媒体云转码的演进：MapReduce、DASH 与稳定婚姻

Alan Zhuang

cheedoong@acm.org

2014 年 3 月 10 日

Tencent 腾讯

Outline

1 背景

2 从 Cloud Transcoder 到 TranscX
- 前腾讯研究院 Cloud Transcoder
- 架平流媒体 TranscX

3 DASH 与稳定婚姻
- DASH
- 稳定匹配

4 Future Vision
- 和一些新技术的结合
- Acknowledgment

Outline

1 背景

2 从 Cloud Transcoder 到 TranscX
- 前腾讯研究院 Cloud Transcoder
- 架平流媒体 TranscX

3 DASH 与稳定婚姻
- DASH
- 稳定匹配

4 Future Vision
- 和一些新技术的结合
- Acknowledgment

多屏时代的挑战

- 多种平台

多屏时代的挑战

- 多种平台

多屏时代的挑战

- 多种平台

多屏时代的挑战

- 多种平台

多屏时代的挑战

- 多种平台

多屏时代的挑战

- 多种平台

- 多种屏幕大小

多屏时代的挑战

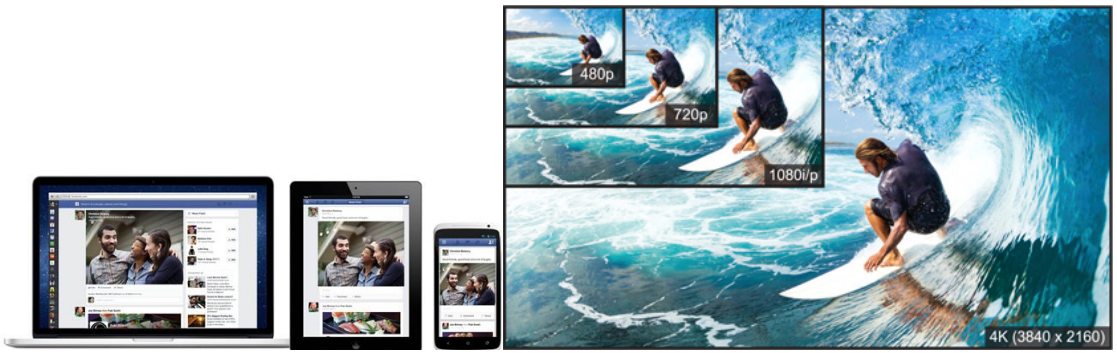
- 多种平台

- 多种屏幕大小

多屏时代的挑战

- 多种平台

- 多种屏幕大小



多屏时代的挑战

- 多种平台

- 多种屏幕大小

多屏时代的挑战

- 多种平台

- 多种屏幕大小

多屏时代的挑战

- 多种码率

多屏时代的挑战

- 多种码率

多屏时代的挑战

- 多种码率

多屏时代的挑战

- 多种码率

多屏时代的挑战

- 多种码率

多屏时代的挑战

- 多种码率

- 多种解码能力

联发科芯片	MT6572	MT6582	MT6588	MT6592
Display	960×540P	1280×720P	1920×1280P	1920×1280P
H.264 Decode	720P@30fps	1080P@30fps	1080P@30fps	1080P@30fps
HEVC Decode	N/A	N/A	720P@30fps	720P@30fps

多屏时代的挑战

- 不同封装容器支持

  mp4, mkv, avi, flv, wmv, rmvb, webm, mpeg-ts...

多屏时代的挑战

- 不同封装容器支持

  mp4, mkv, avi, flv, wmv, rmvb, webm, mpeg-ts...

多屏时代的挑战

- 不同封装容器支持

  mp4, mkv, avi, flv, wmv, rmvb, webm, mpeg-ts...

- 不同编码标准支持

多屏时代的挑战

- 不同封装容器支持

  mp4, mkv, avi, flv, wmv, rmvb, webm, mpeg-ts...

- 不同编码标准支持

多屏时代的挑战

- 不同封装容器支持

  mp4, mkv, avi, flv, wmv, rmvb, webm, mpeg-ts...

- 不同编码标准支持

  H.264(AVC), H.265(HEVC), VC-1, AVS, VP8/9, RealVideo...

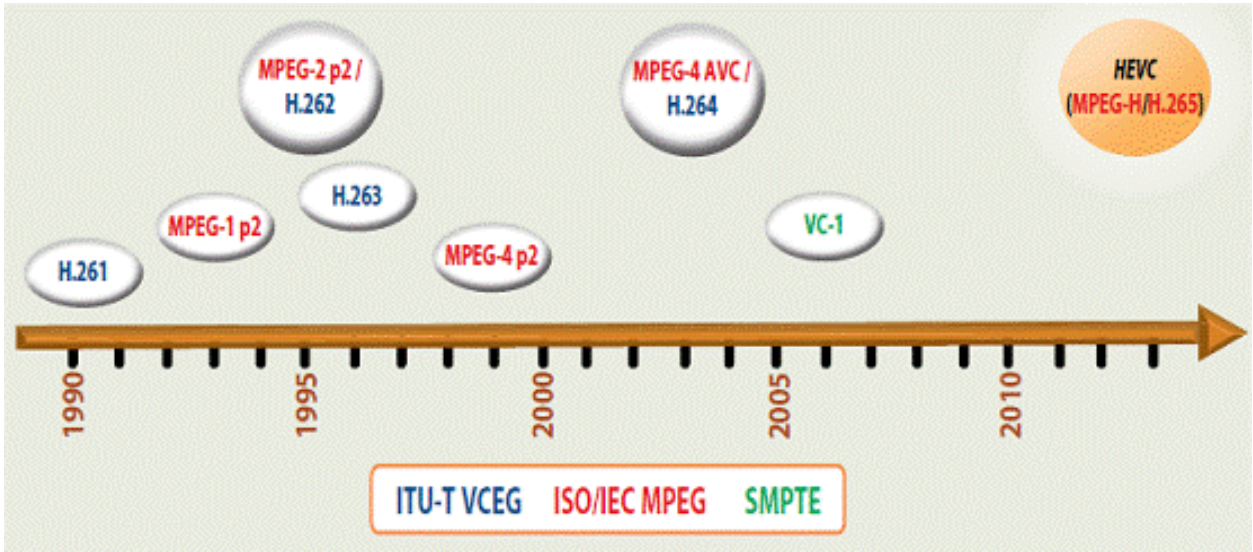
多屏时代的挑战

- 不同封装容器支持

  mp4, mkv, avi, flv, wmv, rmvb, webm, mpeg-ts...

- 不同编码标准支持

  H.264(AVC), H.265(HEVC), VC-1, AVS, VP8/9, RealVideo...



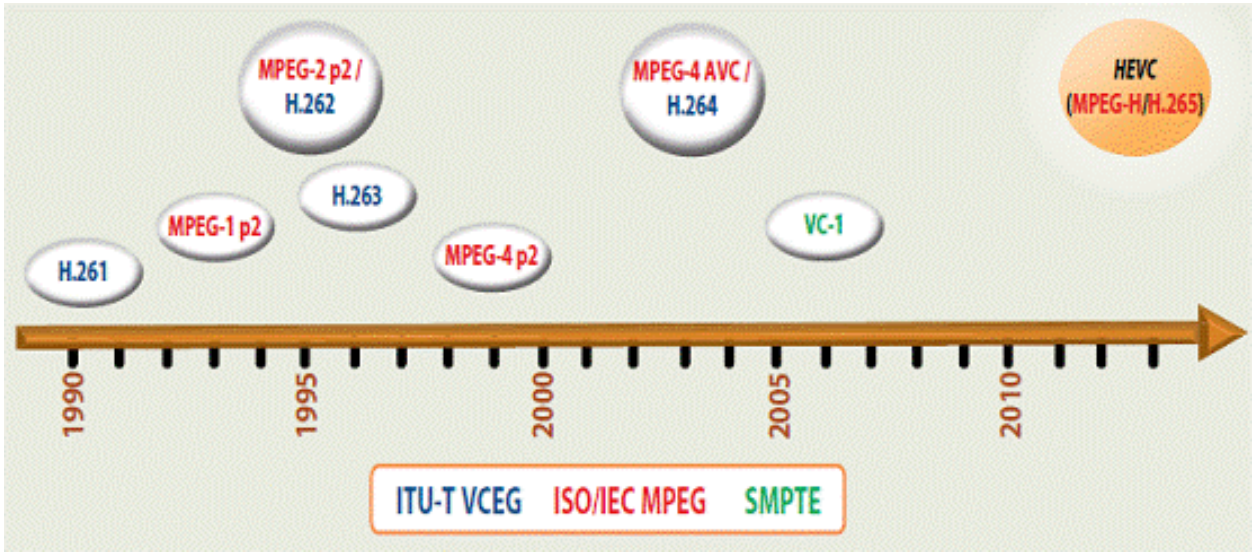
多屏时代的挑战

- 不同封装容器支持

  mp4, mkv, avi, flv, wmv, rmvb, webm, mpeg-ts...

