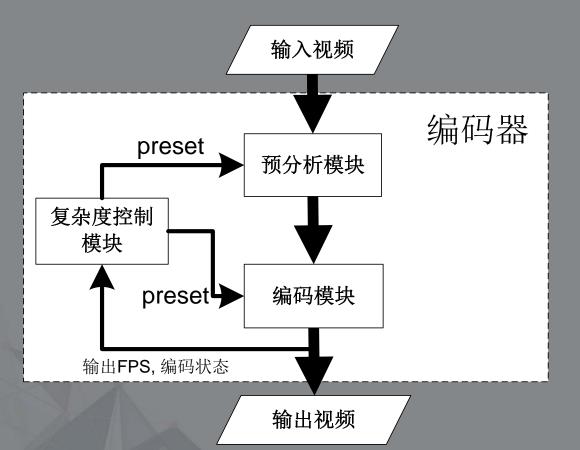
实时 编码 帧率 60fps, 24fps

码率 4mbs, 1mbs

CPU的算力 在运行中发 生变化

普通的解决方案:

- 针对某个码率, 帧率, 分区, 设定一个能够保证实时编码的编码档位
- 针对每个视频都调试一个能够保证实时编码的编码档位



设计难点:

- 编码器档位能无缝切换
- 编码器内部速度指标
- 闭环控制,鲁棒性

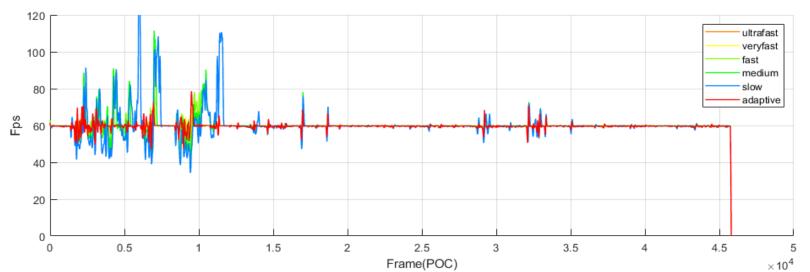
复杂度自适应的视频编码器效果

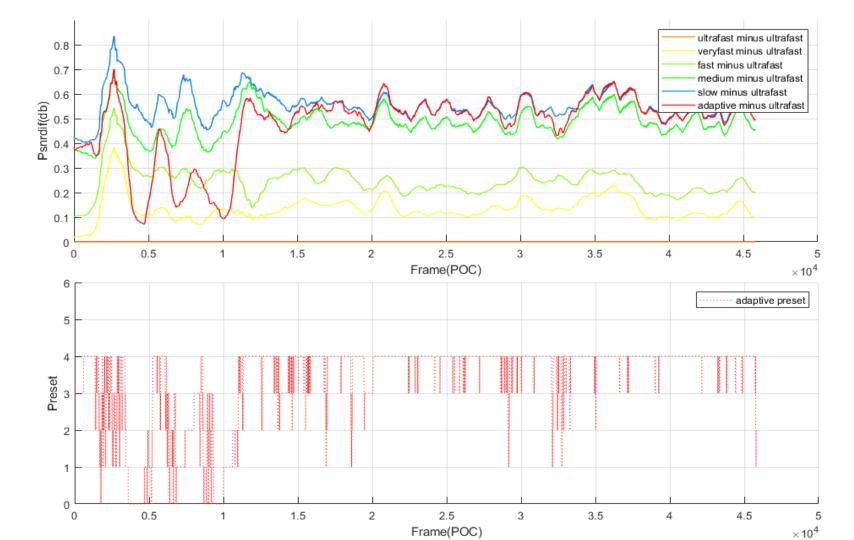




北京 2019 遨游"视"界 做你所想 Explore World, Do What You Want







复杂度自适应的视频编码器状态监测



```
{ "time" : 1563534330, "sample_num" : 10269, "70num" : 20, "50num" : 0, "avg_preset" : 3.47, "avg_psnr" : 43.74, "avg_ssim" : 0.989, "track_id" : "4148863", "tag" : "accumulative" }
{ "time" : 1563534330, "sample_num" : 10269, "fps" : 49.74, "preset" : 3.64, "psnr" : 54.22, "ssim" : 0.998, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534329, "sample_num" : 10268, "fps" : 50, "preset" : 1.72, "psnr" : 53.87, "ssim" : 0.998, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534328, "sample_num" : 10267, "fps" : 49.91, "preset" : 0.52, "psnr" : 48.04, "ssim" : 0.991, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534327, "sample_num" : 10266, "fps" : 48.62, "preset" : 1.6, "psnr" : 40.77, "ssim" : 0.982, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534326, "sample_num" : 10264, "fps" : 47.87, "preset" : 1.64, "psnr" : 40.57, "ssim" : 0.981, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534324, "sample_num" : 10265, "fps" : 51.39, "preset" : 3.66, "psnr" : 40.51, "ssim" : 0.98, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534324, "sample_num" : 10263, "fps" : 47.87, "preset" : 3.66, "psnr" : 40.79, "ssim" : 0.985, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534322, "sample_num" : 10263, "fps" : 47.87, "preset" : 3.66, "psnr" : 40.79, "ssim" : 0.983, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534322, "sample_num" : 10263, "fps" : 47.87, "preset" : 3.66, "psnr" : 40.79, "ssim" : 0.983, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534322, "sample_num" : 10260, "fps" : 49.71, "preset" : 4, "psnr" : 49.71, "ssim" : 0.994, "track_id" : "4148863", "tag" : "realtime" }
{ "time" : 1563534321, "sample_num" : 10260, "fps" : 49.72, "preset" : 4, "psnr" : 51.01, "ssim" : 0.997, "track_id" : "4148863", "tag" : "realtime" }
```

对每路直播转码流可以监控到的信息包括:

- 平均preset
- 实际帧率低于目标帧率70%的次数
- 实际帧率低于目标帧率50%的次数
- 平均psnr
- 平均ssim

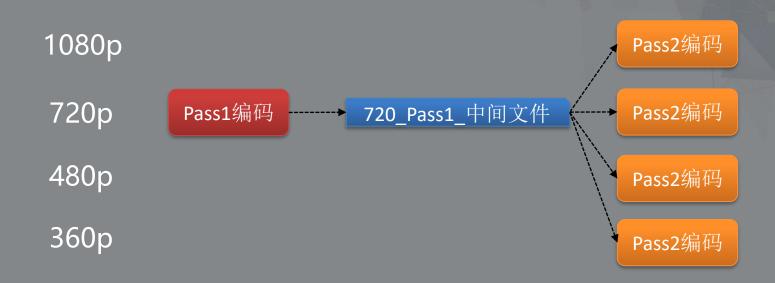
- 1. B站为何要做自有视频编码器
- 2. 怎样做视频编码器
- 3. 自有编码器现状
- 4. 视频编码器针对B站直播业务的优化
- 5. 视频编码器针对B站点播业务的优化





多分辨率转码流程





共享1pass算法的多分辨率转码流程



设计思路

- 1pass的帧信息根据分辨率缩放
- 1pass的cutree信息做二维滤波

效果

■ 编码质量不变, pipeline上的总体编码复杂度降低36%

- Machine learning with video encoder
- Content based video encoder
- AV1, VVC

Thank you



