

视频点播中基于用户行为的自适应资源搜索机制*

郭红方^{1,2}, 林子松^{1,2}, 王宗敏²

(1. 郑州大学 信息工程学院, 郑州 450001; 2. 郑州大学 河南省信息网络重点学科开放实验室, 郑州 450052)

摘 要: 根据用户访问行为特征, 提出一种新的自适应数据搜索结构 U-VoD, 该机制根据用户的自主访问行为自适应地组成行为相似团体, 不需要历史访问记录就能实时地反映用户的访问模式与数据块的流行度。同时, 基于搜索结构所学习的信息, 设计一种有效的数据流行度感知的预存策略。大量的实验结果表明, U-VoD 在频繁的用户交互情景下, 极大地降低了用户 VCR 操作引起的跳转时延, 并减轻了对服务器的访问压力。

关键词: 视频点播; 用户访问模式; 自适应; P2P

中图分类号: TP393

文献标志码: A

文章编号: 1001-3695(2012)04-1504-05

doi:10.3969/j.issn.1001-3695.2012.04.084

User-behavior based self-adaptive indexing for VoD service

GUO Hong-fang^{1,2}, LIN Yu-song^{1,2}, WANG Zong-min²

(1. School of Information & Engineering, Zhengzhou University, Zhengzhou 450001, China; 2. Henan Provincial Key Laboratory on Information Networking, Zhengzhou University, Zhengzhou 450052, China)

Abstract: This paper proposed a novel adaptive indexing structure which was based on user's arbitrary behavior, called U-VoD. U-VoD group users according to user's request pattern. It could reflect user's request pattern and chunk popularity without any history statistics. It used the knowledge extracted from U-VoD to study an efficient popularity-aware prefetching mechanism. The simulation results demonstrate that it greatly reduces start-up delay with low server load under frequent seeking operations, especially for non-uniform access pattern.

Key words: VoD(video on demand); user access pattern; self-adaptive; P2P(peer-to-peer)

近年来 P2P 技术在文件共享、视频直播应用上的成功也为其在 VoD 应用研究开拓了新的发展空间。与直播应用用户在同一时间观看同一个视频内容不同, P2P VoD 用户请求的时间的不同步性、兴趣独立性以及实时的动态访问模式, 使得 VoD 的发展面临着更多挑战, 尤其是如何有效地支持用户对视频内容的自主交互操作^[1]。

据统计^[2], 用户跳转操作很频繁, 这从 PPVA(PPLive video accelerator) 监测收集到的数据得到了验证, 用户 48% 以上的操作来自更换当前播放内容^[3]。点播服务使用户有了更强的主动性与选择性, 实现了按需播放音视频节目。然而这种特点却不利于 P2P 技术优势的发挥, 因为频繁地更改请求的数据内容, 使得用户之间的互相合作、共享数据更难以控制。节点的行为完全自主, 极大地增加了整个网络的不稳定性。当前已有的解决方案侧重讨论了如何支持点播应用中的随意跳转行为, 并假设节点对数据块的访问模式服从均匀分布。而实际上, 对于单个视频内容, 用户对每个数据块的请求频率是不同的, 用户可以跳过不感兴趣的部分, 只关注一些热门片段。最近的一些研究表明, 用户的请求模式在某些情况下服从 log-normal 分布^[4], 这种极不均匀的请求模式使问题变得更加复杂, 这点从 PPVA 的统计数据也得到了证实(图 1)。

对于用户, 其关注的是低跳转时延及流畅的播放体验。如

图 2 所示, 理想状态下, 当用户从 t_1 跳转到 t_2 , 在不产生时延的情况下就能达到播放码率, 当跳到 t_3 时, 由于所请求的数据还没获取, 不能马上播放, 此时就产生了跳转时延, 本文称为启动时延, 影响了用户观看的流畅度。如何在这样一个高度动态的网络结构中有效地找到请求的数据内容而又不影响用户观看体验就成为了 P2P VoD 设计的一个关键因素。针对 VoD 用户请求服务与提供服务的高度动态性特点, 本文提出一个新的数据搜索机制 U-VoD。