- 不同编码标准支持

  H.264(AVC), H.265(HEVC), VC-1, AVS, VP8/9, RealVideo...



多屏时代的挑战

- 巨头角力

多屏时代的挑战

- 巨头角力

多屏时代的挑战

- 巨头角力

幸运的是，

幸运的是，

决大多数设备都支持：

幸运的是，

决大多数设备都支持：

- 编码标准

幸运的是，

决大多数设备都支持：

- 编码标准

幸运的是，

决大多数设备都支持：

- 编码标准

  H.264/AVC (ISO/IEC 14496-10; ITU-T H.264; MPEG-4 Part 10).

- 封装容器

幸运的是，

决大多数设备都支持：

- 编码标准

  H.264/AVC (ISO/IEC 14496-10; ITU-T H.264; MPEG-4 Part 10).

- 封装容器

幸运的是，

决大多数设备都支持：

- 编码标准

  H.264/AVC (ISO/IEC 14496-10; ITU-T H.264; MPEG-4 Part 10).

- 封装容器

  MP4 (ISO/IEC 14496-14; MPEG-4 Part 14).

幸运的是，

决大多数设备都支持：

- 编码标准

  H.264/AVC (ISO/IEC 14496-10; ITU-T H.264; MPEG-4 Part 10).

- 封装容器

  MP4 (ISO/IEC 14496-14; MPEG-4 Part 14).

所以，我们需要把用户/编辑上传的各种不同封装、不同编码的，一般码率比较高的源视频转成若干种适合不同设备的不同码率的 H.264 编码、MP4 封装的视频。

但是，

但是，

- 媒体转码是件及其消耗计算资源的工作
  尤其视频编码，对于目前常见的支持 SSE4 指令集的 x86/x64 CPU 的机器：

但是，

- 媒体转码是件及其消耗计算资源的工作
  尤其视频编码，对于目前常见的支持 SSE4 指令集的 x86/x64 CPU 的机器：
  - 编码 H.264 视频需要耗费播放时长的 1/3 到 2/3

但是，

- 媒体转码是件及其消耗计算资源的工作
  尤其视频编码，对于目前常见的支持 SSE4 指令集的 x86/x64 CPU 的机器：
  - 编码 H.264 视频需要耗费播放时长的 1/3 到 2/3
  - 编码 H.265 视频需要耗费播放时长的 30+ 倍

但是，

- 媒体转码是件及其消耗计算资源的工作
  尤其视频编码，对于目前常见的支持 SSE4 指令集的 x86/x64 CPU 的机器：
  - 编码 H.264 视频需要耗费播放时长的 1/3 到 2/3
  - 编码 H.265 视频需要耗费播放时长的 30+ 倍
  - 单个 CPU 核通常最多可跑 1–2 个编码任务

但是，

- 媒体转码是件及其消耗计算资源的工作
  尤其视频编码，对于目前常见的支持 SSE4 指令集的 x86/x64 CPU 的机器：
  - 编码 H.264 视频需要耗费播放时长的 1/3 到 2/3
  - 编码 H.265 视频需要耗费播放时长的 30+ 倍
  - 单个 CPU 核通常最多可跑 1–2 个编码任务
- 媒体文件大，再加上多码率副本，极其消耗存储

但是，

- 媒体转码是件及其消耗计算资源的工作
  尤其视频编码，对于目前常见的支持 SSE4 指令集的 x86/x64 CPU 的机器：
  - 编码 H.264 视频需要耗费播放时长的 1/3 到 2/3
  - 编码 H.265 视频需要耗费播放时长的 30+ 倍
  - 单个 CPU 核通常最多可跑 1–2 个编码任务
- 媒体文件大，再加上多码率副本，极其消耗存储
- 潜在的带宽消耗

解决

解决

Criteria：

解决

Criteria：

- 单机内

解决

Criteria：

- 单机内
  - 功能划分、数据局部性
    宏块组粒度的并行

解决

Criteria：

- 单机内
  - 功能划分、数据局部性
    宏块组粒度的并行
  - 内存访问、CPU 核心/Cache 拓扑结构和转码格式
    帧级或 GOP（图像组）级并行
    对 NUMA 机器特别友好

解决

Criteria：

- 单机内
  - 功能划分、数据局部性

    宏块组粒度的并行
  - 内存访问、CPU 核心/Cache 拓扑结构和转码格式

    帧级或 GOP（图像组）级并行

    对 NUMA 机器特别友好
- 分布式转码

解决

Criteria：

- 单机内
  - 功能划分、数据局部性

    宏块组粒度的并行
  - 内存访问、CPU 核心/Cache 拓扑结构和转码格式

    帧级或 GOP（图像组）级并行

    对 NUMA 机器特别友好
- 分布式转码
  - 存储

解决

Criteria：

- 单机内
  - 功能划分、数据局部性

    宏块组粒度的并行
  - 内存访问、CPU 核心/Cache 拓扑结构和转码格式

    帧级或 GOP（图像组）级并行

    对 NUMA 机器特别友好
- 分布式转码
  - 存储
  - 路由

解决

Criteria：

- 单机内
  - 功能划分、数据局部性

    宏块组粒度的并行
  - 内存访问、CPU 核心/Cache 拓扑结构和转码格式

    帧级或 GOP（图像组）级并行

    对 NUMA 机器特别友好
- 分布式转码
  - 存储
  - 路由
  - 任务调度

Outline

1 背景

2 从 Cloud Transcoder 到 TranscX
- 前腾讯研究院 Cloud Transcoder
- 架平流媒体 TranscX

3 DASH 与稳定婚姻
- DASH
- 稳定匹配

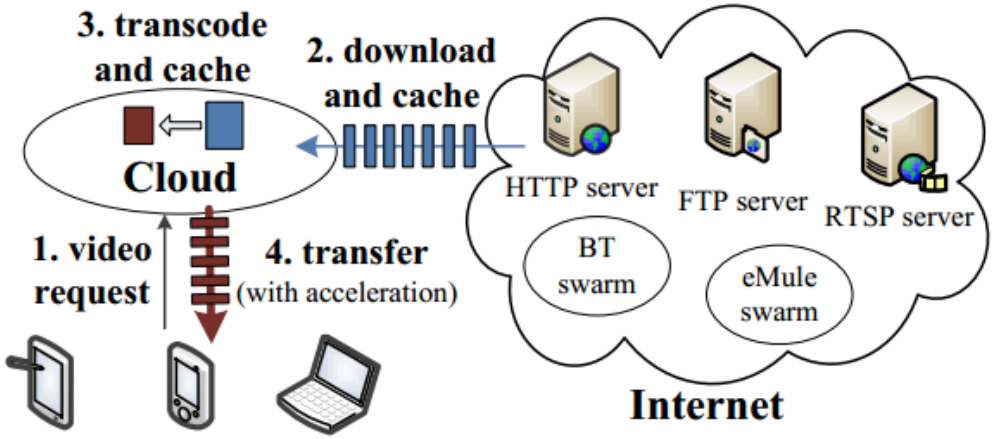
4 Future Vision
- 和一些新技术的结合
- Acknowledgment

前腾讯研究院 Cloud Transcoder

Done 2011-. Gale Huang et al. Cloud transcoder: bridging the format and resolution gap between internet videos and mobile devices. ACM NOSSDAV 2012.

