



论 文

大规模 P2P 视频点播系统: 聚焦客户端

黄琰^①, 徐恪^{②*}, 李海涛^②, 曹杨^①, 姚欣^①

^① 上海聚力传媒技术有限公司, 上海 201204

^② 清华大学计算机系, 北京 100084

* 通信作者. E-mail: xuke@tsinghua.edu.cn

收稿日期: 2008-06-30; 接受日期: 2010-04-01

国家自然科学基金 (批准号: 60970104)、国家高技术研究发展计划重大课题 (批准号: 2008AA01A326)、国家重点基础研究发展计划课题 (批准号: 2009CB320501) 和教育部新世纪优秀人才支持计划资助项目

摘要 使用 P2P 网络提供视频点播 (VOD) 服务可以支持大规模用户群以及节省服务器带宽开销. 由于 VOD 必须支持 VCR 特性 (如拖动、暂停、任意选择观看位置等), 构建出大规模可扩展的高性能 VOD 系统仍然存在很大的挑战. 论文给出了基于 P2P 的 VOD 系统参考模型. 定义了设计目标, 分析了其中的难点. 以 PPLive 点播系统为实例给出了基于客户端的解决方案, 包括磁盘缓存淘汰策略、内存缓存预取策略、邻居关系维护、任务调度、数据上传策略等. PPLive 点播系统目前以投入实际运行, 最大同时在线人数超过 21 万, 在线影片数据超过 700 部, 流畅度与拖动延迟良好, 与传统的客户服务器模式相比, 平均服务器带宽节省率达到 88% 以上, 人数峰值带宽节省率达到 93%.

关键词 P2P VOD 带宽节省率 流畅度 拖动延迟

1 引言

VOD (video on demand) 即视频点播技术的简称, 也称为交互式电视点播系统. 视频点播是计算机技术、网络技术、多媒体技术发展的产物, 基于互联网实现支持大规模用户的视频点播系统一直是互联网新型业务追求的目标. 目前已经出现了许多支持视频直播的系统, 比如 Coolstreaming^[1], PPLive^[2], PPStream^[3], GridCast^[4], 还有 YouTube^[5], YouKu^[6] 等提供较低清晰度的短视频的网站. 传统的视频点播服务主要基于服务器/客户端的模式运营. 媒体数据如视频, 主要集中在服务器端, 用户通过网页的视频索引与相应服务器建立连接并下载视频数据, 达到在线观看的目的. 这种模式一定程度上实现了视频点播服务, 然而这种模式下服务器所能承载的用户很有限. 研究表明 P2P (peer-to-peer) 技术可以应用到 VOD 系统来达到节省服务器带宽, 增加系统可扩展性的目的^[7-9]. 但是真正实现较高清晰度、高流畅度和低拖动延迟的点播系统还有很大的困难. 首先, VOD 系统动态性很高, 除了随机进入、离开之外, 还可以任意拖动、暂停, 使得 P2P 拓扑结构变化剧烈. 其次, 冷门热门频道和影片并存, 随着频道数目的增多, 冷门频道比例会更大, 用户分散在频道的各个位置, 造成很难相互共享数据. 最后, 每个用户上传/下载带宽有很大不同, 而且由于网络动态性特点, 实际带宽能力不断变化, 邻居选择算法必须能够动态选择高带宽的邻居来下载数据.

PPLive 推出的 P2PVOD 系统^[10] 截止 2008 年 6 月最大同时在线人数超过 21 万, 频道数大于 700, 服务器带宽平均节省率超过 88%, 人数峰值带宽节省率高达 93% 以上. 流畅度与拖动延迟良好. 论文结合 PPLive 视频点播系统介绍了 P2PVOD 的系统模型、设计难点、客户端设计, 以及性能评价. 本文的主要贡献如下:

(1) 详细介绍了 P2PVOD 的参考模型, 包括系统部署、流媒体数据存储、分发以及客户端操作.

(2) P2P 系统中起关键作用的是客户端, 论文重点介绍了 P2PVOD 系统客户端的设计, 包括磁盘缓存淘汰算法、内存预取算法、邻居关系维护、任务调度以及数据上传策略. 这些策略被 PPLive P2PVOD 系统所采用, 表现出良好的整体性能.

(3) 运行的实际系统表明 P2P 技术可以实现大规模 VOD 系统, 相比传统的客户/服务器的 VOD 服务提供模式, 服务器带宽节省率可高达 88% 以上, 人数峰值带宽节省率高达 93% 以上, 同时流畅度与拖动延迟等用户满意度指标良好.

本文结构如下: 第 2 节相关工作介绍; 第 3 节描述 P2PVOD 系统参考模型; 第 4 节 P2PVOD 设计目标以及相应难点分析; 第 5 节为客户端策略, 主要包括数据存储、数据调度; 第 6 节介绍现有系统的运行情况, 主要包括服务器带宽节省率, 以及用户体验; 第 7 节总结全文并提出了进一步工作的设想.

2 相关工作

针对传统服务器/客户端模式 VOD 系统不能支持大规模用户的问题人们提出了多种方案, 包括 CDN(content delivery network)^[11] 模式, P2P 模式. CDN 系统通过在网络各处放置节点服务器, 构成在现有互联网基础之上的智能虚拟网络层, 能尽量避开互联网上可能影响数据传输速度和稳定性等瓶颈问题, 使内容传输的更快、更稳. 对用户来说, 通过 CDN 系统, 不但使得到响应的的时间被大大缩短, 而且连接质量也大大提高, 从而有效提高了上网访问的总体性能. 但是对 VOD 服务提供商而言仍然不得不支付高额的硬件费用, 以及巨大的流量费用.

近年来, P2P 在实时流媒体中的应用取得了很大的成功, 产生了大量的观看效果很好, 服务器带宽节省率很高的 P2P 直播软件, 比如 Coolstreaming^[1], PPLive^[2], PPstream^[3] 等. 同时大量测量分析与理论研究对 P2P 直播系统的优化设计提供了支持. 被动嗅探技术^[12] 以及主动爬行方法^[13] 被用于从客户端角度对系统进行测量. Wu 等^[14] 从服务器端角度对流行的 P2P 直播软件进行测量. Hei 等^[15] 应用排队论以及随机流对 P2P 系统进行建模, Zhou 等^[16] 利用滑动窗口技术建模来分析缓存的填充概率.

相比与 P2P 实时流媒体的应用, P2PVOD 研究还处于起步阶段. 现在对 P2PVOD 的研究, 主要集中在以下几个方面:

(1) 测量, 包括 VOD 系统用户行为特点以及 P2PVOD 系统性能. Huang 等^[7] 利用 9 个月的 MSN Video 日志数据, 分析了 VOD 系统中用户行为特点, 同时认为 P2P 技术应用到 VOD 系统中可以节省 95% 的服务器带宽; Cha 等^[17] 通过收集的大量数据, 分析了 YouTube 这种用户提供内容的 VOD 系统中视频生命周期以及用户对不同影片的请求统计特性, 同时对 P2P 技术应用到 YouTube 中可以带来的带宽节省率进行了分析.