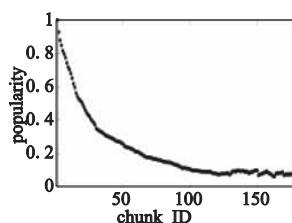


图1 PPVA数据块的流行度分布情况

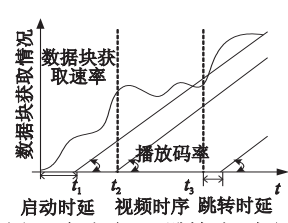


图2 启动时延、跳转时延与用户的观看体验关系示意图

1 相关工作

P2P 在视频直播上的研究与应用已经取得很大的成功^[5]。鉴于此, 近年来越来越多的研究人员与商业公司开始将注意力转向 P2P VoD, 以解决点播应用中服务器开销过大、扩展性差的问题。由于引入 P2P 技术, 使得用户的内容搜索问题变得

收稿日期: 2011-07-15; 修回日期: 2011-08-19 基金项目: 国家“863”计划资助项目(2008AA01A315); 河南省信息化工程项目(2008xxh001); 河南省教育厅 2011 年创新人才项目(2011HASTII003)

作者简介: 郭红方(1981-), 女, 河南濮阳人, 博士研究生, 主要研究方向为网络多媒体、云服务等(hfguo@zzu.edu.cn); 林子松(1973-), 男, 副教授, 硕导, 博士, 主要研究方向为计算机网络、Web 服务; 王宗敏(1964-), 男, 教授, 博导, 博士, 主要研究方向为计算机应用技术。

更加至关重要,据此提出了一系列的解决方案,主要分为两大类。其一是通过构建特有的覆盖网络结构,文献[6]给出一个P2P与CDN混合的流媒体分发模型,减少了由于数据搜索所产生的时延。RING^[7]、VMesh^[8]等结构都有效地支持点播应用,然而TAG对整个结构的维护成本过高,对于RING与VMesh的健壮性和有效性也存在质疑。InstantLeap^[9]的搜索结构负载少,利用其定义的快捷邻居以 $O(1)$ 的代价找到请求的内容,尽管如此,由于未考虑到极度非均匀的用户请求模式,在实际应用中往往达不到理论最优值。Peer-assisted VoD^[10]提出一个以服务器为中心、节点辅助的VoD设计,有效地支持了用户交互操作,却给服务器带来了较重的负载。其二是通过节点缓存管理,对数据块进行预存及对数据备份在整个系统中的分布进行优化等^[2,11,12]。通过预存将来很可能会请求的数据块,极大地提高了由于用户频繁跳转所引起的跳转时延。数据备份策略确保了数据在整个系统中的分布模式,增强了数据块的多样性,提高了节点间协作几率。然而两者在预测不准的情景下,都会造成不必要的带宽消耗与存储空间的浪费。为了提高其准确率,目前的解决方案大多采用主动地收集系统全局信息,这随之带来大量的控制信息,并且收集P2P系统信息本身就是一个非常难解决的问题。文献[13]考虑到了点播用户搜索操作行为特征与媒体数据块受欢迎程度,提出了有效的预取机制。本文侧重点与其不同,本文是根据用户自主行为自适应地构建搜索结构,并根据搜索结构来设计预存策略。根据现有解决方案的不足与最近一些对实际系统统计数据进行分析,本文提出的U-VoD在不需要收集系统信息的情景下,给用户提供了低启动时延与流畅的播放体验。

2 模型分析

本章首先构建P2P VoD内容搜索问题的理论模型,进而对其进行分析并探究其影响搜索性能的因素。

假设系统只有一个源服务器,有 N 个节点同时收看同一个视频,每个节点可自主选择播放进度,其内容可从多个邻居节点获取。若此时系统中无其他节点存储过此数据块,则该节点需向源服务器发出请求。每个节点都维护一个有限的缓冲区。