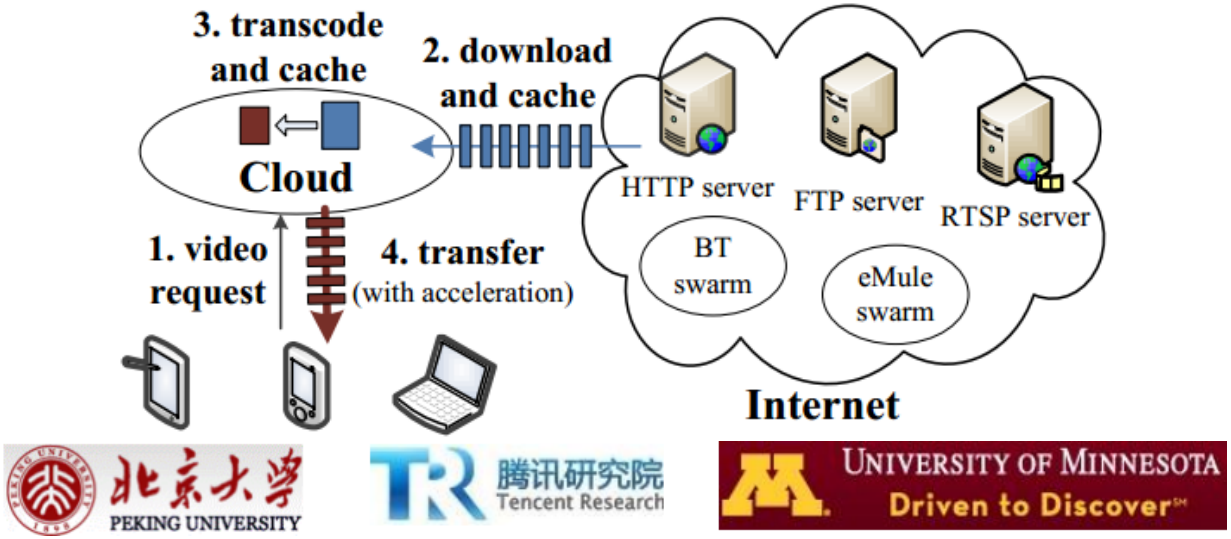
前腾讯研究院 Cloud Transcoder

Done 2011-. Gale Huang et al. Cloud transcoder: bridging the format and resolution gap between internet videos and mobile devices. ACM NOSSDAV 2012.

