文章编号: 1001- 2486(2010) 06- 0059- 05

# 一种新型的 P2P Tracker 实现方法*

黄金锋, 张子文, 程高伟
(国防科技大学 计算机学院, 湖南 长沙 410073)

摘 要: 流媒体数据已经在互联网流量中占据极高的比例, P2P 是目前支撑互联网流媒体数据分发的重要手段。本文提出一种新型的 P2P Tracker 实现方法——eTracker。该方法通过网络边缘的 eTracker 分布实现 peer 列表的保存, 不但消除传统集中式 Tracker 的性能瓶颈, 而且通过 eTracker 对本地 peer 的识别, 优化了 P2P 系统的 peer 选择。

关键词: 流媒体; P2P; peer 选择; 复杂性; 本地化
中图分类号: TP393. 02 文献标识码: A

# An Innovative Implementation Method of P2P Tracker

HUANG Jin-feng, ZHANG Zi-wen, CHENG Gao-wei
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract**: Streaming media accounts for the majority of Internet traffic. P2P is an important method for streaming media distribution on the Internet. This paper proposes an innovative implementation method of P2P Tracker named eTracker. Peer lists are reserved in the numerous distributed eTrackers which locate in the edge network. Our design not only eliminates performance bottleneck of traditional centralized Tracker, but also optimizes peer selection in P2P system through the recognition of local peers.

**Key words**: streaming media; peer-to-peer; peer selection; computational complexity; locality

近年来, 流媒体数据已经在互联网流量中占据极高的比例[1-2], 流媒体对象的数量也迅速增长[3-4]。高效实现流媒体数据的分发对整个互联网的发展具有越来越重要的意义。P2P 是目前支撑互联网流媒体数据分发的重要手段, PPLive[5]、CoolStreaming[6] 等基于 P2P 的流媒体分发系统得到广泛的应用。例如, 2008 年 1 月, 互联网大规模视频直播软件 PPLive 同时在线的用户数已经超过 15 万。

由于 P2P 应用系统的 peer 选择建立在层叠网之上, 难以考虑物理网络带宽等资源利用情况, 因此 P2P 应用在满足用户需求的同时可能会造成带宽的浪费, 因此 P2P 应用模式与 ISP( Internet Service Provider) 的矛盾逐渐显现[7], 如何解决 P2P 应用的网络效率问题得到广泛关注。P4P[7] 等由 ISP 引导进行 peer 选择的技术逐渐出现, IETF 也专门为应用层流量优化成立 AUTO 工作组[8], 对相关的协议进行标准化。P4P 等技术必须得到 ISP 设置的 iTracker 的支持, 因此其应用程序的部署必须受限于 ISP 提供的服务。

为解决 peer 选择优化对 ISP 依赖的问题, 本文提出了基于 eTracker( edge Tracker) 的 P2P 系统 peer 选择技术。该技术通过网络边缘的 eTracker 分布实现 peer 列表的保存, 不但消除传统集中式 Tracker 的性能瓶颈, 而且通过 eTracker 对本地 peer 的识别, 优化了 P2P 系统的 peer 选择。

## 1 基于 Tracker 的 peer 选择

设 P2P 系统有 $m$ 个数据对象, 保存第 $i$ 个数据对象的 peer 共有 $s_i$ 个。基于 Tracker 的 P2P 系统进行 peer 选择时, 对于每个新加入 P2P 系统的客户端都希望 Tracker 服务器上保存有如图 1( a) 所示阵列。即对于任意对象 $i$, Tracker 可以对其 $s_i$ 个 peer 进行全局排序, 最优的( 如物理位置最近、带宽最高等) 为

* 收稿日期: 2010- 08- 01
基金项目: 国家"863" 计划基金资助项目( 2008AA01A323; 2008AA01A325)
作者简介: 黄金锋( 1965—) , 男, 副研究员, 硕士。