用户操作的过程以时间间隔表示,视频按时序分为等长的 C 个数据块,节点可以请求其中任何数据块。每个数据块的热度定义为 $P = (p_1(t), p_2(t), p_3(t), \dots, p_c(t))$,其中 $p_i(t)$ 表示在 t 时刻节点请求 i^{th} 数据块的概率,则 $\sum_{i=1}^{i=C} p_i(t) = 1$ 。因此 i^{th} 数据块在 T 时段内的平均被请求概率可以表示为

$$p_i = \frac{p_i(0) + p_i(1) + \dots + p_i(T)}{T}$$

在P2P VoD中,数据块分布在每个节点有限的缓冲区中,定义 $L = (l_1(t), l_2(t), l_3(t), \dots, l_m(t))$ 代表节点缓存的状态,其中 $l_i(t)$ 代表在 t 时刻存有数据块 i 的节点数目。由于缓存大小是有限的,当新的数据块到来时,必须通过一定的替换算法进行缓存管理,因此 $N \times p_i(t) \geq l_i(t)$ 。当节点有跳转操作时,定义 $h_{ij}(t)$ 为节点 i 请求数据块 j 的跳转时延。进而,内容搜索优化问题可以表示成以下形式:

$$\text{Minimize } \frac{\sum_{i=0}^T \sum_{j=1}^C \sum_{i=1}^{N \times p_j(t)} h_{ij}(t)}{N \times T} = \sum_{j=1}^C p_j \times h_j \quad (1)$$

其中: $\sum_{j=1}^C p_j = 1, \forall i \in N; h_j \geq 0, j = 1, 2, \dots, N; t = 1, 2, \dots, T$ 。

式(1)中的 p_j 是整个过程中节点请求 j^{th} 数据块的概率, h_j 是获取请求数据块所需的平均时延。在已有的理论分析中, p_j 一般是作为已知量来应用的,然而用户的行为是不可预知的、自主的, p_j 并不能事先得知。直观认为, h_j 与系统中数据块备份数有关,即 $h_j \propto 1/l_j$ 。但这并不符合实际VoD系统中存在大量搭顺风车(free-riders)的节点,它们不给其他节点提供服务。所以数据块备份的多少并不能反映 h_j 的性能。另一种方案是设计专门服务器,以得到节点的历史数据,分析得出 p_j ,进而根据其值决定主动分布该数据块备份的多少。然而 p_j 值是多变的,节点很可能在下一秒就改变了兴趣。虽然确保了上一秒受欢迎的数据在系统中大量存在,并不能有效地提高用户的观看体验,反而增加了存储空间的浪费。所以在没有历史系统信息的情况下,如何快速找到所需的数据块的关键因素是实时用户的行为模式。

很明显, p_i 值对用户访问行为很敏感,这也给内容搜索效率问题的解决提供了一个机遇,解决方案需要满足以下两个条件:a) p_i 能实时准确地反映用户的访问行为;b) p_i 值高的数据块能更容易、更快捷地被访问到。假设 p_j 已经由a)得知,为得到式(1)的最优值,可根据贪婪算法, p_j 越大,得到 h_j 越小。这种方案是最优的,在此需要证明此方法满足两个属性:

a) greedy-choice。定义Cost是最小的代价, a^{th} 数据块是最受欢迎的。假设另一数据块 b 也是最受欢迎的,已知 $h_a < h_b$, $h'_b = h_a$ 。

$$\begin{aligned} \text{Cost} - \text{Cost}' &= \sum_{j=1}^C p_j \times h_j - \sum_{j=1}^C p_j \times h'_j = \\ p_a \times h_a + p_b \times h_b - p_a \times h'_a - p_b \times h'_b &= \\ (p_a - p_b)(h_a - h_b) &\geq 0 \end{aligned}$$

由此得 $\text{Cost} \geq \text{Cost}'$,又因Cost是最优的,所以 $\text{Cost} = \text{Cost}'$ 。

b) optimal-substructure。假设Cost是最优搜索代价, a 是最受欢迎数据块, $h_a = 1$; Cost'代表一个不包括 a 的最优搜索代价,可得

$$\text{Cost} = \sum_{j=1, j \neq a}^C p_j \times h_j + p_a \times 1 = \text{Cost}' + p_a$$

如果Cost'不是除了 a 的最优搜索代价,则存在 $\text{Cost}'' < \text{Cost}'$,进而得到 $\text{Cost} > \text{Cost}'' + p_a$,这与之前假设Cost是最优的冲突,因此可证substructure是最优。

3 U-VoD 设计原理

3.1 伸展树

扩展树是一个自适应的二叉搜索树^[14],除了有搜索树的搜索功能外,还有一个重要的特点是使被访问过的元素在下次被访问时更容易,访问模式可影响树的结构。当访问某个元素时,扩展树通过一系列的操作移动此节点成为根节点,这样有利于该元素下次被请求。