(2) P2PVOD 共享应用层网络设计. 树形的应用层网络结构被广泛用于组织节点^[18-21]. Cui 等^[22] 提出利用扩展应用层多播, 在 Peer 上缓存数据的 Ostream 方案来支持 VOD; Annapureddy 等^[8] 用模拟的方法研究了 Mesh 网络给 P2PVOD 网络带来的灵活性.

(3) 对 VCR(video cassette recorder) 的支持. Wang 等^[23]提出了一种动态分布式结构 DSL(dynamic skip list)(DSL) 来动态支持 VCR 操作; Guo 等^[24]提出动态提高 VCR 操作数据请求优先级的方法支持 VCR; He 等^[25]提出一种混合的缓存机制以及预取机制提高 VCR 的延迟; Cheng 等^[26]提出了一种 RINDY 结构来支持 VCR.

(4) 数据存储. Cheng 等^[27]比较了磁盘缓存策略中单视频共享与多视频共享的性能区别, 认为多视频共享的方案可以带来很大的性能提高; Guo 等^[28]提出了分片技术进而依靠 BitTorrent 方法来共享分片; 理论分析认为在 P2P 网络中, Peer 按请求等比例存储数据可以达到最优化的用户性能以及网络性能, 并且认分布式 LRU(least recently used) 淘汰算法可以很接近按请求等比例存储数据的效果^[29,30].

以往研究大部分侧重局部算法对特定性能的改进, 本文从整体设计的角度, 结合了已经大规模部署的 P2P VOD 系统 — PPLive 点播系统, 给出了系统参考模型, 分析了此类系统设计的主要难点, 给出了最为重要的客户端的设计方案. 运行情况表明服务器平均带宽节省率可高达 88%, 人数峰值带宽节省率高达 93% 以上, 同时流畅度与拖动延迟指标良好.

3 P2P VOD 参考模型

在互联网上支持 VOD 服务必须能够支持大规模的用户群体. 我们认为, 在当前的网络条件下, 下述的 P2P VOD 系统设计目标是合理的并且可行的: 支持百万以上的在线用户, 支持万部以上的在线影片, 支持 400 kbps 以上的码率, 支持用户的拖动等点播操作. 而随着用户下载带宽的提升, 特别是随着 ISP(internet service provider) 提供的 ADSL(asymmetric digital subscriber line) 接入带宽超过 2 Mbps, 超过 1 Mbps 的播放码率也将成为下一步的设计目标. 从设计目标可以看出, 如何设计高效的 P2P 客户端是系统是否达到设计目标的关键点. 这里首先需要提出基于 P2P 的系统参考模型.

3.1 系统部署

由于 P2P VOD 系统必须能够提供稳定的服务, 而在用户主机上运行的客户端 Peer 程序具有很强的动态性, 完全依赖用户的 Peer 端程序是不现实的, 因此在 P2P VOD 系统中必须要引入服务器. P2P VOD 系统中服务器的设置可以在一定条件下很好的调整整个系统的性能, 为用户提供更为可靠的服务, 但是服务器的存在又有可能限制系统的可扩展性. 因此服务器如何部署, 以及承担何种功能非常重要. 综合考虑各种因素, 我们提出一种混合式的 P2P 网络结构, 建立索引服务器 Tracker, 提供最基本的资源查找功能. 同时 Peer 之间建立非结构化的网状网络, Peer 也可以通过该网络进行资源的定位和查找.

我们提出的混合式的 P2P 网络结构如图 1 所示, 主要由 Tracker, Publish Server, Image Server, Trans Server, Log Server, Peer 构成.

Tracker: 系统 P2P 索引服务器, 用来管理各区域内的节点, 存储客户端节点所拥有的数据资源信息, 负责 P2P 网络中资源 Peer 的发现和查找功能, 客户端在启动时需向 Tracker 注册加入, 并定期汇报自己所拥有的资源信息.

发布服务器 (Publish Server): 主要负责节目的发布, 向 Tracker 注册已发布的资源信息, 同时推送节目数据到镜像服务器, 建立节目的多个数据源副本, 生成节目列表.

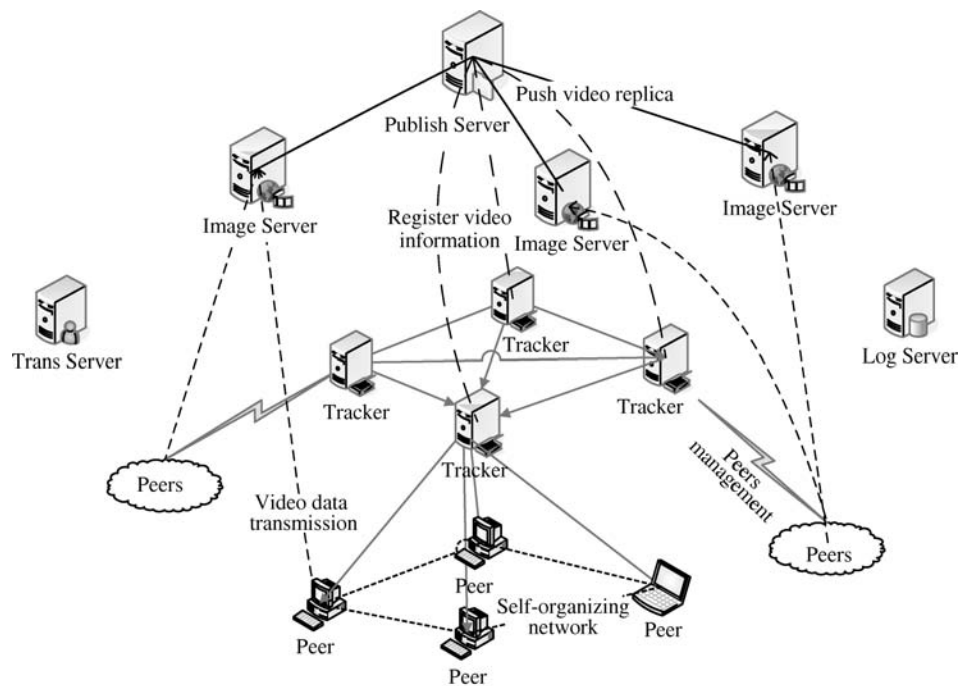


图 1 P2PVOD 系统部署图

Figure 1 P2PVOD architecture

镜像服务器 (Image Server): 存储节目源数据, 驱动整个 P2P 网络的数据扩散, 待节目数据扩散到一定程度后, 作为数据补足服务器。

穿透服务器 (Trans Server): 专门用于 UDP(user datagram protocol) 报文的 NAT(network address translation) 穿透。

日志服务器 (Log Server): 收集 Log 的服务器, 后台数据分析系统可进行用户的行为分析和统计。

PPLive 点播系统就参照该模型进行了部署, PPLive 全局拥有一个 Publish Server, 系统支持跨异构网络运行 (教育网、电信、网通等不同的 ISP) 在每个支持的网络中至少拥有一个 Tracker, 在每个异构网络中拥有自己的 ImageServer。通过让整个系统的网络进行区域化管理, 减小 ISP 骨干网的数据传输量, 区域内的结点进行自组织 (Gossip 方法), 网络也便于管理和维护, 有很好的可控性。客户端 (Peer): 直接面向用户的应用软件, 在 P2P 网络中作为一个普通节点, 可以从网络获取流媒体数据, 进行视频的重组、解码及播放, 同时对网络提供物理数据存储和服务, 可与 P2P 网络中其他节点交换资源。