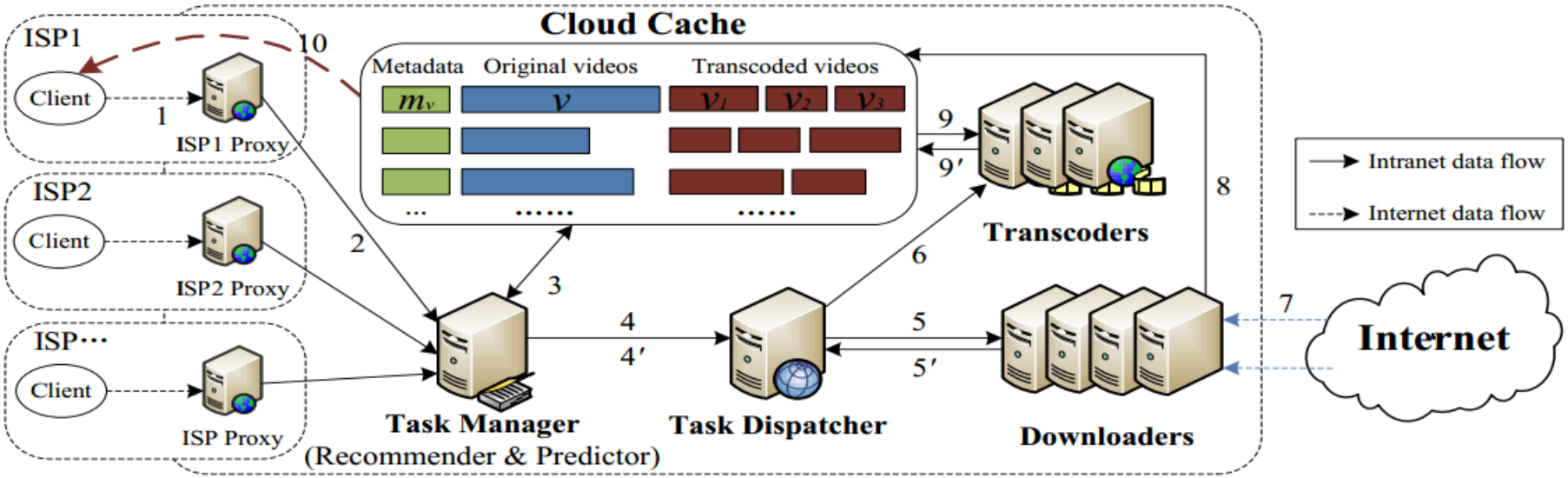


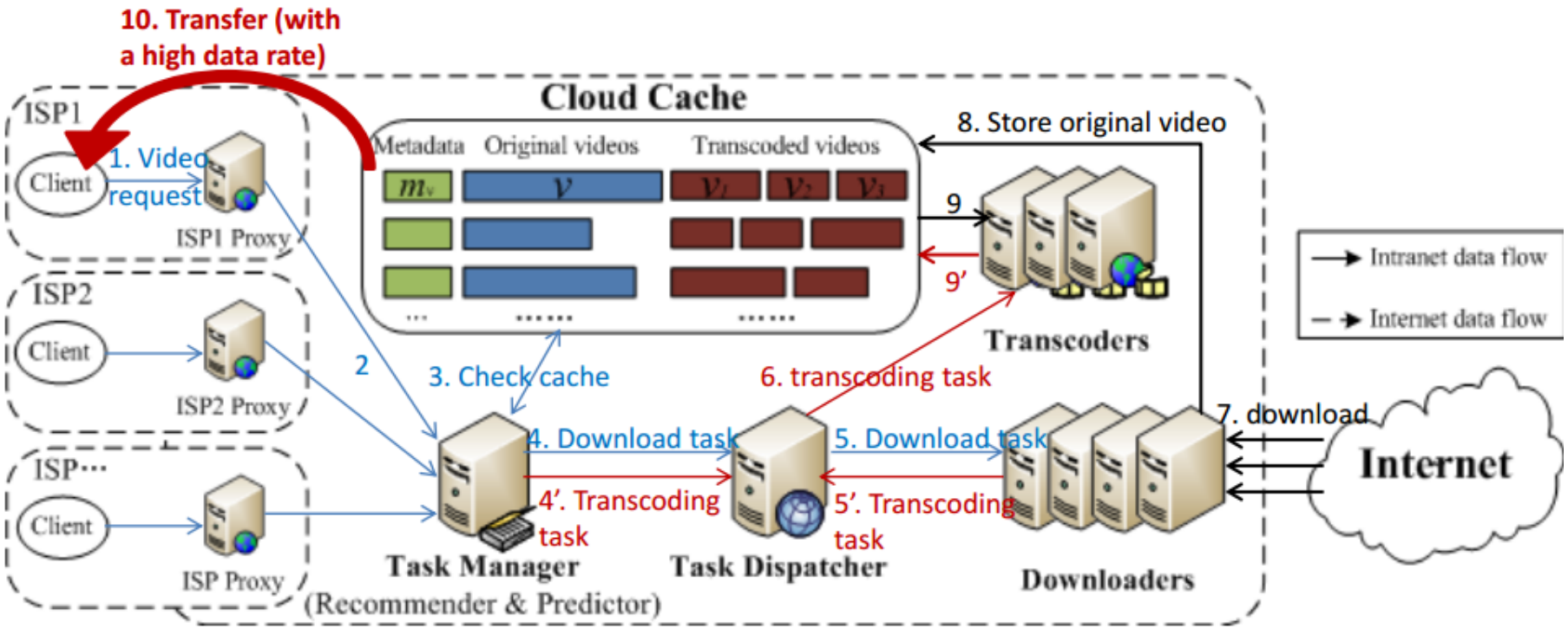
前腾讯研究院 Cloud Transcoder

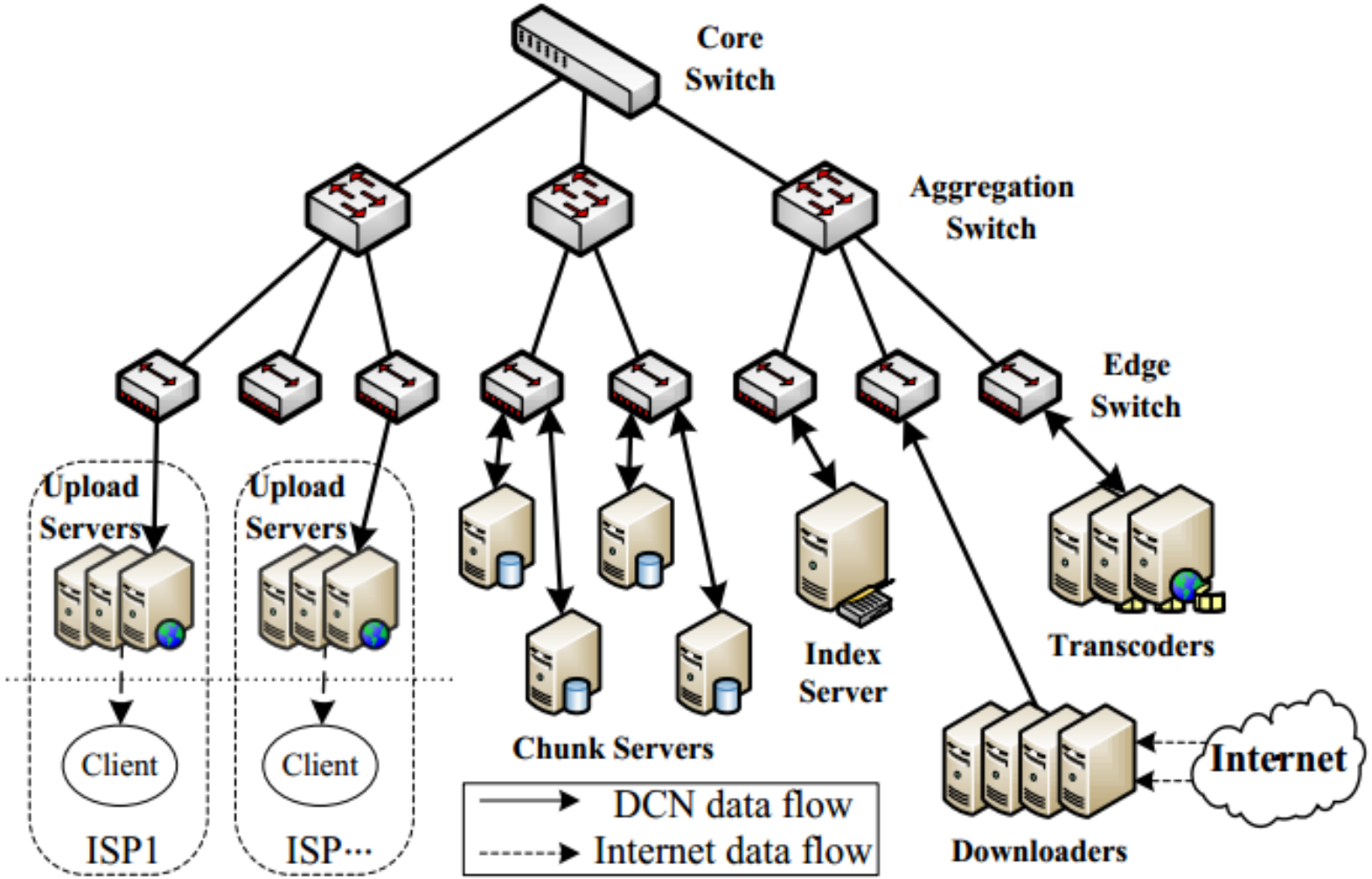
Done 2011-. Gale Huang et al. Cloud transcoder: bridging the format and resolution gap between internet videos and mobile devices. ACM NOSSDAV 2012.



Tencent 腾讯


ISP1
Client
ISP1 Proxy
ISP2
Client
ISP2 Proxy
ISP···
Client
ISP Proxy
Cloud Cache
Metadata
Original videos
Transcoded videos
Transcoders
Task Manager
(Recommender & Predictor)
Task Dispatcher
Downloaders
Internet
Intranet data flow
Internet data flow
1
2
3
4
4′
5
5′
6
7
8
9
9′
10

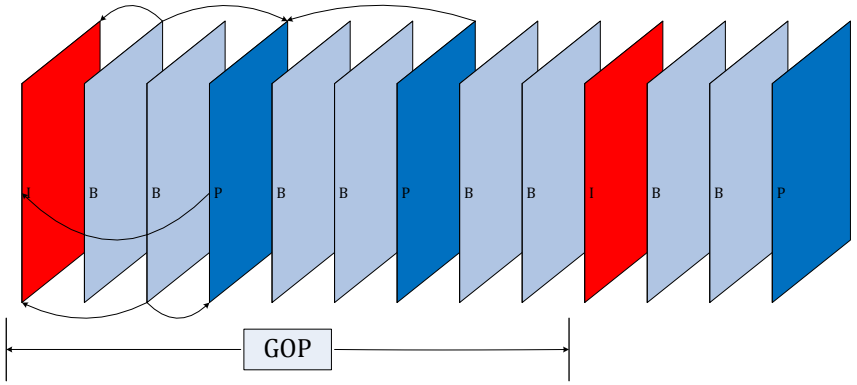

10. Transfer (with a high data rate)
Cloud Cache
ISP1
Client
1. Video request
ISP1 Proxy
ISP2
Client
ISP2 Proxy
ISP…
Client
ISP Proxy
Metadata
Original videos
Transcoded videos
8. Store original video
9
9'
Transcoders
2
3. Check cache
6. transcoding task
4. Download task
4'. Transcoding task
5. Download task
5'. Transcoding task
7. download
Internet
Task Manager
Task Dispatcher
Downloaders
(Recommender & Predictor)
Intranet data flow
Internet data flow


Core Switch
Aggregation Switch
Edge Switch
Upload Servers
Upload Servers
Client
Client
ISP1
ISP···
Chunk Servers
Index Server
Transcoders
Downloaders
Internet
DCN data flow
Internet data flow

架平流媒体 TranscX

TranscX:

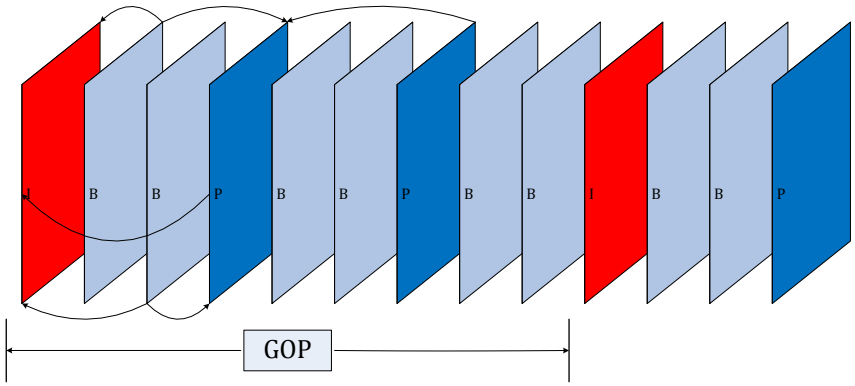
- Transcoding eXpress/eXperience/eXtreme...; transc(x)



架平流媒体 TranscX

TranscX:

- Transcoding eXpress/eXperience/eXtreme...; transc(x)



- GOP-level parallelism without REAL splitting

  Media head parsing,

  $< begin\_time, end\_time > orGOP\_id \rightarrow < start_offset, size >$

- Job/Map/Thread, accurate control and CPU & I/O limitation

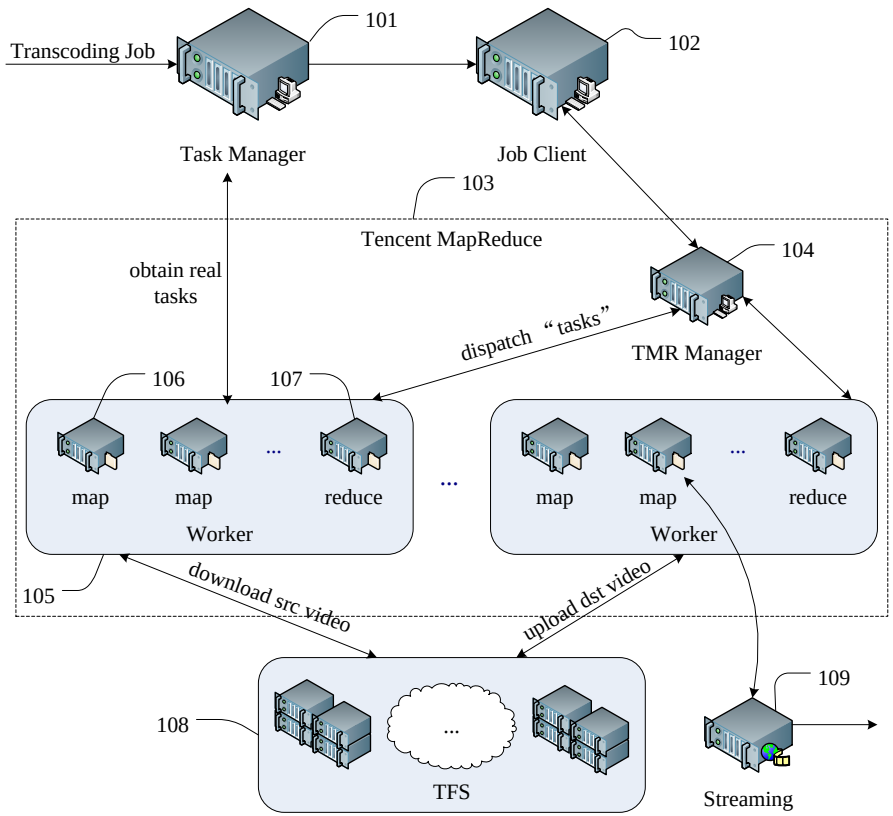
- Job/Map/Thread, accurate control and CPU & I/O limitation
- Real-time support for DASH and Live Broadcasting

