大家下午好！

我来讲讲媒体云转码的演进，同时也会牵扯出来怎么样追到更适合自己的男神女神、怎么样使婚姻更稳定的问题。

先自我介绍一下，我叫Alan Zhuang，许多同学也用俄语西语的发音阿兰来称呼我。先前在TEG架构平台部做关于流媒体平台的工作，现在SNG社交平台部从事WebRTC（浏览器上的实时通信）、PeerAcc（构建在浏览器上的内容分发和数据接力）和社交媒体有关的预研工作。

首先做一个互动，在座的各位谁拥有下面至少一种设备，请举一下手？…

恭喜大家拥有一款下个时代仍不过时的设备，具体细节请继续听哦。

我们现在正处于这样一个数字时代：我们有多种不同的平台；多种不同的屏幕大小（Apple公司的三剑客、从480P到4K视频的演示（4K就是横向分辨率超过4000）、索尼最新的PSP4）；腾讯视频、优酷、搜狐视频、百度爱奇艺大部分视频都支4种以上不同的码率，爱奇艺甚至支持到6种；我们各个终端设备芯片的硬解码能力也各不相同，比如去年主流的MT6572就只支持720P的H.264视频，而刚发布不久的红米Note就支持1080P的H.264视频，甚至720P的H.265视频，刚才提到的那些手机型号，大多芯片比这些还要好一些，都支持H.265视频的播放；我们还面临着设备对不同封装、不同编码支持的考验；这背后还存在各个巨头为了各自利益的角力，比如华为思科爱立信等公司各自围绕着HEVC(H.265)投入研发力量，并通过MPEGLA组织从专利池中攫取利益，Google则力推自己的VP9编码和WebM封装容器，为Youtube节省成本，而苹果则在H.264迁移到H.265的过程中摇摆不定，不过我们预测下一代的iPhone 6将会支持H.265。

幸运的是，因为被ISO组织标准化，绝大多数设备都支持H.264编码和MP4封装容器。所以，我们需要把各种来源，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。当然更高的码率一般也意味着更高的清晰度或更好的视觉体验。

但是，…

以往，人们从单机内的并行，和多机器之间的分布式处理两方面来解决这个问题…

2012年，前腾讯研究院的Gale Huang等人，和北大、明尼苏达的研究人员，联合在ACM的NOSSDAV会议上发表了一篇论文，描述了一个他们在2011年前完成的云转码系统。这样一批下载机器，从Internet上，通过HTTP，BitTorrent等协议下载到所需要的视频，Task Dispatcher负责任务分配、Task Manager负责管理转码前后的媒体文件，并决定何时向何地分发转码后的资源。这是它里面的数据流动，这是它的网络拓扑结构。腾讯研究院的Cloud Transcoder从整体上来看确实是个优秀的系统。但是，它还存在一些问题,,,

于是，架平流媒体团队在2012年开发了一套复用现有存储服务器的云转码系统TranscX，名字有这些意思，同时也有超越函数的意思—把复杂的事情变简单。它汲取了前腾讯研究院Cloud Transcoder的设计优点，同时，也解决了几个问题：比如不需要真正对数据进行切割，完美实现GOP级的并行。也就是，通过解析媒体文件，获得大致的时间点和文件偏移的映射。然后各个mapper通过分布式文件系统和网络协议自己取出各自要转码的数据段---GOP。那么什么是GOP呢？

假设我们要拍摄一部由一位美女老师和一位帅哥老师出演的影片，这位美女老师大家应该熟悉吧？在视频编码中，通常有帧内和帧间两种压缩编码模式，像这样前后变化不大的帧，就可以参考其前后的帧，捕捉、预测和补偿运动来减少信息冗余，但现在，忽然这位帅哥老师进来了，这个突然出现的场景中几乎没有任何前面的大块信息可以参考，这样这个帧就只能采用帧内编码，但它后面的帧却又可以参考这一帧，但如果刚才那位美女不出现，它们就不可能参考前面的帧。我们把刚才突然开始出现美女或帅哥的这种帧叫即时刷新关键帧，从一个这种帧到下一个这种帧叫做一个闭GOP。显然，每个闭GOP都是可以独立解码的。…

我们都知道，MapReduce通常是用来做搜索引擎的离线索引计算，以及分布式分析的。媒体转码正好是数据密集型的，可以用分而治之的策略实现数据并行。我们用MapReduce框架统一起来，减少了调度复杂性，增强了可扩展性，通过TFS/CFS和网络协议，处理海量海量媒体资源…

刚才提到了DASH，DASH是最近两三年在在线流媒体领域非常火的一个词。那么什么是DASH，为什么要有它呢？首先，视频文件通常比较大，结构比较复杂，用户也不总是线性观看，还会有拖拽等VCR行为，这样如果视频长的话，就会造成比较大的初始和VCR时延。即使进行简单切片，也不能完全解决这个问题。其次， 我们的网络环境是不稳定的，视频通常是以比较恒定的码率来编码的，这样在我们网速比较快时，可能得不到更高清晰度的体验，而在网速不给力时，又可能会遇到缓冲等待。于是，DASH应运而生，HTTP协议之上的动态自适应流媒体。最早提出的是Apple，我们的iPhone, iPad, Safari浏览器都支持，后来微软、Adobe也提出了各自的DASH标准，2012年，MPEG综合了几者，制定了MPEG-DASH标准，并且被ISO组织认定为国际标准。但以上的几种标准都是互不兼容的。Apple的HLS比较简单，基本思想就是讲视频各个码率的副本切片，在服务端维护一个包含各个码率描述、地址的播放列表；而客户端则会根据用户过去一段时间的下载速度和当前码率来动态地调整下面要播放的码率副本。这是它的索引方式。而MPEG-DASH就要复杂一些，架构… ，数据模型是这样，索引方式又有这么多种。…

…LXC, Docker… 而且，这样数据常驻内存的实时MapReduce系统，更适合机器学习、图计算这样迭代式的数据流和计算模型，而在我们的转码应用中，即使考虑多码率副本，因为毕竟编码的代价比解码高出两个数量级，所以常驻内存也没有太大的必要了…

看到这些我们想到了什么？时间序列分析？自回归？支持向量机？神经网络？短时间内的线性回归？对！综合这些我们就可以训练出模型，比较准确地预测下一个时刻的状态。…

其中有个关键的问题：每个用户到每个CDN边缘节点的连接延迟、下载速度都不一样；每台流媒体服务器以什么样码率的副本服务什么样的用户，转码还是不转码，系统付出的代价也不一样。这是个双方市场，考虑成调度问题还是匹配问题呢？这让我们想起了稳定婚姻问题。所谓稳定婚姻问题，就是给一群男人女人配对，使得对于每一个人，在他心目中比他当前伴侣更好的异性，都不会认为他也是一个更好的选择。也就是说，任何人想和他对象劈腿，那么他想找的异性都不会愿意。当然文章和姚笛事件正好就违背了这个规则…

在这个问题解决的过程中，我们意识到，最最重要的事情就是Satisfying Women，这也印证了腾讯“一切以用户价值为依归”的理念，先考虑用户，再考虑服务器负载、资源代价诸如此类。刚才那位苍老师和那位“毕老师”没有成就一段佳缘，也许就是因为“毕老师”没有这位帅哥知道怎么样Satisfying Women，而只知道Satisfying Himself。

前几天，我们看到某某云，在各大媒体上宣传造势，看势头又要来颠覆电视产业了。… 大家仔细看，这里面讲的这些问题，不正是我们之前做过的事情么？在此，我引用《腾云》杂志的近未来专刊的广告语：未来已经发生，只是尚未普及。谢谢大家！