关于此系统部署模型的使用应结合不同要求灵活处理。比如 PPLive 在实际部署中, 我们发现 Tracker 分区自制能够带来以上优点之外, 存在一个性能问题, 即不同区域的 Peer 不能很好的通信, 尤其当某影片拥有的在线人数很少的时候问题很突出, 此时单个区域的 Peer 很难构成有效地 P2P 网络, 由于资源缺乏不能从 Peer 获取足够快的下载数据。而如果打通各个 Tracker 间的联系, 则可以提高 P2P 网络中 Peer 的数目, 进而提高相互间的下载速率。系统设计人员参考该模型时应该权衡这两种不同策略的利弊。

如图 2 所示客户端操作从用户的角度看主要分为启动、观看节目、暂停、拖动、退出。下面各小节进行简要介绍。

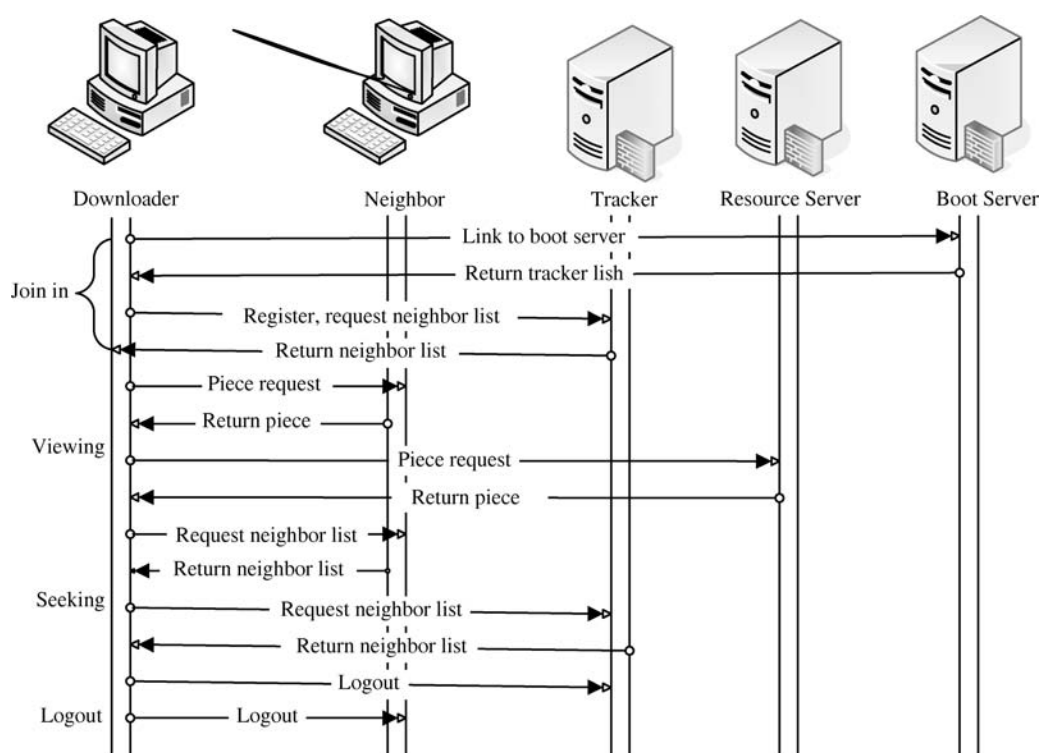


图 2 系统操作

Figure 2 Client operation

3.2 启动和退出

客户端启动后, 首先加入 P2P 网络, 先连接启动服务器 (Bootstrap Server), 获取本节点所在 ISP 网络内的 Tracker 列表, 向这些 Tracker 列表发送连接请求, 选择响应时间最短且同意加入的 Tracker, 向其注册本地基本信息 (拥有的资源情况等), 同时每隔一段时间向 Tracker 进行保活注册操作, 如果本地资源记录有变化则通知该 Tracker. 客户端退出系统, 首先要退出 P2P 网络, 通知自己的邻居节点和所有正在向自己请求数据的节点退出消息, 向所注册的 Tracker 通知退出消息.

3.3 播放影片

客户端播放一部影片时, 首先从节目列表中得到源服务器地址, 同时向 Tracker 获取一批拥有该影片数据的在线 Peer 地址 (Download Peers 包括正在观看和本地有数据而不在观看该影片的 Peer), 分别向这些 Peer 请求 BBM (base band modem) 信息, 同时把正在观看影片的 Peer 加入到邻居列表中. 当收到 Peer 的资源索引信息后, 即可向这些 Peer 发送数据下载请求. 当可下载的 Peer 数低于阈值时, 会从邻居列表中的 Peers 处获取它们当前使用的 Peer 来补充本地的记录. 所有拥有影片资源数据的节点组成了一个网状网络, 其中正在观看影片的节点作为该网络的建立和维护者的角色出现.

在下载数据的初始阶段, 源服务器作为一个普通的 Peer 加入到下载请求列表中, 数据会有一部分从源服务器获得, 当本地的缓冲数据到达一定数量后, 源服务器会逐渐退出下载任务, 数据下载全部向 Peer 获取. 当 Peer 的数量及资源分布不够时, 源服务器会被重新使用, 作为数据下载的补足, 保证数据下载及播放质量.

影片数据的下载是按照数据的请求顺序, 根据每个 Peer 的资源拥有情况和网络能力, 分别向不同的 Peer 分配分片的下载任务, 下载到的数据放入缓冲区中, 等待达到 10–30 s 播放时间的缓冲数据即可启动播放器播放影片, 同时继续下载到的数据会不断的填满缓冲.

已经播放过的影片数据会被存储到本地磁盘缓存中, 可以提供影片快速回拖播放的功能, 同时可以提供这些数据给网络中的其他节点使用.

3.4 拖动和停止

VOD 系统必须支持用户观看时的拖动操作, 当用户向前或向后拖动进度条, 定位到某个时间点开始播放影片时, 系统会先检查本地磁盘是否已有数据, 如有则直接快速启动播放器播放影片, 否则, 继续向正在使用的 Peers 下载所需的数据, 待缓冲区充足后再启动播放器播放影片.

当影片播放完毕或用户手动停止, 系统会通知当前的邻居节点自身的停止播放信息, 这些邻居节点会删除相应记录, 而在网络中也不再是正在播放某个资源的状态. 此时数据下载任务被结束, 但数据上传操作不会停止.

4 设计目标与难点

P2PVOD 系统设计目标主要包括从系统角度出发的服务器带宽节省率, 以及从用户体验角度出发的流畅度、拖动延迟. 同时对应每个设计目标分析了设计难点. 针对这些难点我们在第 5 节从客户端设计角度给出了解决方案, 在第 6 节中给出这 3 个性能指标的 PPLive 测量结果.

4.1 服务器带宽节省率

服务器带宽开销过大是导致 VOD 服务难以盈利的最主要原因, 同时直接影响 P2PVOD 系统可扩展性, 因此服务器带宽节省率可以说是评价 P2PVOD 系统的最根本指标. 指标越大, 性能越好, 最理想的结果是接近 100%, 即几乎所有下载流量可以通过 Peer 间传递数据解决. 我们给出模型定义如下.