$p_{i,1}$, 其次为 $p_{i,2}$, 最差的为 $p_{i,s_i}$, $M$ 为媒体对象的数目。一旦该用户请求对象 $i$ 的 peer, Tracker 会按照排列的由高至低的优先级返回 peer 的集合。然而由于收集信息的不完整性, 应用程序的 Tracker 根本无法提供图 1(a) 所示阵列。所有的 peer 只能按照图 1(b) 所示阵列保存, 其中 $pg_i$ 是保存对象 $i$ 的 peer 集合。

$$\begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,s_1} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,s_2} \\ \vdots & \vdots & & \vdots \\ p_{M,1} & p_{M,2} & \cdots & p_{M,s_M} \end{bmatrix} \qquad \begin{pmatrix} pg_1 \\ pg_2 \\ \vdots \\ pg_M \end{pmatrix}$$

(a) 理想情况下的 peer 组织　　(b) 实际保存的 peer 信息

图 1　Tracker 上 peer 信息的组织

Fig. 1　Data structure of peers in Tracker

由于集合中元素是无序的, Tracker 很难为每个加入的用户生成理想的 peer 阵列。因此 Tracker 向用户返回的 peer 可能是非最优的, 甚至是相对较差的 peer( 如网络吞吐量低、RTT 时间大, 消耗网络资源多等), 造成 P2P 应用程序与 ISP 的矛盾, 也影响了 P2P 应用的性能。

目前 peer 选择优化的主要思路是: Tracker 接收到用户对数据对象 $i$ 的请求后, 将集合 $pg_i$ 展开成向量: $\overrightarrow{pg_i}\left[p_{i,1} \quad p_{i,2} \quad \cdots \quad p_{i,k}\right]$。其中 $p_{i,j}$ 按优先级从高至低排序。向量展开主要依赖以下两种方法: 一是 Tracker 根据应用层的测量对 peer 进行排序, 如 Eugene Ng[9] 提出的通过对 RTT、TCP 吞吐量、瓶颈带宽等因素进行端到端测量来评选性能好的 peer 返回给请求者。二是基于 P4P 的思想, Tracker 借助 ISP 的 iTracker 帮助排序。这两种方法在用户请求时动态对集合进行展开, 最大缺点是随着 $M$ 数目和 peer 数目的增加, Tracker 会成为性能瓶颈。

## 2　eTracker 基本原理

eTracker 的基本思想是把 P2P Tracker 的功能分布到位于核心网络边缘的多个 eTracker 上实现。每个 eTracker 负责维护其周围边缘网络内部的 peer 的信息。若 eTracker 无法满足用户的 peer 请求时, 再从全局的 Tracker( 称为主 Tracker) 中获取其他网络中的 peer 信息, 并返回给用户。