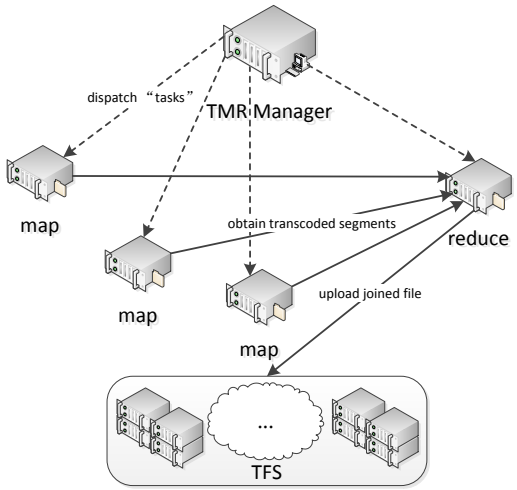
- Job/Map/Thread, accurate control and CPU & I/O limitation
- Real-time support for DASH and Live Broadcasting
- Migration from Computing to Storage: soul of cloud tech

- Job/Map/Thread, accurate control and CPU & I/O limitation
- Real-time support for DASH and Live Broadcasting
- Migration from Computing to Storage: soul of cloud tech
- It later supported WeChat, Qzone and Weishi

- Job/Map/Thread, accurate control and CPU & I/O limitation
- Real-time support for DASH and Live Broadcasting
- Migration from Computing to Storage: soul of cloud tech
- It later supported WeChat, Qzone and Weishi
- 用 MapReduce 框架统一了起来，减少了调度的复杂性，增强了可扩展性

- Job/Map/Thread, accurate control and CPU & I/O limitation
- Real-time support for DASH and Live Broadcasting
- Migration from Computing to Storage: soul of cloud tech
- It later supported WeChat, Qzone and Weishi
- 用 MapReduce 框架统一了起来，减少了调度的复杂性，增强了可扩展性
- 复用了存储服务器的空间计算资源，节省了成本


Transcoding Job
101
102
Task Manager
Job Client
103
Tencent MapReduce
obtain real tasks
104
dispatch "tasks"
TMR Manager
106
107
map
map
reduce
Worker
map
map
reduce
Worker
105
download src video
upload dst video
108
TFS
109
Streaming


dispatch "tasks"
TMR Manager
map
obtain transcoded segments
reduce
map
upload joined file
map
TFS

Outline

1 背景

2 从 Cloud Transcoder 到 TranscX
- 前腾讯研究院 Cloud Transcoder
- 架平流媒体 TranscX

3 DASH 与稳定婚姻
- DASH
- 稳定匹配

4 Future Vision
- 和一些新技术的结合
- Acknowledgment

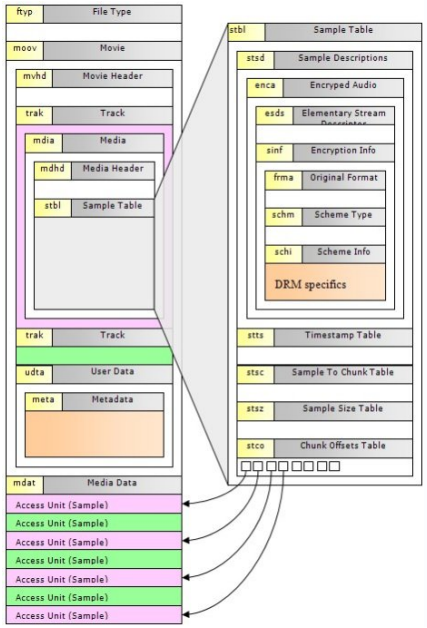
Why DASH?

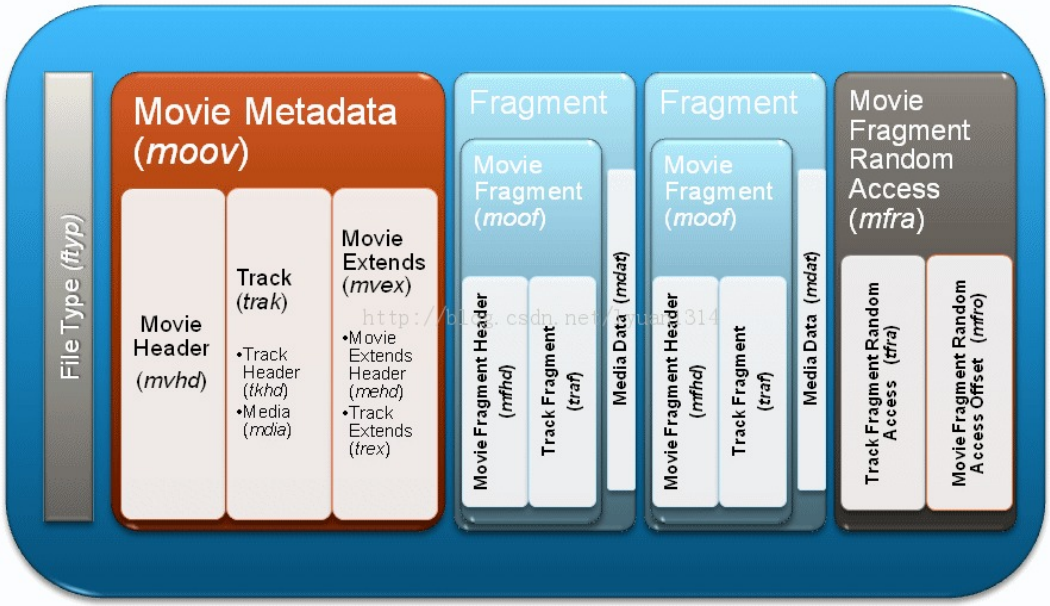
DASH: Dynamic Adaptive Streaming over HTTP.

Why DASH?

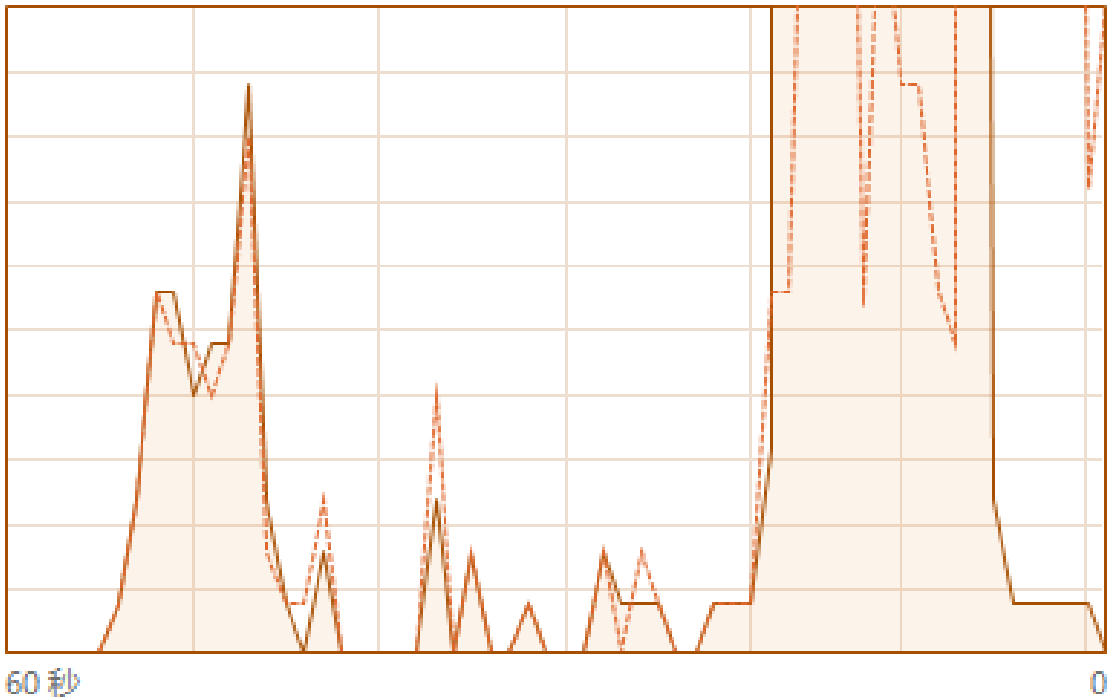
DASH: Dynamic Adaptive Streaming over HTTP.