假设系统中在 t 时刻有 M 个影片, 影片 i 有 N_i 个用户在线, 所有影片总共有 N 个用户在线. t 时刻影片 i 的第 j 个用户 (用 U_{ij} 表示) 从其他 Peer 下载数据量为 P_{ij} , 从服务器下载数据量为 S_{ij} . 那么 t 时刻, U_{ij} 的服务器带宽节省率为 R_{ij} , 影片 i 的服务器带宽节省率为 R_i , 整个系统的服务器带宽节省率为 R .

$$R_{ij} = \frac{P_{ij}}{P_{ij} + S_{ij}}; \quad (1)$$

$$R_i = \frac{1}{N_i} \sum_{j=1}^{N_i} R_{ij}; \quad (2)$$

$$R = \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^{N_i} R_{ij}. \quad (3)$$

假设网络性能给定的前提下, 提高服务器带宽节省率的难点在于: 一要 Peers 有足够多的潜在资源可以共享; 二要 Peer 能够选择一个或一组节点进行下载. 第一点主要由磁盘缓存策略决定,

第二点主要由任务调度策略决定. 在第 5 节分别给出了二者在 PPLive 中的实现策略. 下面具体介绍磁盘缓存策略和任务调度策略存在哪些具体的设计难点.

磁盘缓存策略包括以下几个方面:

(1) 单影片共享还是多影片共享. 所谓单个影片共享就是 Peer 只存储和共享当前正在观看的影片, 所谓多影片共享就是只要缓存空间足够就存储并共享所有观看过的影片. 单影片共享能够提供最简单的共享策略, 多影片共享策略给系统带来灵活性的同时也给设计带来了极大的复杂性. 设计时选择哪种策略是个难点.

(2) 共享磁盘缓存大小. 从系统角度越大越好, 但是从用户角度却是越小越好. 设计时如何做一个折中是一个设计难点. 我们认为会有这样一个拐点, 即当共享空间超过这个拐点时, 再增加共享空间性能不会有很大提升, 当小于这个拐点时, 性能会随共享空间减小急剧下降. 因此如何找到这个拐点也是一个研究难点.

(3) 各个影片副本最优分布. 即在整个系统共享磁盘空间一定, 影片数一定, 用户数一定的情况下, 各个影片在 Peer 间有多大比例副本的分布. 直观上认为与该影片的观看请求成正比的分布是一个不错的选择, 但是真正什么是最好的分布需要理论或实践验证.

(4) 如何设计淘汰算法达到该最优分布. 可以依靠 Tracker 统计全局的资源 and 需求分布, 并把统计结果分发给各个 Peer, 指导 Peer 磁盘进行缓存淘汰. 这种全局淘汰算法虽然能到达很好的指导效果, 但是限制了系统的可扩展性. 而只需局部信息的分布式淘汰算法可以改善可扩展性, 最直接的方法是淘汰最早观看的影片, 但势必影响淘汰效果. 设计中如何在全局淘汰算法与分布式淘汰算法做折中是一个难点.

任务调度策略应该满足下面几个条件:

(1) 稳定性, 当有多个伙伴节点可以为节点 A 提供数据时, 可以充分利用每个伙伴节点的资源进行选取数据, 当某个节点离开或不能提供数据时, 节点 A 还可以从其他伙伴节点获取数据, 保证了视频播放的稳定性.

(2) 自适应性, 为节点 A 提供数据的伙伴节点的服务能力不尽相同, 该算法可以自适应伙伴节点的服务能力的变化, 不断的调整.

(3) 负载均衡性, 节点 A 能够充分利用每个伙伴节点的资源, 在调整的过程中, 考虑到每个伙伴节点的服务能力, 做均衡的数据分配, 保证了节点之间的负载均衡. 保持当前在使用的 Peer 中有高、中、低连接速度分布, 即能保持高速下载, 又能充分利用到低上传带宽节点.

4.2 用户体验

用户体验最主要包括两个指标: 一是流畅度, 即在顺序播放的时候因数据下载速度过慢而造成的播放停顿的情况; 二是拖动延迟, 即拖动某位置后到缓存一定播放时长 (PPLive 为 20 s) 开始播放的时间间隔. 模型定义如下.

假设系统中有 M 个影片, 影片 i 有 N_i 个用户在线, 所有影片总共有 N 个用户在线. 假设第 i 个影片第 j 个用户 (用 U_{ij} 表示) 拖动 C_{ij} 次, $T_s(i, j)$ 表示拖动的起始位置对应的影片播放时间, $T_e(i, j)$ 表示此次拖动观看结束位置对应的影片播放时间, 意味着接下来进行下一次拖动, 那么 $T_e(i, j) - T_s(i, j)$ 表示有效观看时间, $T_a(i, j)$ 表示此次拖动中实际的观看用时, 由于存在拖动后缓冲时间以及观看过程由于下载速率不足的缓冲时间, 一般 $T_a(i, j)$ 大于 $T_e(i, j) - T_s(i, j)$. F_{ij} 表示用户 U_{ij} 的观看流畅度, F_i 表示影片 i 的观看流畅度, F 表示这个系统中所有影片的观看流畅度. $T_b(i, j)$ 表示拖动后缓冲足够

后开始播放的时刻. D_{ij} 表示用户 U_{ij} 平均拖动延迟, D_i 表示影片 i 平均拖动延迟, D 表示整个系统影片的平均拖动延迟.

$$F_{ij} = \frac{1}{C_{ij}} \frac{\sum_{k=1}^{C_{ij}} (T_e(i, j) - T_s(i, j))}{\sum_{k=1}^{N_{ij}} T_a(i, j)}; \quad (4)$$

$$F_i = \frac{1}{N_i} \sum_{j=1}^{N_i} F_{ij}; \quad (5)$$

$$F = \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^{N_i} F_{ij}; \quad (6)$$

$$D_{ij} = \frac{1}{C_{ij}} \sum_{k=1}^{C_{ij}} (T_b(i, j) - T_s(i, j)); \quad (7)$$

$$D_i = \frac{1}{N_i} \sum_{j=1}^{N_i} D_{ij}; \quad (8)$$

$$D = \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^{N_i} D_{ij}. \quad (9)$$

可以看出影响流畅度的核心因素为下载速度, 同样主要受到磁盘缓冲以及任务调度算法的影响. 当下载速度大于播放速度时, 即有持续的高效的 Peers 为其提供下载数据, 流畅度理论上可以达到 100%. 拖动延迟相对复杂, 影响此指标的因素主要包括: 一, 拖动后一批新的邻居 Peers 的获取及建立连接的时间; 二, 新邻居 Peers 相对下载方的上传能力.

在 VOD 系统中除了随机进入、离开之外, 还可以任意拖动、暂停. 而且每个用户在并不是拥有影片的所有情况, 比如一个用户观看到影片的一半, 只能提供该影片前半一半数据的共享. 这些使得 P2P 拓扑结构变化剧烈. VOD 系统的高动态性给提高拖动延迟带来极大的困难. 我们在实现时观察到, 当前保持下载的 Peers 很可能不具备拖动后观看点的数据, 尤其是当有大的幅度的拖动时这种问题就更加突出. 这样不得不重新寻找大批新的具有拖动后位置数据的邻居并重新建立连接, 而且要在尽快挑选出一些高下载速度的节点进行数据下载. 在这个过程中, 用户的下载速度会急剧下降. 因此如何提高拖动后新的下载邻居的选择速度, 并且尽快建立连接是一个设计难点.