伸展树的重构是通过伸展操作来实现,重复以下步骤直到 x 成为树根为止,如图3所示。

Zig:如果 x 的父节点 y 是树根,则旋转连接 x 和 y 的边(这种情况是最后一步)。

Zig-Zig: 如果 y 不是树根, 而且 x 和 y 本身都是左孩子或者都是右孩子, 则先旋转连接 y 和 x 的祖父节点 z 的边, 然后再旋转连接 x 和 y 的边。

Zig-Zag: 如果 y 不是树根, 且 x 是左孩子, y 是右孩子 (或相反), 则先旋转连接 x 和 y 的边, 再旋转连接 x 和新的 y 的边。

在节点 x 处进行 splay 操作的时间是与查找 x 所需的时间成比例的。Splay 操作不单是把 x 搬移到了树根, 而且还把查找路径上的每个节点的深度都大致减掉了一半。

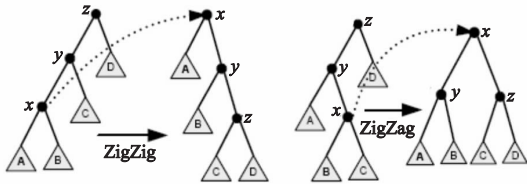


图3 伸展树的基本操作

3.2 基于用户行为的自适应搜索机制

点播用户的兴趣可能会随着时间而变化, 在某一时间段对某些数据块极其喜欢, 而在另一时间段喜欢另外的, 这样引起了全局上的数据访问模式是均匀分布, 而在某一时间段是极不均匀分布。在这种情景下, U-VoD 采用根据某时间段内数据请求达到所有子问题的最优, 以达到全局最优, 这可以归结为一个简单的动态规划问题。为了达到子问题的最优, 搜索结构根据用户的请求实时地重构, 以反映用户行为模式。

本文采用扩展树的机制来实现内容搜索有以下原因: a) 在点播应用中, 用户对数据块的访问概率是不均匀的, 扩展树能有效支持不均匀请求的内容搜索, 确保了高访问概率的数据块能被快速地访问; b) 点播应用要求整个搜索网络拓扑能与用户请求行为相适应, 这样计算数据块的受欢迎程度与一些额外的控制信息就不再是影响点播服务的因素; c) 容易实现与维护。

1) 伸展树与点播应用之间的映射 伸展树之所以优于别的搜索结构, 最重要的因素就是它的伸展操作, 一系列的旋转操作可以把访问的节点移动到根节点, 每个节点的值可以是任何可排序的属性值, 如大小、权值等。如图 4 所示, U-VoD 利用正在播放的视频数据块 ID 作为值来组织构建整个搜索拓扑。每个搜索节点维护一个存有 i^{th} 数据块的所有节点信息 l_i , 这些节点都属于一个逻辑的搜索节点。

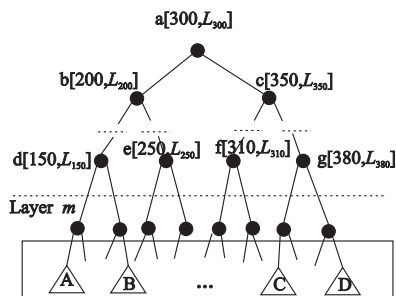


图4 U-VoD 自适应搜索网络拓扑

2) 改进后的伸展树 尽管伸展树特有的自优化功能对于频繁交互的点播应用非常具有吸引力, 然而将它应用于该服务并不是那么直接。首先, 受欢迎程度比较低的数据块, 请求它的节点也相应地少。然而根据原伸展树的设计, 一旦它被访问, 该搜索节点就会被移至整个搜索树的树根, 由于该数据块受欢迎程度低, 在节点多次访问操作后, 代表此数据块的搜索

节点又会被移至树的子节点甚至叶子节点。

另外, 对于一些受欢迎程度相近的数据块, 节点对它们的访问频率是相同的, 若每次访问都引发一个伸展操作, 则会加重网络动态性, 产生额外的控制信息, 最终无益于系统性能。如果所有数据块的受访问概率都相近, 这就属于均匀的访问模式了。如图 3 的 zigzig 操作, 假设请求节点 z 的数目是 n_1 , 请求 x 的数目是 n_2 , 根据扩展树的特点, $n_1 > n_2$ 。当不进行 zigzig 操作时, 访问 x 与 z 的时延可以由从树根节点到目的节点所经历的总跳数来表示:

不旋转时: $n_1 \times 1 + 3 \times n_2 = n_1 + 3n_2$ 。

含旋转时: $3 + (n_2 - 1) \times 1 + (n_2 - 1) \times 3 + (n_1 - n_2 + 1) \times 1 = n_1 + 3n_2$ 。

不难看出, 当访问 x 时, 没必要把它移至树根节点, 因为两种情况下搜索效率是一样的。这个过程不但没有提高搜索效率, 反而增加了由于移动节点操作而产生的控制信息。

为了解决以上问题, U-VoD 提出了两个策略以便扩展树能更好地适应高交互性的视频应用。其一, 当用户访问某一数据块时, 不会直接把此数据块对应的搜索节点移到树根, 旋转操作次数限制为 num_s , $\text{num_s} > 1$; 每进行一次旋转操作, 搜索节点就会以 $h \in \{1, 2\}$ 的跳数靠近树根。受欢迎的节点在进行大概 $\lceil \log C / (\text{num_s} \times h) \rceil$ 次访问后, 还是会被推向离树根近的位置, 以便下一次更容易地被访问。根据用户的实时请求, 此机制自适应地区分了受欢迎的数据块与不受欢迎的数据块。

另一方面, 对于用户访问概率相当的数据块, 在搜索结构中, 其应用的搜索节点必定位置相近。如图 4 所示, m 层以上的搜索节点所代表的数据块被访问的概率相近, 所以本文规定了 m 层以上的搜索节点作为逻辑的搜索节点, 当有用户访问 m 层以上的各成员节点时, 搜索结构不随用户的访问而重构。这样不但充分利用了自组织的特性, 而且极大地降低了控制信息的开销。

算法 1 给出了点播用户交互行为的运作过程。

算法 1

```

1 if 新节点  $p$  请求加入 then
2   节点请求数据块  $i$ 
3   Peerjoin( $p, i$ ); end if
4 if 节点跳转到数据块  $j$  then
5   Peerjump( $j$ ); end if
6 if 节点离开 then
7   Peerleave( $p$ ); end if
8 function Peerjoin(peer  $p$ , chunk  $i$ )
9 {
10  找到 U-VoD 搜索结构的树根 root;
11  if 数据块  $i$  是首次被请求; then
12     $p$  向服务器请求数据, 同时作为数据块  $i$  的源加入搜索网络
13  else
14    找到目标搜索节点维持的  $L$  列表, 节点向  $L$  列表里的节点请求数据块  $i$ , 同时将这些节点加入  $L$  列表
15  }
16 function Peerjump(chunk  $i$ )
17 {
18  if 本地缓存有数据块  $i$  then
19    直接播放
20  else if 邻居节点有  $i$  then
21    向邻居节点请求
22  else Peerjoin(this,  $i$ );
23 }
24 function Peerleave(peer  $p$ )

```

```

25 {
26 if  $p$  主动离开 then
27   通知邻居节点,其搜索节点将其从维持的  $L$  表中删除(不会引起网络重构)
28 if 此节点为  $L$  列表中最后一个节点 then
29   从列表中删除,并删除相应搜索节点(网络重构)

```

4 U-VoD 搜索结构辅助的预取策略

有效的预取策略能极大地提高用户的观看体验,并同时减少网络资源的消耗。然而,若预存的数据块不准确,即不是节点以后要访问的数据块,则不仅不能满足用户的需求,同时还会增加网络资源的浪费。现有的解决方案大多利用服务器上有关用户访问方式的信息来确保用户预取数据块的准确性。然而服务器如何收集并准确地分析用户的访问请求本身就是一个很复杂的问题。