- 庞大的文件头，导致在线播放时较大的 initial/VCR delay





- 非平滑的码率间切换 due to varying download speed



DASH

DASH: Dynamic Adaptive Streaming over HTTP.

几种 DASH 标准：

DASH

DASH: Dynamic Adaptive Streaming over HTTP.

几种 DASH 标准：

- Apple HLS (HTTP Live Streaming) 2009

DASH

DASH: Dynamic Adaptive Streaming over HTTP.

几种 DASH 标准：

- Apple HLS (HTTP Live Streaming) 2009
- Microsoft HSS (HTTP Smooth Streaming) 2010

DASH

DASH: Dynamic Adaptive Streaming over HTTP.

几种 DASH 标准：

- Apple HLS (HTTP Live Streaming) 2009
- Microsoft HSS (HTTP Smooth Streaming) 2010
- Adobe HDS (HTTP Dynamic Streaming) 2010

DASH

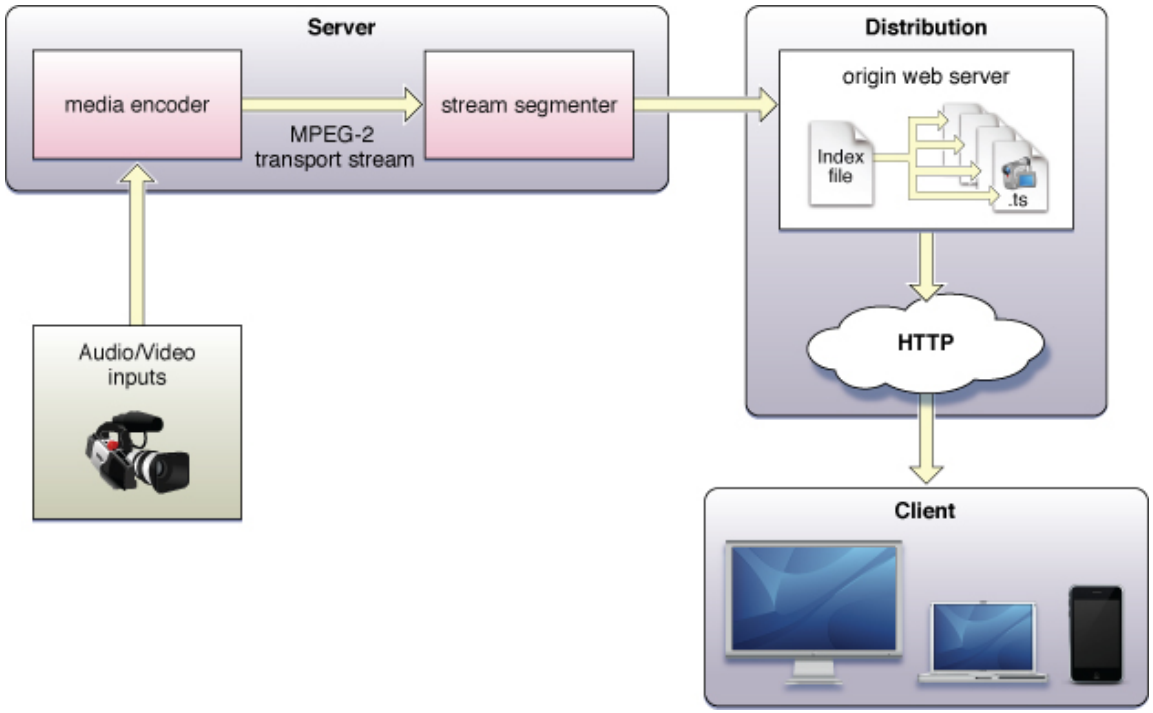
DASH: Dynamic Adaptive Streaming over HTTP.

几种 DASH 标准：

- Apple HLS (HTTP Live Streaming) 2009
- Microsoft HSS (HTTP Smooth Streaming) 2010
- Adobe HDS (HTTP Dynamic Streaming) 2010
- MPEG-DASH (ISO/IEC 23009-1) 2012