5 用户体验

要实现系统的可扩展性降低服务器带宽消耗, 适应用户操作的动态性, 提高用户观看效果, 降低用户资源开销. 关键是优化网络中 Peer 的数据存储以及数据的动态调度.

5.1 客户端概览

客户端是 VOD 系统中实现 P2P 功能的主体, 既可以作为数据发送方, 也可作为数据接收方. 如图 3 所示, 按逻辑层次我们可以把客户端分为 3 个层次, 网络层、P2P 层和应用层, 网络层负责收发所有报文; P2P 层驱动网络数据请求, 负责分析高层的报文类型; 应用层负责媒体播放. P2P 层

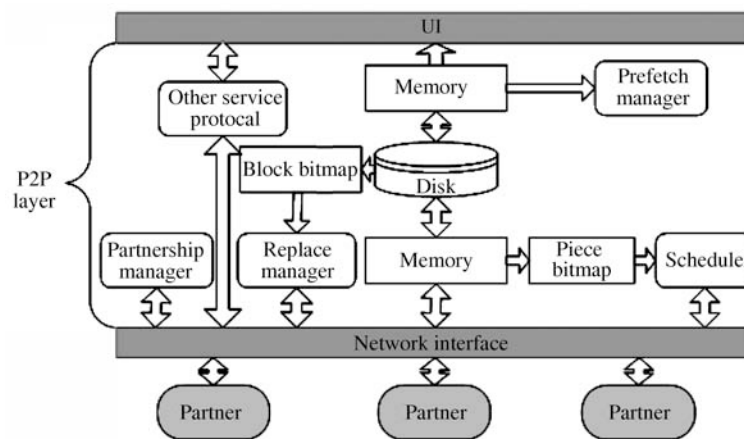


图 3 客户端分层结构

Figure 3 Client architecture

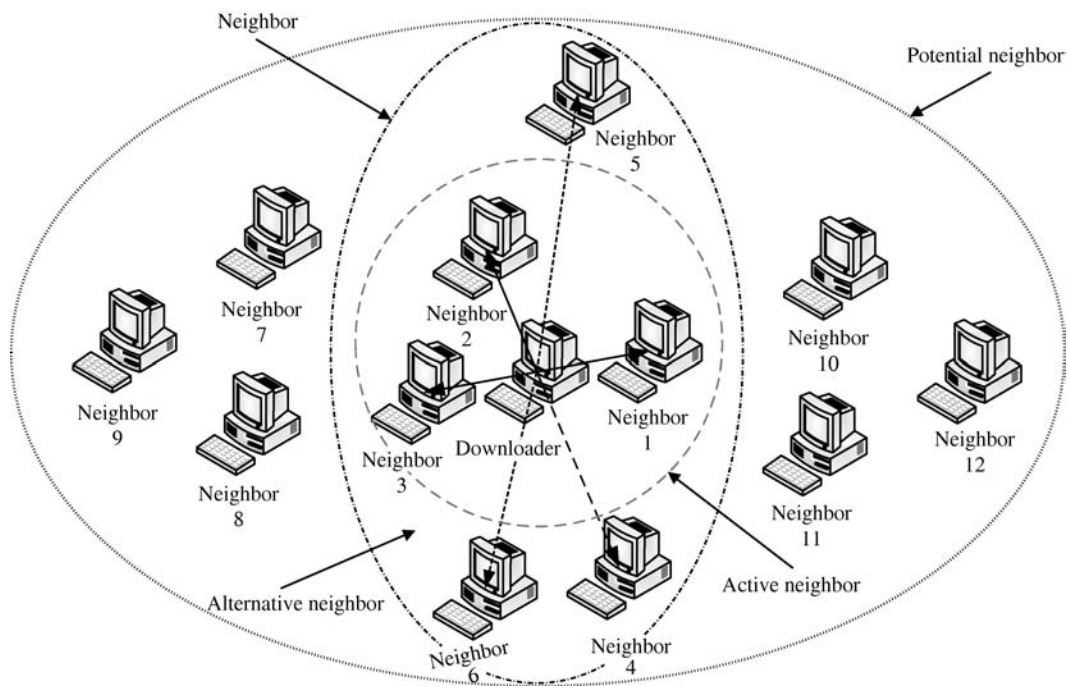


图 4 邻居关系图

Figure 4 Neighborhood

主要有 4 个关键模块: (1) Partnership Manager, 负责增加、删除、修改邻居节点信息, 维持彼此邻居关系; (2) Scheduler, 调度数据传输, 包括数据下载和上传; (3) Replace Manager, 负责磁盘缓存的淘汰策略; (4) Prefetch Manager, 负责内存数据淘汰及预取策略。

媒体数据的请求采用纯“拉”的模式, 由应用层驱动, 把请求顺序的交给 P2P 层, P2P 层负责从 P2P 网络中获取资源 Peer, 进行邻居关系维护、Peer 的选取、多点下载、任务调度等任务, 要满足流媒体实时播放的质量保证, 即要考虑数据的下载要满足节目的顺序、流畅播放。通过底层网络的数据传

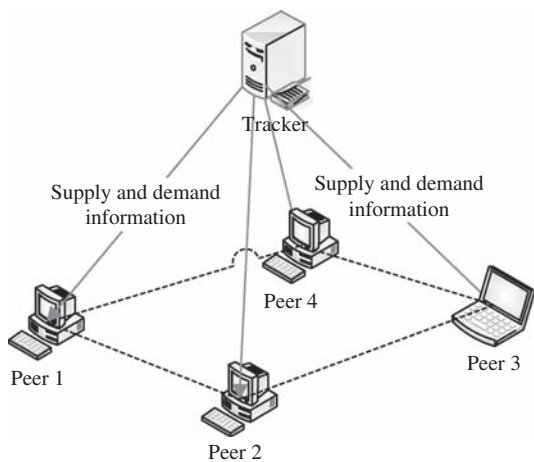


图 5 供需比信息收集与分发

Figure 5 Collection and publishing of supply-demand ratio

Peer 1 watches A and owns	Peer 2 watches C and owns	Peer 3 watches B and owns	Peer 4 watches D and owns
B50			
C100		C50	A50
D50	A50	D50	C50

图 6 资源分布

Figure 6 Resource distribution

输, P2P 层获得所需数据, 再交给应用层, 由应用层进行数据分片重组, 待缓冲数据满后再交给播放器播放. 分片数据及协议信令的传输都采用 UDP 报文方式, 以下的算法都是基于 UDP 方式进行设计.

5.2 数据下载

数据下载过程首先选定一组 Peer 与其建立邻居联系作为可能下载对象, 接下来根据这些邻居选取一组向其请求数据, 其他的作为备选邻居, 暂时不使用. 根据不同邻居对请求数据的返回速度, 动态调整分给各个邻居的下载任务. 在整个过程同时维护邻居关系, 引进一些新的邻居, 放弃一些连通性不好或没有索要数据邻居. 下面分别讲解邻居关系的维护以及任务调度.