本文充分利用了自适应 U-VoD 搜索结构,提出了一个简单有效的预取策略。如图 5 所示,从搜索结构中能很直观地得到用户对数据请求的情况,在没有数据间关系及用户历史请求记录的情况下,得到实时的数据块被喜爱程度。从下到上,数据块的访问频率是升高的,位于树根的数据块是最受欢迎的,越受欢迎的数据将会被用户访问的概率越大。根据从搜索结构上得到的访问频率信息,本文引入预取优先级的参数,其值取决于访问频率。节点的访问频率高,则其预取优先级越高,可见,位于树根的数据块,其预取优先级最高。优先级的值用节点所在结构中的层数具体表示,如某节点正在播放数据块 C_3 ,在带宽冗余的情况下,可预存该节点上层的数据块(如 C_1 、 C_2)或同层的数据块(如 C_4 、 C_5)。若节点还没请求过 C_1 、 C_2 、 C_4 、 C_5 ,则预存 C_1 、 C_2 的优先级大于 C_4 、 C_5 ;若此节点已经请求过 C_1 数据,则其预存 C_1 的优先级就设置为 0。

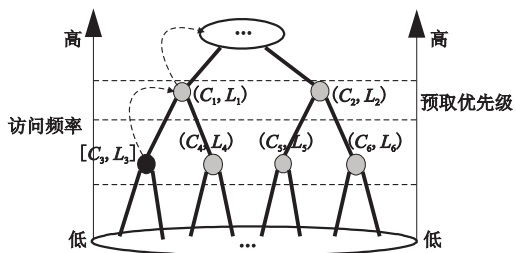


图5 基于搜索结构的预取策略

5 性能测试与分析

5.1 实验环境与配置

本文采用 C++ 编程语言实现了一个基于事件驱动的视频点播应用模拟器。初始阶段,服务器是唯一视频数据源。本实验中的事件主要包括节点加入、离开、跳转,其中跳转操作是本文主要的关注点。在实验中,测试的数据流是一个 120 min、500 kbps 播放码率的视频,每个数据块大小是 10 s,这也是传输的最小单位,整个视频流共包括 720 个数据块。初始情况下,每个节点的请求可以是 1~720 的任意一数据块。不均匀的用户请求分布遵循 log-normal 分布^[4],其中 $\sigma = 1.35$, $\mu = 0.0159$ 。

根据最近对现有实际点播系统的用户行为分析,主要从以下三个场景对 U-VoD 进行了一系列的模拟测试:a)用户均匀

请求模式下跳转时延;b)用户非均匀请求模式下跳转时延;c)U-VoD 辅助的预存策略下用户的满意度情况。本文所提到的时延为搜索目标数据块所经历的跳数。

5.2 实验结果与分析

a)用户请求均匀分布情况下,随机地触发 48% (基于 PP-VA 的统计数据)的自主用户执行跳转操作,InstantLeap 和 RINDY 高效的搜索性能促使本文以它们作为比较对象,同时也比较了 AVL 树。如图 6 所示,随着用户数据量的增加,U-VoD、RINDY 的跳转时延均保持在 5 跳以内,扩展性强;AVL 对用户数量的大小很敏感,随着用户数据量的增加,其时延也不断增加;尽管 InstantLeap 不随用户的增加而变化,但此时它的跳转时延均高于 U-VoD、RINDY,这是因为根据其工作原理,它需要维持 720 个节点组,为了找到目标数据开销变大了。

b)用户请求非均匀分布情况下,一些热门视频片段总是更能吸引用户的眼球,这种情况使得在某段时间内,数据块被访问频率严重地不均匀。

在用户请求服从 log-normal 分布时,图 7 给出了 U-VoD 在支持用户连续 120 次请求跳转时延的分布情况。不难看出,绝大多数的请求可以在 1 跳内完成,也有个别需要经过多跳,这主要是因为有些数据块是第一次被访问或者是不受欢迎的。从整个请求过程来看,越靠前的请求,其跳转时延越长;随着请求的不断增长,长的跳转时延数量明显减少,这也反映了 U-VoD 的特性,根据用户的请求逐渐地重构搜索网络,使具有高访问率的数据块位于下次易被访问的位置。