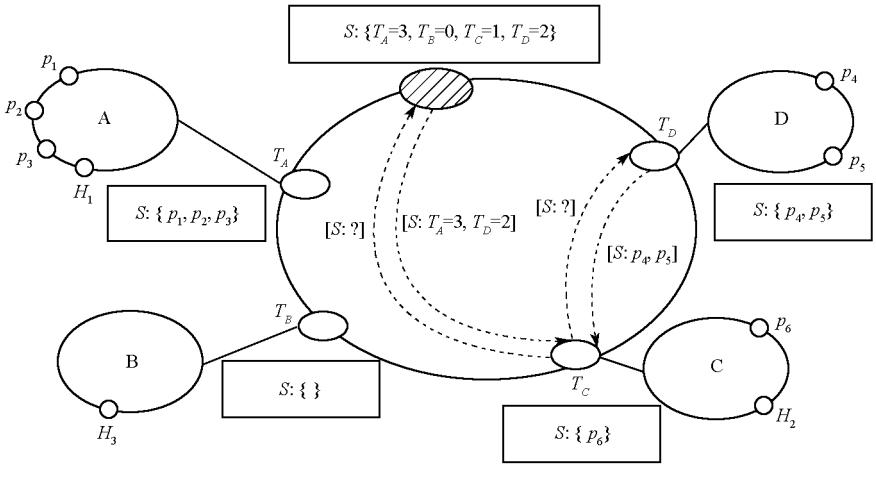


图 2　eTracker 的基本工作原理

Fig. 2　Basic principle of eTracker

eTracker 的基本工作原理如图 2 所示。其中 T 表示 P2P 应用的主 Tracker, $T_A$、$T_B$、$T_C$ 和 $T_D$ 分别为 4 个分布在网络边缘的 eTracker, 每个 eTracker 分别维护 $A$、$B$、$C$ 和 $D$ 这四个边缘网络中的 peer 信息 $p_1$、$p_2$、$p_3$、$p_4$、$p_5$ 和 $p_6$。$T_A$ 中保存记录 $S:\{p_1, p_2, p_3\}$ 表示网络 $A$ 中有 $p_1$、$p_2$ 和 $p_3$ 三个 peer 可提供数据对象 $S$。主 Tracker $T$ 中记录 $S:\{T_A = 3, T_B = 0, T_C = 1, T_D = 2\}$ 表示在边缘网络 $A$、$B$、$C$ 和 $D$ 中分别有 3

个、0 个、1 个和 2 个可提供数据 $S$ 的 peer。

主 Tracker 把 peer 管理功能重定向到就近的 eTracker 上实现, 例如主机 $H_1$ 的请求由 $T_A$ 完成, 那么当 $H_1$ 请求对象 $S$ 时, $T_A$ 根据其内部记录, 按照一定策略选取 peer 的信息返回。当 peer 的上载总能力不能够满足主机对于内容的请求时, 由代理的 eTracker $T_i$ 向 $T$ 发送询问请求, 由 $T$ 根据内部记录按照一定策略选择若干个 eTracker 返回, $T_i$ 可以从该结果的集合中选取若干个 eTracker 发出请求, 如图 2 中所示 $T_C$ 先向 $T$ 发起请求后向 $T_D$ 询问 peer 信息的交互过程。在请求用户获得 peer 列表开始下载之后, 它也成为新的 peer, 需要修改 eTracker 和 Tracker, 如 $H_2$ 获取数据对象 $S$ 后, $T_C$ 的记录将会改为 $S:\{P_6, H_2\}$, $T$ 的记录相应的改为 $S:\{T_A=3, T_B=0, T_C=2, T_D=2\}$。

由以上工作流程可知, 对于用户端 peer 的请求, eTracker 优先选择本地的 peer 使得流量尽可能本地化[10], 只有在本地 peer 无法满足需求的时候, 才返回非本地的 peer, 这会产生骨干网络流量。与 P4P 技术相比, eTracker 根据自身的物理位置对 peer 进行划分, 如根据接入的边缘网络划分, 在不需 ISP 辅助的情况下就可以实现 peer 选择的优化。

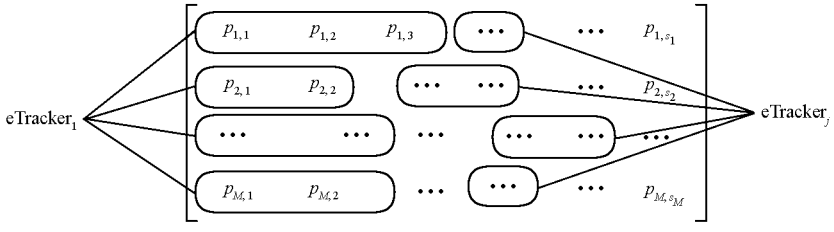


图 3 基于 eTracker 的 peer 聚合
Fig. 3 Peer aggregation based on eTracker

与图 1(a) 所示理想情况下 peer 组织相比, eTracker 实现了近似最优的 peer 组织。设 P2P 应用将网络划分为有 $K$ 个边缘网络区域, 每个区域对应一个 eTracker, 那么对于任意对象 $i$, eTracker 方法实际上是将其 $s_i$ 个 peer 分成 $K$ 组, 然后按照 1 至 $K$ 的 eTracker 顺序进行排序 $PG_{i,1}$, $PG_{i,2}$ …$PG_{i,K}$。