5.2.1 邻居关系

图 4 给出了 4 种邻居关系.
潜在邻居: 系统中所有拥有用户观看资源的 Peer;

邻居: 客户端本地维护的一定数目的具有用户当前观看资源的 Peer, 包括活跃邻居和备用邻居;

活跃邻居: 指邻居中当前正在被该用户下载的 Peer;

备用邻居: 指邻居中除去活跃邻居的那部分 Peer. 当活跃邻居由于各种原因被淘汰时, 用来作为活跃邻居的备选. 例如没有当前请求的数据或者当前活跃邻居下载速度突然急剧变小, 低于最低播放速度, 启用备用邻居.

邻居关系维护模块通过动态调整使活跃邻居提供的总下载速度稍大于播放速度 (平均码流率). 维护几个备选邻居目的是当活跃邻居提供的下载速度不够时, 能够很快替换某些活跃邻居或直接加入活跃邻居, 达到快速增加下载速度的目的. 潜在邻居是当保持链接的邻居数太少时是可以通过 Tracker 或已经建立链接的邻居获取他们的列表来建立链接.

5.2.2 任务调度

初始化时, 根据潜在邻居分片资源拥有信息, 选取一组作为邻居, 并设定下载端与每个邻居的传输性能相同, 任务等概率分配给各个邻居. 所述传输性能可以用下载速度, 即下载每个分片所需要的时间描述. 根据任务分配向邻居并行发送数据请求, 接收返回的数据分片. 接下来, 下载端定时更新与各个邻居的下载速度, 重新分配下载任务, 下载速度高的分配任务相应多, 并发送数据请求接收数据. 根据各个分配拥有信息以及传输性能的变化信息, 定期判断和更新邻居及其任务分配的比例. 此调整过程一直进行, 直到数据传送完毕. 这样保证了动态网络环境下, 根据各个邻居的当前传输速度高低情况各尽其能, 使得整体下载数据保持稳定.

5.3 数据上传

客户端接收到其他 Peer 的数据请求, 尽力的快速公平的处理每个请求, 但有以下两种限定条件: (1) 上传限速, 防止过量的数据上传报文影响其他报文交互; 实时监测除上传报文以外的其他报文通信状态, 当出现上传过高而影响报文通信时, 重新设定减小上传限速值; (2) 连接限制, 把有限的带宽提供给有限的 Peer, 保证对每个 Peer 的服务质量, 不对所有 Peer 都开放上传; 根据上传速度的正反馈, 来设置可服务的 Peer 最大连接数, 上传速度越高, 最大连接数也会相应增加; 当 Peer 连接个数超过最大连接数值, 上传模块会返回不能连接的信息给新加入的请求 Peer; 而当上传速度超过最大值时, 直接进行丢请求包操作, 不做返回. 这种上传方只需公平的处理所有请求的数据上传策略不但程序简单, 实际运行中表明只通过下载方动态调整下载任务, 就能够达到数据上传方优先传输系统中稀缺的资源的目的.

5.4 磁盘缓存淘汰

理论研究表明在 P2P 网络中, 按需求按比例存储副本能够达到整体最优化的网络性能以及用户感受^[11,12]. PPLive 点播系统目前采用利用全局供求比信息的磁盘缓存淘汰算法来实现这一分布, 即选择磁盘缓存空间所有节目中供需比最大的节目, 并将属于该节目的所有块删除. 所谓全局供求比就是该节目各块的副本总和/观看该影片的在线人数. 收发示意图如图 5 所示. 假设有 4 部影片 A, B, C, D; 4 个 Peer, 各自观看的影片 (假设都是刚开始看选中的影片) 以及拥有的影片副本如图 6 所示, Tracker 收集和分发所有用户观看的影片以及存储的影片信息. A, B, C, D 各有一人观看,

$$A : 50 + 50 = 100, \quad B : 50, \quad C : 100 + 50 + 50 = 200, \quad D : 50 + 50 = 100.$$

假设 Peer1 此时缓存已满, 则从 Tracker 获取的供需比信息由大到小为 C, D, A. 那么选择 C 淘汰.

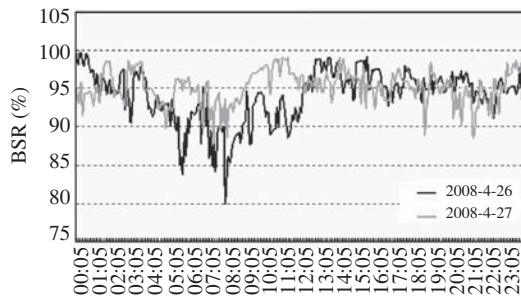


图 7 movie1 带宽节省率

Figure 7 BSR of movie 1

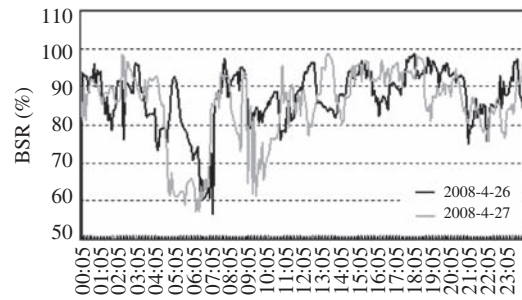


图 8 movie2 带宽节省率

Figure 8 BSR of movie 2

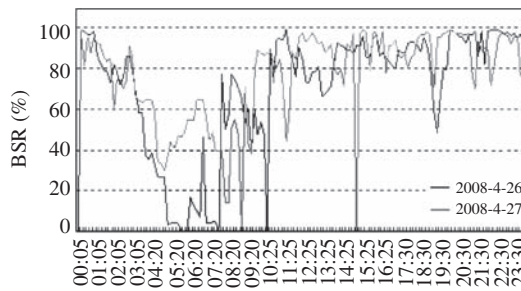


图 9 movie3 带宽节省率

Figure 9 BSR of movie 3

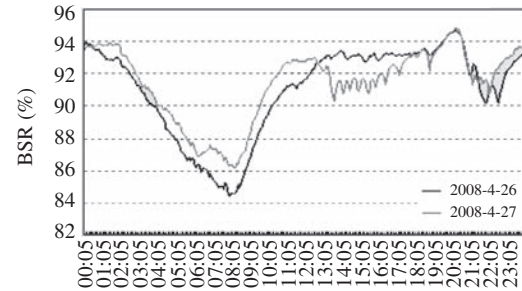


图 10 整体带宽节省率

Figure 10 BSR of the whole system

6 PPLive P2PVOD 性能评价

本节内容将用实际部署的大规模 P2PVOD 系统, 即 PPLive 视频点播系统在带宽节省率, 以及流畅度, 拖动延迟方面的性能指标来说明设计方案的有效性. 这些性能指标在第 3 节进行定义. 测试影片包括 3 部热度不同的单个影片, 以及系统中所有影片的整体数据, 测试时间为 2007-12-11 到 2007-12-17 七天, 以及 2008-4-26, 2008-4-27 两天的数据.