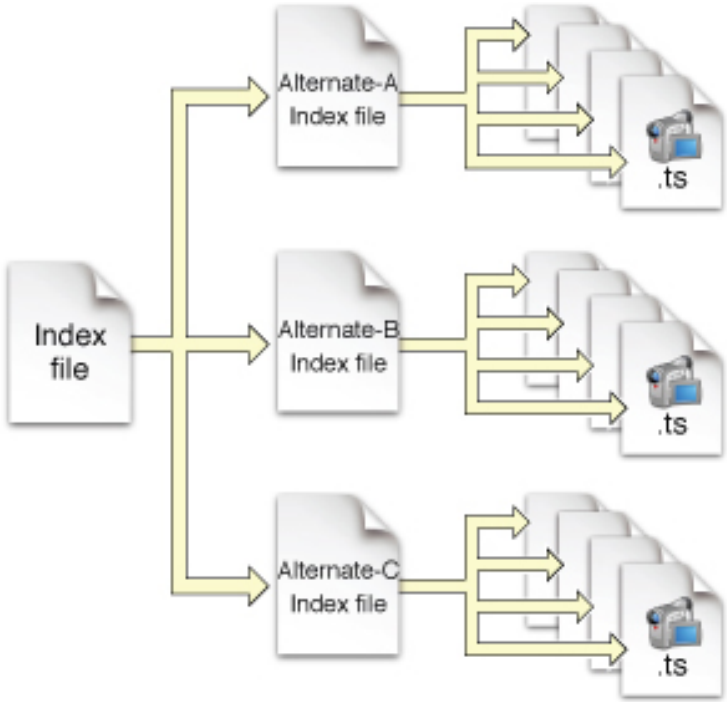
Apple HLS

- Architecture



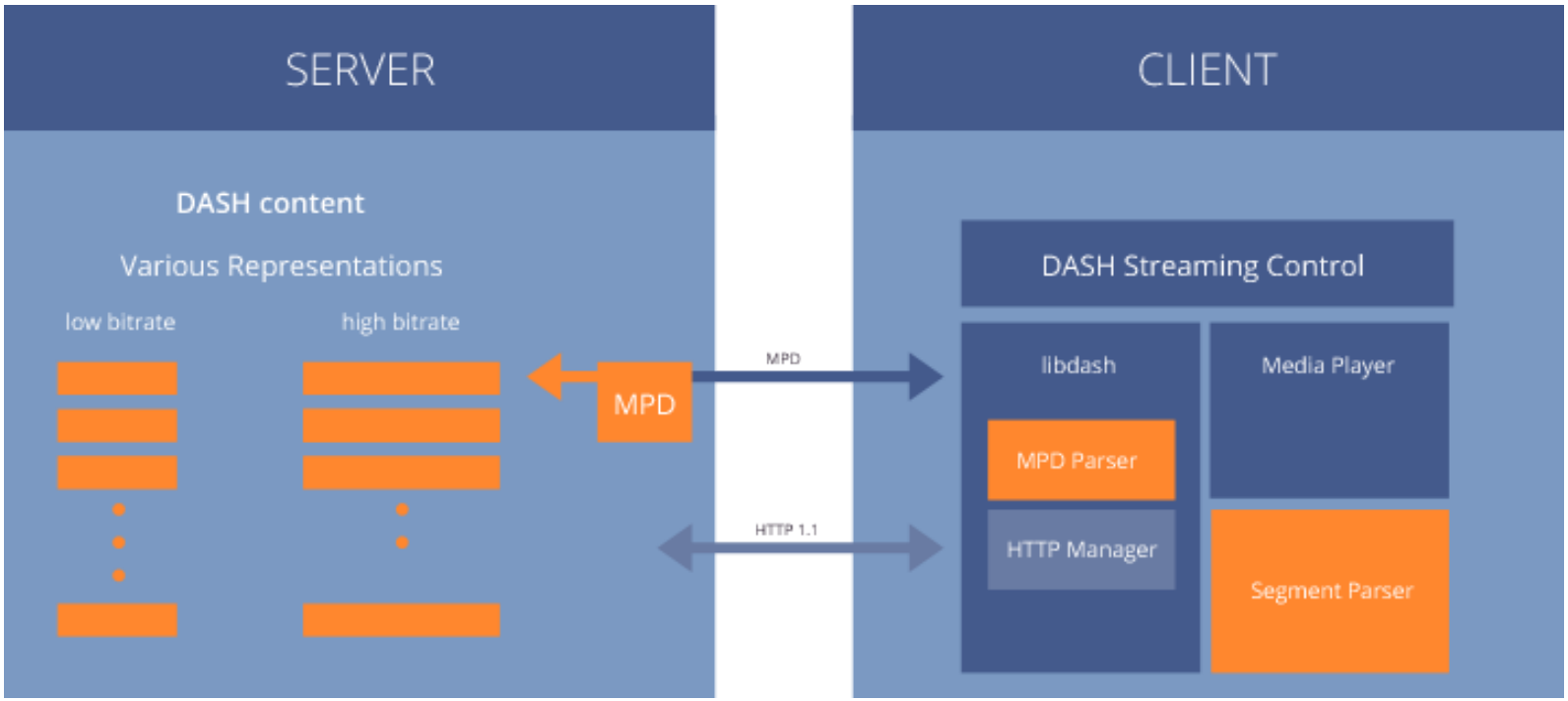
Apple HLS

- Segment Indexing



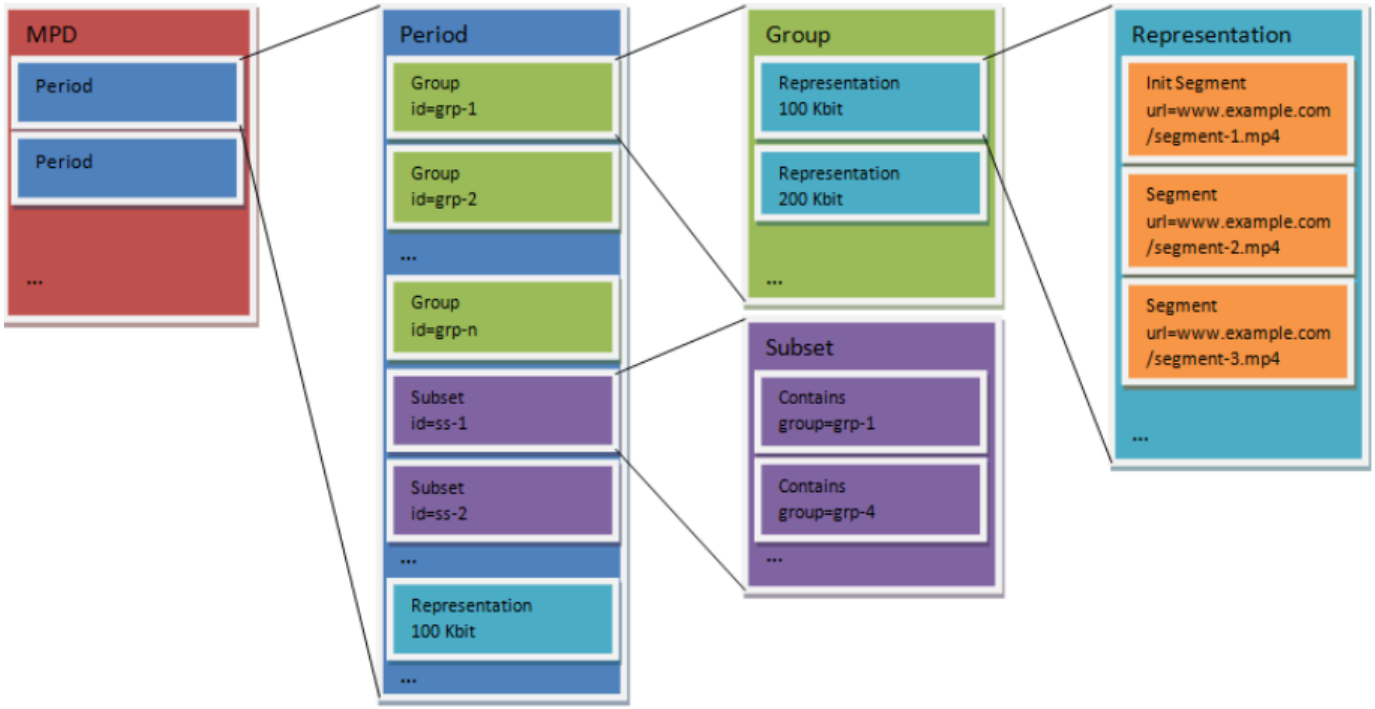
MPEG-DASH

- Architecture



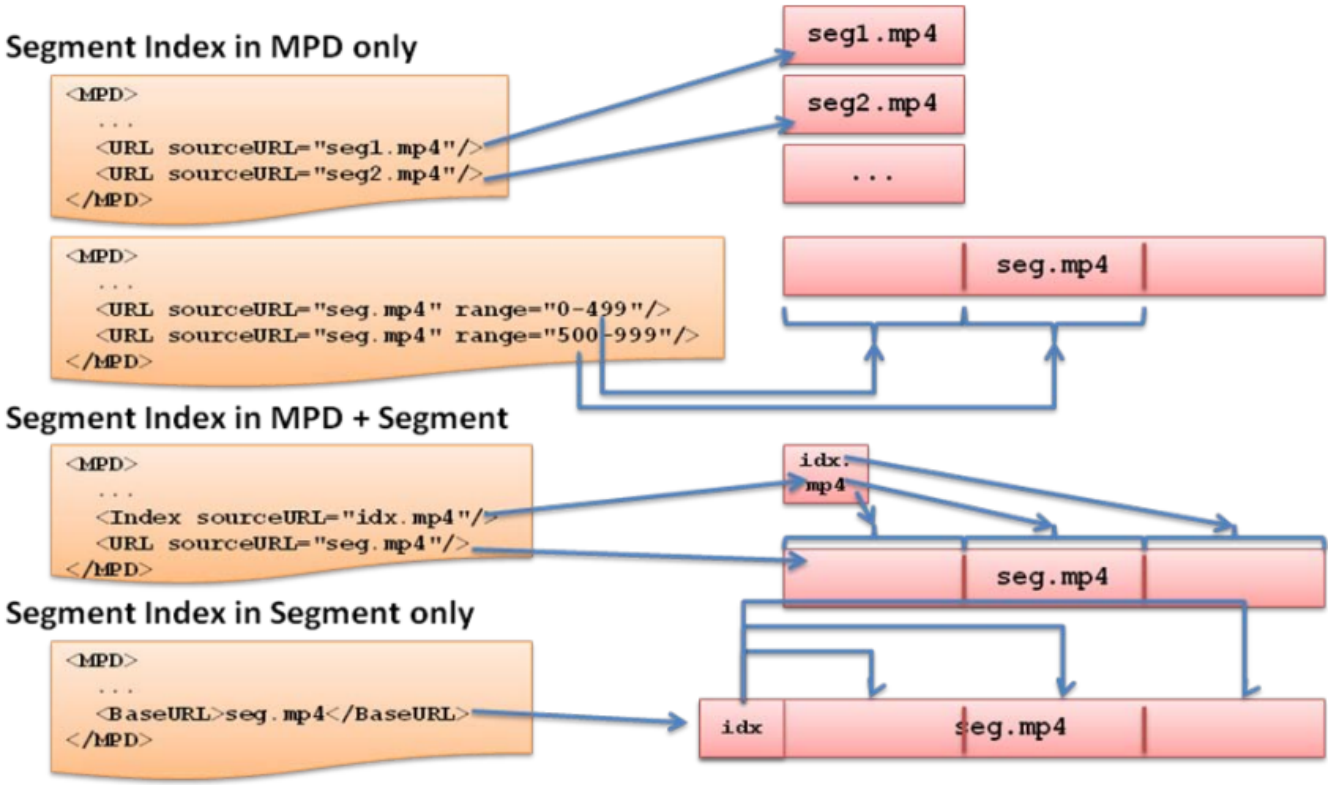
MPEG-DASH

- Data Model



MPEG-DASH

- Segment Indexing



架平流媒体对 DASH 的使用

- 尽量只转封装，不转编码

架平流媒体对 DASH 的使用

- 尽量只转封装，不转编码
- 实时、按需转封装/转码

架平流媒体对 DASH 的使用

- 尽量只转封装，不转编码
- 实时、按需转封装/转码
- 使用/自写的多种协议，解决并发 I/O 流少的问题

架平流媒体对 DASH 的使用

- 尽量只转封装，不转编码
- 实时、按需转封装/转码
- 使用/自写的多种协议，解决并发 I/O 流少的问题

  tmpfs 最简单、方便，对 Cache 管理要求不强的场合

架平流媒体对 DASH 的使用

- 尽量只转封装，不转编码
- 实时、按需转封装/转码
- 使用/自写的多种协议，解决并发 I/O 流少的问题

  tmpfs 最简单、方便，对 Cache 管理要求不强的场合

  in_mem in memory, 进程内的编解码 I/O

架平流媒体对 DASH 的使用

- 尽量只转封装，不转编码
- 实时、按需转封装/转码
- 使用/自写的多种协议，解决并发 I/O 流少的问题

  tmpfs 最简单、方便，对 Cache 管理要求不强的场合

  in_mem in memory, 进程内的编解码 I/O

  shm_mem shared memory, 多进程的编解码 I/O，可自定义 Cache 管理、淘汰算法

架平流媒体对 DASH 的使用

- 尽量只转封装，不转编码
- 实时、按需转封装/转码
- 使用/自写的多种协议，解决并发 I/O 流少的问题

  tmpfs 最简单、方便，对 Cache 管理要求不强的场合

  in_mem in memory, 进程内的编解码 I/O

  shm_mem shared memory, 多进程的编解码 I/O，可自定义 Cache 管理、淘汰算法

  AVIOContext FFMPEG built-in

架平流媒体对 DASH 的使用

- 尽量只转封装，不转编码
- 实时、按需转封装/转码
- 使用/自写的多种协议，解决并发 I/O 流少的问题

tmpfs	最简单、方便，对 Cache 管理要求不强的场合
in_mem	in memory, 进程内的编解码 I/O
shm_mem	shared memory, 多进程的编解码 I/O，可自定义 Cache 管理、淘汰算法
AVIOContext	FFMPEG built-in
async_http	适用直播，数据到达不可预期