eTracker 方法事实上是将图 1(a) 所示阵列划分成 $K$ 个块, 保存到分布在网络边缘的 $K$ 个 eTracker 上, 如图 3 所示。块划分和 peer 聚合之后, 阵列被化简为一个 $M \times K$ 的矩阵, 而主 Tracker 保存的就是这个矩阵, 其中每一项都代表经过简化之后每个 eTracker 拥有对象 $S$ 的控制信息, 如有多少个 peer。对于区域 $j(j=1, 2, \cdots, k)$ 中的 peer 请求, $PG_{i,j}$ 中的 peer 具有最高优先级, 其他 peer 的优先级由主 Tracker 根据一定策略决定。

## 3 性能分析

本文对 eTracker 体系结构的性能评价包括对存储开销和计算开销的分析。存储开销包括了 eTracker 和主 Tracker 服务器的存储空间耗费。计算开销为用户加入 P2P 系统之后请求 peer 的开销、用户 peer 新加入以及离开系统对于 eTracker 和主 Tracker 的开销, 另外由于 peer 性能的动态变化还需要考虑对每个 peer 的链表进行重排序以及 peer 不足时发起新的 peer 请求的开销。

### 3.1 存储开销

在 eTracker 的框架下, 由于内容对象的数目极大, 为了便于 peer 的查找、插入和删除等操作本文采用链表地址法保存 peer 信息。以对象 $S$ 的名称作为 key, Hash(key) = $d$, 以 $d$ 作为地址访问对象表, 对象表中的每一项记录了在该 eTracker 的范围内对象 $S$ 的 peer 链表。当发生 hash 冲突, 即不同对象名映射到同一个 hash 值时, 我们采用分离的同义词子表解决冲突。

对于对象 $i$, eTracker $j$ 存储开销主要为保存该对象 peer 信息的开销。每个 peer 的信息典型包括: 对象名、peer 序号、peer 地址、端口号、上传速率等, 设上述 peer 基本信息的存储开销为 $M_{\text{peer}}$。eTracker 中链表的索引范围为 0~ $L_1-1$, 那么对于 $K$ 个 eTracker 的 hash 查找表的总存储开销为 $O(L_1 * K)$。每个

eTracker中都保存各自peer属性和指针的开销,故所有peer的总存储开销为$(M_{peer}+\log_2 L_1)\sum_{j=1}^{K}\sum_{i=1}^{M}e_{ij}$,其中$e_{ij}$表示边缘网络$j$内对象$i$的peer个数。因此eTracker的总开销为$L_1 * K+(M_{peer}+\log_2 L_1)\sum_{j=1}^{K}\sum_{i=1}^{M}e_{ij}$。本设计根据对象的个数可以调节$L_1$值,以在存储开销和hash冲突之间取得较好的折中。

对于主Tracker来说它记录了基于所属eTracker分块简化后的各个边缘网络中每个对象的控制信息,如在每个eTracker范围内某个对象的数目。我们也可以采用链表地址法保存eTracker信息,其查找表的开销为$O(L_2)$,对于每个对象对应eTracker的存储开销就为$\sum_{i=1}^{M}d_i M_{eT}$,其中$d_i$是拥有内容$i$的eTracker数目,$M_{eT}$是每个eTracker项保存的信息的存储开销。

## 3.2 计算开销

### 3.2.1 响应用户peer请求

用户对Tracker服务器的请求主要为查找peer列表,响应用户的请求有两种情况。第一种情况是边缘网络内部的peer可以满足需求时,eTracker直接返回这个对象的合适的peer列表。查找首先进行hash处理,可在常量时间完成,在冲突情况下通过折半查找找寻所需要的对象名,则查找的开销为$O(\log_2 H)$,其中$H$是散列冲突的平均冲突深度(当不冲突时$H=1$)。由于每个对象的peer列表按照上载能力的降序排列,因此在返回用户请求时只需按要求取一定数目的peer即可。设用户与本地eTracker的通信时间为$R_1$,则总的时间复杂性为$O(R_1+\log_2 H)$。