6.1 带宽节省率

为了定量带宽开销节省率, 利用 3.1 小节定义的公式 (2) 和 (3) 我们对 3 类不同热度的影片以及系统中所有影片进行了采样. 采样的 3 部影片分别是高热度的 movie1, 低热度的 movie3, 以及一般热度的 movie2. 这里热度越高表示在线人数越多. 采样时间为 48 小时, 即从 2008-4-26 0: 00 到 2008-4-27 24: 00.

图 7-9 所示为 3 部不同热度的影片服务器带宽节省率对比图. 从图中可以看出: (1) 带宽节省率有以天周期变化的趋势. (2) 单个影片服务器带宽节省率抖动很大, 而且热度越低, 抖动越大. 热度最高的 movie1 大部分在 85%–100% 之间, 热度居中的 movie2 带宽节省率分散在 60%–100% 之间, 热度最低的 movie3 分散在 0–100% 之间. (3) 热度越高带宽节省率越高, 平均值分别为 95%, 80%, 60% 左右.

图 10 为两天内系统所有影片整体服务器带宽节省率对比图. (1) 平均带宽节省率在 90% 左右.

表 1 同时在线人数达到峰值时带宽节省率

Table 1 Server bandwidth saving at peak time

Peak users	BSR	Playback rate	Saved bandwidth	RMB/((Gbps*month)	Saves/month
210000	93%	380 kbps	77.406 Gbps	50000	3870300

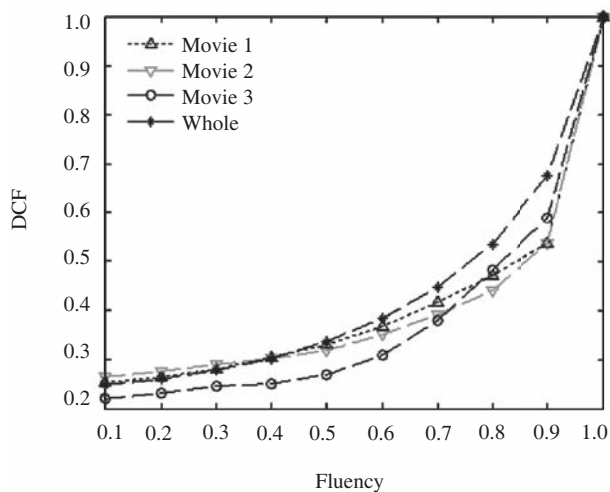


图 11 观看流畅度的分布

Figure 11 Viewing fluency distribution

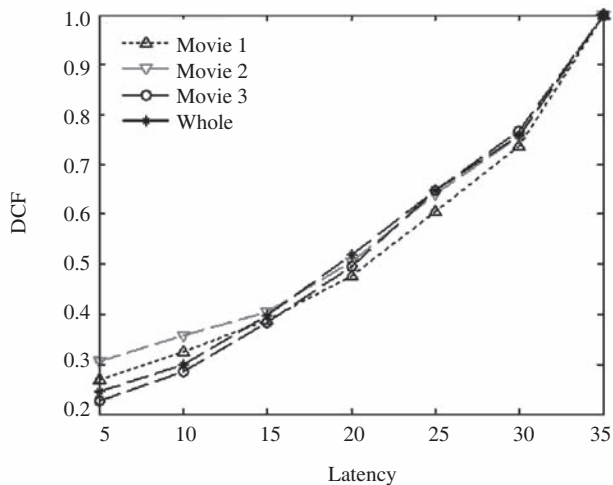


图 12 拖动延迟分布

Figure 12 Drag delay distribution

(2) 以天为周期有很强的规律性, 变化非常平滑. (3) 最低点出现在 8:00 左右, 接近 85%, 22:30 左右出现一个小低谷, 90%左右, 最高峰出现在 20:00 左右大约 95%. 根据用户在线信息可以分析以上结果的出现原因, 8 点左右会出现人数低谷, 很多影片由于人数过低 P2P 机制很难发挥效用, 造成了带宽节省率最低点; 22:30 的小低谷对应在一天内人数最多的时刻, 可见人数的激增可能造成带宽节省率的

下降。(4) 整体看服务器带宽节省率会随着人数增加而提高, 稍后我们会看到这种特性对于实际带宽节省率非常有益处, 因为人数峰值时的带宽决定了 VOD 服务提供商的购买带宽。

比较每日的平均带宽节省率, 人数峰值时的带宽节省率更有实际意义。因为此时基本是服务器付出带宽最高的时刻, 而 VOD 服务提供商购买的网络带宽是一个定值, 不可能早晨用的少就买得少, 晚上带宽用的多就多买, 为了所有时刻都能提供很好的服务, 购买的带宽为峰值时的带宽。人数峰值时的带宽节省率直接决定着实际购买带宽的节省情况。

表 1 所示为同时在线人数达到峰值时带宽节省率情况。可以看出峰值时带宽节省率比 88% 的平均带宽节省率要高, 为 93%, 在线人数 21 万人, 播放码率为 380 kbps, 实际节省带宽高达 77.406 Gbps, 按现在 1 Gbps 每月 5 万元计算, 每月节省费用为 387.03 万元。

6.2 用户体验

为了量化流畅度以及拖动延迟, 利用 4.2 小节定义的公式 (5), (6), (8), (9) 我们对 3 类不同热度的影片以及系统中所有影片进行了采样。采样的 3 部影片分别是高热度的 movie1, 低热度的 movie3 以及一般热度 movie2。采样时间为 24 小时即从 2008-4-21 00:00 到 2008-4-21 24:00。

图 11 所示为 3 部不同热度的影片以及所有影片的观看流畅度的累计分布。由图可以看出整体来看有 68% 的观看流畅度小于 0.9。不同热度的影片的流畅度相差不大, 可以看到不同热度的影片都能得到公平的支持, 即便不能从 Peer 获取足够多的下载速度, 影片服务器可以提供额外的下载流量。流畅度低的原因在于 Log 服务器进行统计的方法, 拖动以及暂停的时间也算到等待时间里。

图 12 所示为 3 部不同热度的影片以及所有影片的拖动时间的累计分布。由图所示有 70% 以上的拖动时间在 30 s 以内。启动速度从 5 到 35 s 分布均匀, 启动速度主要由拖动后找到的邻居节点下载速度决定。由于拖动后采用随机的选取邻居节点下载, 不同热度的影片的启动拖动时间相差不大。

7 总结及下一步工作

本文首先提出了一种混合式的 P2P 网络结构, 以 Tracker 为索引服务器, 提供最基本的资源查找功能。同时 Peer 之间建立非结构化的网状网络, Peer 也可以通过该网络进行资源的定位和查找。基于这种结构介绍了流媒体数据存储和分发过程以及用户操作行为。基于此模型分析了 P2PVOD 设计面临的主要难点, 我们认为要克服这些难点的关键在于充分发挥 Peer 端的能力, 尽量减少服务器端的负载。主要技术手段是通过研究数据副本如何分布更合理并设计出相应的淘汰算法, 动态维护邻居关系和调整数据下载任务。客户端策略的设计在解决这些难点中发挥了决定性的作用。结合 PPLive 推出的 P2PVOD 系统, 给出了客户端的设计模型、磁盘缓存淘汰算法、内存缓存预取算法、数据下载以及数据上传策略。最后对 PPLive P2PVOD 系统运行情况包括服务器带宽节省率、观看流畅度、拖动速度分别从所有影片以及 3 类热度不同的影片进行了分析。数据表明系统的平均服务器带宽节省率可高达 88% 以上, 人数峰值带宽节省率高达 93% 以上。本文以实际系统验证了 P2P 技术在 VOD 服务提供中的巨大优势, 并可以为 P2PVOD 研究和设计者提供一个关于客户端设计的参考样本。