架平流媒体对 DASH 的使用

- 尽量只转封装，不转编码
- 实时、按需转封装/转码
- 使用/自写的多种协议，解决并发 I/O 流少的问题

  tmpfs 最简单、方便，对 Cache 管理要求不强的场合

  in_mem in memory, 进程内的编解码 I/O

  shm_mem shared memory, 多进程的编解码 I/O，可自定义 Cache 管理、淘汰算法

  AVIOContext FFMPEG built-in

  async_http 适用直播，数据到达不可预期
- FPGA HEVC Encoding (proposed)

Motivation

DASH 和 Wechat, Weishi, Qzone 等社交 UGC 视频转码中存在的问题：

- 视频长度较短

  一般仅一个到若干个 GOP，MapReduce 时仅需要一个 Mapper

Motivation

DASH 和 Wechat, Weishi, Qzone 等社交 UGC 视频转码中存在的问题：

- 视频长度较短

  一般仅一个到若干个 GOP，MapReduce 时仅需要一个 Mapper
- 但 MapReduce 任务调度的延时较大

  有时甚至超过播放时长。可通过 Linux Container 守护进程的实时 MapReduce 来缓解，但非根本解决之道

Motivation

DASH 和 Wechat, Weishi, Qzone 等社交 UGC 视频转码中存在的问题：

- 视频长度较短

  一般仅一个到若干个 GOP，MapReduce 时仅需要一个 Mapper
- 但 MapReduce 任务调度的延时较大

  有时甚至超过播放时长。可通过 Linux Container 守护进程的实时 MapReduce 来缓解，但非根本解决之道
- 大部分视频的未来访问少于 X 次

  还有比较全码率转码 + 全量分发么？

Motivation

- 对数千台后台服务器连续一周运行状态的测量：

Motivation

- 对数千台后台服务器连续一周运行状态的测量：
  - 不仅 DC，OC 的多数服务器大部分时间 CPU 负载也较低甚至包括某些 HTTP 服务器

Motivation

- 对数千台后台服务器连续一周运行状态的测量：
  - 不仅 DC，OC 的多数服务器大部分时间 CPU 负载也较低
    甚至包括某些 HTTP 服务器
  - 多数服务器的 CPU 负载基本不会出现突变
    CPU 负载随时间相对平稳，但不同地区的平均负载存在差异

Motivation

- 对数千台后台服务器连续一周运行状态的测量：
  - 不仅 DC，OC 的多数服务器大部分时间 CPU 负载也较低
    甚至包括某些 HTTP 服务器
  - 多数服务器的 CPU 负载基本不会出现突变
    CPU 负载随时间相对平稳，但不同地区的平均负载存在差异
- 对应时间所有用户访问和调度情况的测量：

Motivation

- 对数千台后台服务器连续一周运行状态的测量：
  - 不仅 DC，OC 的多数服务器大部分时间 CPU 负载也较低
    甚至包括某些 HTTP 服务器
  - 多数服务器的 CPU 负载基本不会出现突变
    CPU 负载随时间相对平稳，但不同地区的平均负载存在差异
- 对应时间所有用户访问和调度情况的测量：
  - 用户对 CDN 的 region 选择存在偏好
    由于网络拓扑关系，用户去不同的 CDN 边缘节点下载的速率各不相同，而目前的调度算法，至少能得到一个次优解

Motivation

- 对数千台后台服务器连续一周运行状态的测量：
  - 不仅 DC，OC 的多数服务器大部分时间 CPU 负载也较低
    甚至包括某些 HTTP 服务器
  - 多数服务器的 CPU 负载基本不会出现突变
    CPU 负载随时间相对平稳，但不同地区的平均负载存在差异
- 对应时间所有用户访问和调度情况的测量：
  - 用户对 CDN 的 region 选择存在偏好
    由于网络拓扑关系，用户去不同的 CDN 边缘节点下载的速率各不相同，而目前的调度算法，至少能得到一个次优解
  - 不同不同地区的用户对码率存在偏好
    在前面次优解的前提下，不同地区得到平均服务质量有差异

优化问题

- 选择选择...

  to be written...

优化问题

- 选择选择...

  to be written...

- 稳定稳定匹配

  to be written...

算法应用

准备与预测框架

- 用户对所有 CDN 边缘节点的偏好

  根据过去的测速数据来 Rank 用户对所有 CDN 边缘节点的偏好

算法应用

准备与预测框架

- 用户对所有 CDN 边缘节点的偏好

  根据过去的测速数据来 Rank 用户对所有 CDN 边缘节点的偏好
- 每段视频未来一段时间的访问频率

  根据 CDN 到用户的带宽和过去一段时间用户的访问频次预测未来一段时间所有有可能有访问的视频的访问频次

算法应用

准备与预测框架

- 用户对所有 CDN 边缘节点的偏好

  根据过去的测速数据来 Rank 用户对所有 CDN 边缘节点的偏好
- 每段视频未来一段时间的访问频率

  根据 CDN 到用户的带宽和过去一段时间用户的访问频次预测未来一段时间所有有可能有访问的视频的访问频次
- 未来一小段时间服务器的期望空间计算资源

  根据分工作日/周末两种自回归模型预测未来一段时间每个 OC 节点服务器的期望 CPU 负载

Outline

1 背景

2 从 Cloud Transcoder 到 TranscX
- 前腾讯研究院 Cloud Transcoder
- 架平流媒体 TranscX

3 DASH 与稳定婚姻
- DASH
- 稳定匹配

4 Future Vision
- 和一些新技术的结合
- Acknowledgment

和一些新技术的结合

- WebRTC

  Client-end caching & delivery & broadcasting

和一些新技术的结合

- WebRTC

  Client-end caching & delivery & broadcasting
- HTML5 WebSocket, HTTP 2.0

  会话保持，多流传输

和一些新技术的结合

- WebRTC

  Client-end caching & delivery & broadcasting
- HTML5 WebSocket, HTTP 2.0

  会话保持，多流传输
- Social Could TV

和一些新技术的结合

- WebRTC

  Client-end caching & delivery & broadcasting
- HTML5 WebSocket, HTTP 2.0

  会话保持，多流传输
- Social Could TV
- SVC, MVC, HEVC, NC

  编码、多副本多版本机制、多视角、传输

Acknowledgment

本 slides 中部分内容源自与以下人员的协作/交流：

- 架平流媒体

  Leon Ouyang, Devin Zeng, Guita Zhang, Kernel He, Chris Su

Acknowledgment

本 slides 中部分内容源自与以下人员的协作/交流：

- 架平流媒体

  Leon Ouyang, Devin Zeng, Guita Zhang, Kernel He, Chris Su

- Tsinghua-Tencent Joint Lab.

  Zhi Wang

Acknowledgment

本 slides 中部分内容源自与以下人员的协作/交流：

- 架平流媒体

  Leon Ouyang, Devin Zeng, Guita Zhang, Kernel He, Chris Su

- Tsinghua-Tencent Joint Lab.

  Zhi Wang

- SNG 社交平台部

  Stone Huang

Acknowledgment

本 slides 中部分内容源自与以下人员的协作/交流：

- 架平流媒体

  Leon Ouyang, Devin Zeng, Guita Zhang, Kernel He, Chris Su

- Tsinghua-Tencent Joint Lab.

  Zhi Wang

- SNG 社交平台部

  Stone Huang

- NTU

  Young Wen