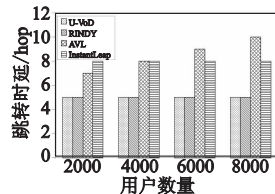


图6 用户请求服从均匀模式下各策略跳转时延的比较

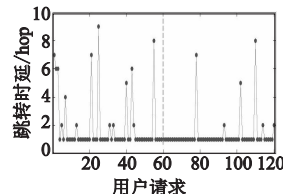


图7 用户请求服从非均匀模式下整个工作过程

图 8 描述了用户分布服从 log-normal 分布情况下,U-VoD 与其他策略在时延方面的表现。与在均匀用户请求下相同,AVL 随着用户的不断增加,其跳转时延不断增加,同样 RINDY 也有时延增加的趋势。然而对于 InstantLeap,其性能不随用户数量的增加而降低,这与 InstantLeap 中无论用户请求分布模式怎样,都与随机选择快捷邻居节点有关,其搜索目标数据块的时延一样。U-VoD 用户越多,请求也越多,跳转时延有降低的趋势,当用户达到 12 000 时,平均时延达到 1。

为了更深入地了解 U-VoD 支持非均匀用户请求的情况,图 9 对不同用户请求模式进行了比较,其中 zipf 分布参数值设置为 0.57^[15]。从图 9 可以看到,U-VoD 在处理非均匀分布用户请求时表现出了其优越性。

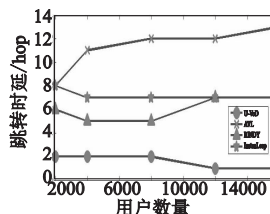


图8 用户请求服从非均匀模式下各策略跳转时延性能比较

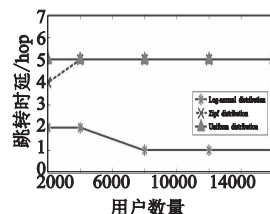


图9 用户请求不均匀程度对跳转时延的影响

c)U-VoD 辅助的预存策略。为了证明所提出的预存策略

的高效性,本文选择了文献[9]介绍的随机预取策略与文献[13]中提出的 Client-90-10-Two-Ranges(CTR)策略作为比较对象。对于随机预取策略,节点会随机预取本地未存储过的数据块;而采用 CTR 策略的节点以 90% 的概率预取当前播放数据块后的第一块数据,以 10% 的概率预取其后的第二块数据。

定义命中率为从本地缓存获取数据的请求数与总的请求数的比值。从图 10 可知,在两种不同的用户请求模式下,U-VoD 的命中率均高于另外两种预存策略。随机预存策略在两种场景下其平均命中率一样。U-VoD 在请求均匀的情况下表现不如 CTR 和随机策略,因此此情景下搜索结构并不能充分体现其优越性,预取在该节点上层的数据块,缩小的预取范围没有随机策略灵活。但在非均匀分布情况下,其性能远远高于 CTR 和随机策略,相对 CTR 8.2% 的命中率,U-VoD 能使 60% 的用户请求通过本地缓存获取所需数据。

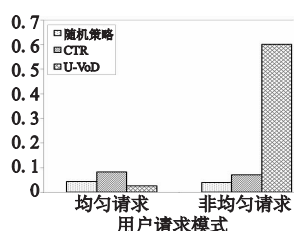


图10 U-VoD辅助的预存策略性能

6 结束语

基于点播应用的模型,分析在频繁交互操作下如何得到资源搜索的最优策略,并根据此分析,提出一个基于用户实时行为构建的搜索网络 U-VoD,用户的请求操作参与搜索网络的重构,并形成行为相似团体。在没有用户间关系及历史行为记录信息下,实时反映用户的访问行为,极大地提高了搜索效率。同时,充分利用对 U-VoD 搜索网络分析得到的信息,提出了简单有效的预存策略,极大地提高了用户观看视频的流畅度。大量的实验结果充分验证了 U-VoD 在支持用户频繁交互操作时的高效性,尤其是在用户的请求模式不可预测、服从非均匀分布时,极大降低了用户 VCR 操作产生的跳转时延,提高了用户的满意度。

参考文献:

- [1] 沈时军,李三立. 基于 P2P 的视频点播系统综述[J]. 计算机学

报,2010,33(4):613-624.