在研究过程中, 我们观察到同一部影片内各数据块的分布很不均匀, 通常是前多后少, 下一步我们计划考虑部署更细粒度的磁盘淘汰算法。同时我们考虑为 Tracker 提供邻居时做进一步筛选, 目标是进一步提高拖动后的响应速度。

参考文献

- 1 Zhang X, Liu J, Li B, et al. Coolstreaming/donet: A data-driven overlay network for efficient live media streaming. In: Proceedings of INFOCOM. Miami, FL, USA, 2005. 2102–2111
- 2 PPLive. <http://www.pplive.com/>
- 3 PPStream. <http://www.ppstream.com/>
- 4 GridCast. <http://www.gridcast.cn/>
- 5 YouTube. <http://www.youtube.com/>
- 6 YouKu. <http://www.youku.com/>
- 7 Huang C, Li J, Ross K W. Can internet video-on-demand be profitable? In: Proceedings of ACM SIGCOMM'07, Kyoto, Japan, 2007. 133–144
- 8 Annapureddy S, Gkantsidis C, Rodriguez P R, et al. Providing video-on-demand using peer-to-peer networks. Microsoft Research Technical Report. MSR-TR-2005-147, Oct. 2005
- 9 Huang C, Li J, Ross K W. Peer-assisted VOD: Making internet video distribution cheap. In: Proceedings of IPTPS. Bellevue, WA, USA, 2007. 1–5
- 10 PPLIVE-VOD. <http://www.pplive.com/subject/20070808pfsvod/>
- 11 Cherkasova L, Lee J. FastReplica: Efficient large file distribution within content delivery networks. In: Proceedings of 4th USENIX Symposium on Internet Technologies and Systems. Seattle, WA, USA, 2003. 7
- 12 Ali S, Mathur A, Zhang H. Measurement of commercial peer-to-peer live video streaming. In: Proceedings of 1st Workshop on Recent Advances in P2P Streaming. Hilton, Singapore, 2005. 1–6
- 13 Hei X, Liang C, Liu Y, et al. A measurement study of a large-scale P2P IPTV system. IEEE Trans Multimedia, 2007, 9: 1672–1687
- 14 Wu C, Li B, Zhao S. Multi-channel live P2P streaming: refocusing on servers. In: Proceedings of IEEE INFOCOM 2008, Phoenix, AZ, USA, 2008. 1355–1363
- 15 Hei X, Liu Y, Ross K W. Stochastic fluid theory for P2P streaming systems. In: Proceedings of INFOCOM 2007, Anchorage, AK, USA, 2007. 919–927
- 16 Zhou Y, Chiu D M, Lui J C S. A simple model for analyzing P2P streaming protocols. In: Proceedings of IEEE ICNP 2007, Beijing, China, 2007. 226–235
- 17 Cha M, Kwak H, Rodriguez P, et al. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In: Proceedings of IMC. 2007, San Diego, California, USA. 2007. 1–14
- 18 Al Hamra A, Biersack E W, Urvoy-Keller G. A pull-based approach for a VoD service in P2P networks. In: Proceedings of HSNMC 2004, Toulouse, France, 2004, 3079: 995–1006
- 19 Guo Y, Suh K, Kurose J, et al. P2Cast: Peer-to-peer patching scheme for VoD service. In: Proceedings of WWW 2003, Budapest, Hungary, 2003. 301–309
- 20 Tai D, Kien H A, Mounir T. P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment. In: Proceedings of ICC 2004, Orlando, FL, USA. 2004, 3: 1467–1472
- 21 Wang D, Liu J. Peer-to-peer asynchronous video streaming using skip list. In: Proceedings of ICME 2006, Toronto, Ont, Canada. 2006. 1397–1400
- 22 Cui Y, Li B, Nahrstedt K. Ostream: A synchronous streaming multicast in application-layer overlay networks. IEEE JSAC, 2004, 22: 91–106
- 23 Wang D, Liu J C. A dynamic skip list-based overlay for on-demand media streaming with VCR interactions. IEEE Trans Paral Distr Syst, 2008, 19: 503–514
- 24 Guo Y, Yu S C, Liu H, et al. Supporting VCR operation in a mesh-based P2P VoD system. In: Proceedings of Consumer Communications and Networking Conference. Las Vegas, NV, USA, 2008. 452–457
- 25 He Y, Liu Y H. Supporting VCR in peer-to-peer video-on-demand. In: Proceedings of ICNP 2007, Beijing, China, 2007. 328–329
- 26 Cheng B, Jin H, Liao X F. Supporting VCR functions in P2P VoD services using ring-assisted overlays. In: Proceedings of ICC 2007, Glasgow, Scotland. 2007. 1698–1703

- 27 Cheng B, Stein L, Jin H, et al. Towards cinematic internet video-on-demand. In: Proceedings of the 3rd ACM SIGOPS/EuroSys 2008, Glasgow, Scotland UK, 2008. 109–122
- 28 Guo Y, Mathur S, Ramaswamy K, et al. PONDER: Providing commercial-quality video-on-demand service using peer-to-peer network. In: Proceedings of GLOCOM 2007, Washington, DC, USA, 2007. 225–230
- 29 Tewari S, Kleinrock L. Analysis of search and replication in unstructured peer-to-peer networks. In: Proceedings of ACM SIGMETRICS 2005, New York, NY, USA, 2005. 404–405
- 30 Tewari S, Kleinrock L. Proportional replication in peer-to-peer networks. In: Proceedings of INFOCOM 2006, Barcelona, Spain, 2006. 1–12

Large-scale P2PVOD system: Focusing on clients

HUANG Yan¹, XU Ke^{2*}, LI HaiTao², CAO Yang¹ & YAO Xin¹

¹ Shanghai SynaCast Media Tech Co., Ltd, Shanghai 201204, China;

² Department of Computer Science & Technology, Tsinghua University, Beijing 100084, China

*E-mail: xuke@tsinghua.edu.cn

Abstract Using P2P network to provide video on demand (VOD) service can support large-scale users so as to save the bandwidth cost of servers. As VOD must support VCR characters (drag, pause, randomly choosing watching position, etc.), building a large-scale expansion of high-performance VOD system still faces huge challenges. The paper first presents a P2P based reference model, defines the design object, and analyzes the difficulties. And then taking PPLive demand system as an example, it gives client-based solutions, including disk cache-out strategy, memory cache prefetch strategy, maintaining relations between neighbors, task scheduling, data upload strategies and so on. PPLive demand system is putting into practice at present with satisfactory fluency and drag delay. The largest number of simultaneous online users is over 210,000, and online video data is over 700. Compared with traditional client-server model, the average bandwidth saving rate of servers is 88% or more, and the bandwidth saving rate during the peak time can be up to 93%.

Keywords P2P, VOD, server bandwidth saving rate, fluency, drag delay