第二种情况边缘网络内部的peer不能满足需求,这时需要引发远程请求,远程交互的时间开销为$R_2$。主Tracker根据用户的请求基于对象名进行散列查找,由于冲突的情况相同,其时间复杂性也为$O(\log_2 H)$。随后请求用户所在区域的eTracker根据主Tracker的应答远程询问其他eTracker的内容情况,需要再次进行peer在eTracker上的查找,所以第二种情况的时间为$O(R_1+2R_2+3\log_2 H)$。

综上所述,设用户的peer请求不能在本区域得到满足的概率为β,那么用户对于peer请求总的时间复杂性为$R_1+2\beta R_2+(1+2\beta)\log_2 H$,由于$R_1>R_2$,本算法尽可能地减少参数β,使得peer请求尽可能在本地完成。

### 3.2.2 维护peer链表开销

当有新的peer加入或者离开P2P系统时都会引发peer链表的插入和删除,查找开销$O(\log_2 H)$是必需的。eTracker体系结构中,每个对象在一个eTracker内的peer是按照上载能力进行降序排序的,这样可以保证优先返回给用户上载能力强的peer。由于顺序排列,插入一个peer可以根据其上载能力采用二分插入的办法,那么对于eTracker所需要的时间开销为$O(\log_2 e_{ij})$。删除与加入peer节点的情况类似,故插入和删除操作对于eTracker的时间复杂性都是$R_1+\log_2 H+\log_2 e_{ij}$。另外一方面插入删除peer也可能引起主Tracker的内容更新,其时间开销为$R_2+\log_2 H$。

由于peer的上传能力处于动态变化中,所以eTracker需要定期对peer的排序进行刷新,该过程与所有peer建立通信连接,然后根据返回的上载能力进行peer的重新排序。设eTracker与peer的通信是并行的,本设计采用复杂度最低的折半插入排序,那么总的时间复杂性为$R_1+e_{ij}\log_2 e_{ij}$。

### 3.2.3 跨区域的peer选择优化

eTracker的部署相对稳定,peer之间的网络开销通过eTracker之间的开销就可以得到较为准确的表示,开销包括距离、网络带宽情况等网络基本信息。主Tracker上记录各eTracker之间的网络开销能够帮助peer选择开销小的eTracker内的peer节点。主Tracker可以根据事先度量和保存下来的eTracker之间的开销优先选择距离当前eTracker代价小的eTracker中的节点作为数据源。这样能够显著的减少跨多个域的流量或者带宽瓶颈链路,从而避免网络拥塞,同时减少用户的访问延时。

eTracker之间开销需要通过预先的网络测量获得,如度量任意两个eTracker之间的距离,然后保存

在二维数组中。由于其值保持相对稳定, 能够极大的减少计算复杂性。在 eTracker$_j$ 发起 peer 请求时, 主Tracker 需要将目前拥有该对象所需内容的 $e$ 个 eTracker 按照距离 eTracker$_j$ 的网络开销降序排列。然后根据用户请求的 peer 数目返回开销小的节点, 则其时间复杂性为 $O(e\log_2 e)$。

表 1 P2P Tracker 实现的性能比较

Tab. 1 Performance compare of P2P Tracker implementation

| | 存储开销 | 响应请求 | 维护 | peer 选择优化 |
|---|---|---|---|---|
| 传统 Tracker | $L_2 + (M_{\text{peer}} + \log_2 L_2) \cdot \sum_{j=1}^{K}\sum_{i=1}^{M} e_{ij}$ | $R_2 + \log_2 H$ | 插入删除: $R_2 + \log_2 e_i$<br>刷新: $R_2 + \log_2 e_i$ | × |
| eTracker | $L_1 * K + (M_{\text{peer}} + \log_2 L_1) \cdot \sum_{j=1}^{K}\sum_{i=1}^{M} e_{ij} \sum_{i=1}^{M} d_i M_{eT}$ | $R_1 + 2\beta R_2 + (1+2\beta)\log_2 H$ | 插入删除: $R_1 + \log_2 H + \log_2 e_{ij}$<br>刷新: $R_1 + e_{ij}\log_2 e_{ij}$ | $O(e\log_2 e)$ |