- [2] ZHENG C, SHEN G, LI S. Distributed prefetching scheme for random seek support in peer-to-peer streaming applications[C]//Proc of ACM Multimedia Conference, Workshop on Advances in Peer-to-Peer Multimedia Streaming. 2005:29-38.
- [3] PPLive video accelerator[EB/OL]. <http://www.ppacc.com/en/index.html>.
- [4] BRAMPTON A, MacQUIRE A, RAI I A, et al. Characterizing user interactivity for sports video-on-demand[C]//Proc of ACM NOSS-DAV. 2007.
- [5] LIU J, RAO S G, LI B, et al. Opportunities and challenges of peer-to-peer Internet video broadcast[J]. IEEE Special Issue on Recent Advances in Distributed Multimedia Communications, 2008,96(1):11-24.
- [6] 臧运港,陈光喜. P2P 和 CDN 的混合流媒体分发模型及分析[J], 计算机应用研究,2010,27(6):2208-2213.
- [7] CHENG B, JIN H, LIAO X. Supporting VCR functions in P2P VoD services using ring-assisted overlays[C]//Proc of IEEE ICC. 2007:1698-1703.
- [8] YIU W P K, JIN X, CHAN S H G. VMesh: distributed segment storage for peer-to-peer interactive video streaming[J]. IEEE Journal on Selected Areas in Communications,2007,25(9):1717-1731.
- [9] QIU X, WU C, LIN X, et al. InstantLeap: fast neighbor discovery in P2P VoD streaming[C]//Proc of ACM NOSSDAV. 2009:19-24.
- [10] HUANG C, LI J, ROSS K W. Peer-Assisted VoD: making Internet video distribution cheap[C]//Proc of IPTPS. 2007:1-6.
- [11] VRATONJIC N, GUPTA P, KNEŽEVIC N, et al. Enabling DVD-like features in P2P video-on-demand systems[C]//Proc of ACM Workshop on Peer-to-Peer Streaming and IP-TV. New York: ACM, 2007:329-334.
- [12] ANNAPUREDDY S, GKANTSIDIS C, RODRIGUEZ P, et al. Providing video-on-demand using peer-to-peer networks[C]//Proc of the 16th World Wide Web Conference. 2007.
- [13] 王娟,纪其进,朱艳琴. 基于用户行为特征的 P2P 视频点播系统数据预取机制[J]. 小型微型计算机系统,2010,31(10):2049-2053.
- [14] SLEATOR D D, TARJAN R E. Self-adjusting binary search trees[J]. Journal of the ACM,1985,32(3):652-686.
- [15] CHENG Xu, LIU Jiang-chuan. NetTube: exploring social networks for peer-to-peer short video sharing[C]//Proc of IEEE INFOCOM. 2009:1152-1160.

(上接第 1472 页)了网络能量,并延长了网络寿命。

参考文献:

- [1] 马祖长,孙怡宁,梅涛. 无线传感器网络综述[J]. 通信学报,2010,25(4):114-124.
- [2] 孙利民,李建中,陈渝,等. 无线传感器网络[M]. 北京:清华大学出版社,2009.
- [3] 任丰原,黄海宁,林闯. 无线传感器网络[J]. 软件学报,2011,14(7):1282-1291.
- [4] 崔莉,鞠海玲,苗勇,等. 无线传感器网络研究进展[J]. 计算机研究与发展,2009,42(1):163-174.
- [5] QI Hai-rong, IYENGAR S S, CHAKRABARTY K. Multiresolution data integration using mobile agents in distributed sensor networks[J]. IEEE Trans on Systems, Man, and Cybernetics-Part C:

Applications and Reviews,2011,31(3):383-291.

- [6] 郑巍,刘三阳,寇晓丽. 动态传感器网络移动代理路由算法[J]. 控制与决策,2010,25(7):1035-1039.
- [7] 胡建理,周斌,吴泉源,等. ICA:一种基于混合智能算法的移动 agent 路由算法[J]. 小型微型计算机系统,2010,31(2):348-354.
- [8] 徐云剑,彭沛夫,郭艾寅,等. 基于改进蚁群算法的 WSN 移动代理路由算法研究[J]. 计算机工程与应用,2009,45(4):126-130.
- [9] 姚永杰,席庆彪,刘慧霞. 基于改进遗传蚁群算法的无人机航路规划[J]. 计算机仿真,2011,28(6):44-48.
- [10] 方旺盛,黎飞龙. WSN 中基于改进自适应遗传算法的移动代理路由算法[J]. 计算机与数字工程,2010,38(12):4-9.
- [11] PEOTTA L, VANDERGHEYNST P. Matching pursuit with block incoherent dictionary[J]. IEEE Trans on Signal Processing, 2007,55(9):4549-4557.