综上所述, eTracker 方式与传统 Tracker 方式的存储和计算复杂性比较如表 1 所示。通过上面的分析可以得出, 当用户对于 peer 的请求对象不能在本地的 eTracker 上得到满足, 就需要向远程的主 Tracker 和其他的 eTracker 发出请求, 远程通信开销 $R_2$ 是主要的性能瓶颈, 同时选择的 peer 在距离远的地方会导致传输延时增大, 浪费网络带宽。这种情况下的方法是在 eTracker 上增加数据平面, 补充 peer 对于内容对象的需求。分别增加热门和冷门缓存, 从而尽可能地在网络边缘满足用户请求。

P4P 是运营商主动参与, 并提供网络底层信息来优化 peer 的选择。而 eTracker 根据一定的规则人为的划分一定规模大小的边缘网络, 直接返回边缘网络内部的 peer 资源, 系统的响应时间会更快。P4P 和 eTracker 都是通过对 peer 的本地化选择来对流量进行优化的, 只是使用的方式不同。

## 4 结束语

视频流量占网络流量的比例越来越大, 而网络本身对于视频流媒体不能提供很好的支持。不断增大的视频流量需求, 给网络带来很多的问题, 最终导致服务质量的下降。

网络边缘是服务的重要实施点, 通过部署在网络边缘的 eTracker 服务器能够对流媒体传输进行更好的优化, 如 peer 本地化、缓存等。这种具有 eTracker 服务功能的边缘基础设施既可以由内容提供商建设(如在边缘使用磁盘阵列缓存该内容提供商的热点内容), 也可以由 ISP 统一建设后由内容提供商根据需要购买资源, 具有较大的灵活性。新型路由器模型的发展, 为 eTracker 服务器的部署提供了方便, 各个应用程序可以根据需求在路由器部署自己的 eTracker 服务器, 以对本应用程序的性能进行优化。这种优化是在不损坏别的应用程序的情况下的共赢的优化。

基于 eTracker 的边缘网络传输优化系统, 确实对系统的整体性能带来很大的改善, 也节约了很多出口流量, 降低了运营商的成本。在系统的实际部署之前, 还有很多工作要做。下一步主要研究冗余流量的减少、报文缓存的写入和读出对于报文转发平面的影响等, 从而使系统能够以比较低的代价获得流媒体传输的较高性能。

## 参 考 文 献:


[1] Schulze H, Mochalski K. Internet Study 2008/2009 [R]. www.ipoque.com, 1999.

[2] Wu H Q. 2007 BroadBand World Forum Asia [R]. http://www.iec.org/events/2007/bbwf_asia/.

[3] Inktomi. Streaming Media Caching White Paper [R]. http://www.inktomi.com/products/traffic/streaming.html, 1999.

[4] Gibson G, Vitter J, Wilkes J. Storage and I/O Issues in Large-scale Computing [J]. ACM Workshop on Strategic Directions in Computing Research, 1996.

[5] Huang Y, Fu T, Chiu D, et al. Challenges, Design and Analysis of a Large-scale P2P VoD System [C]// ACM SIGCOMM'08, 2008.

[6] Zhang X Y, Liu J C, Li B, et al. CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-peer Live Media Streaming [C]// IEEE INFOCOM'05, 2005.

[7] Xie H, Yang Y, Krishnamurthy A, et al. P4P: Provider Portal for Applications [C]// ACM SIGCOMM'08, 2008.

[8] Application-Layer Traffic Optimization (alto) [EB/OL]. https://datatracker.ietf.org/wg/alto/charter/.

[9] Ng T S E, Chu Y H, Rao S G, et al. Measurement-based Optimization Techniques for Bandwidth-demanding Peer-to-peer Systems [C]// IEEE INFOCOM'03, 2003.

[10] Karagiannis T, Rodriguez P, Papagiannaki D. Should Internet Service Providers Fear Peer-assisted Content Distribution [C]// Internet Measurement Conference (IMC), 2005.