

# 一种基于质量命中率的流媒体代理缓存布局方法<sup>\*</sup>)

王泰 杜旭 程文青 杨宗凯

(华中科技大学电子与信息工程系 武汉 430074)

**摘要** 在网络边缘对流媒体的热点片段进行缓存,是流媒体分发研究领域的一个热点。异构环境下不同种类的接入用户对流媒体的播放质量具有不同的要求。为了使代理缓存最大限度地满足全体用户对收看质量的要求,本文首先提出了质量命中率的定义来衡量代理缓存对提高用户观看质量的贡献,然后提出了一种基于质量命中率的流媒体代理缓存布局方法。仿真结果表明,该布局方法不仅能使全体用户的平均质量命中率最大,而且有效地利用了代理缓存的磁盘空间,显著降低了片段的播放时延。

**关键词** 流媒体,代理缓存,质量命中率

## A Streaming Media Proxy Cache Layout Method Based on Quality Hit Ratio

WANG Tai DU Xu CHENG Wen-Qing YANG Zong-Kai

(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074)

**Abstract** Caching video objects at the edge of the backbone network has become a hot spot in the video delivery research. Clients with different access bandwidths request different kinds of perceptual quality in the heterogeneous environment. This paper proposes a streaming media proxy cache layout method to make the proxy try its best to provide the maximum overall quality hit ratio(QHR). Simulation results demonstrate that our method not only achieves the highest QHR but also reduces the segment resume delay, taking full advantage of the proxy storage.

**Keywords** Streaming media, Proxy caching, Quality hit ratio

## 1 引言

近年来,为了缓解流媒体业务的急速增长给骨干网络资源带来的压力,在网络边缘部署代理缓存,成为流媒体分发领域的一个研究热点。由于流媒体代理缓存在靠近接入用户的地方存储了大多数用户关心的热点对象,使得对这些热点对象的请求能够就近得到响应,不仅减轻了骨干网络和服务器器的负载,还缩短了用户请求的响应时延。

一部流媒体影片的体积通常比较大,如果缓存整部影片,那么几部影片就会将代理缓存十分有限的存储空间消耗殆尽。这样一种布局方式无法容纳更多的流媒体内容,将会导致相当低的比特命中率和磁盘空间利用率。考虑到一部影片最流行的内容一般会集中在若干特定的位置<sup>[1]</sup>,许多学者提出了不同的缓存空间布局方法<sup>[2~6]</sup>。这些方法都采用分段存储的思路来捕捉更多的流行度高的流媒体内容,不同点在于如何分段和优先存储哪一段。但是它们并没有考虑到在异构环境下具有不同接入带宽的用户对流媒体内容播放质量的不同要求。

文[7~9]结合分层编码或转码来解决这一问题,但是仍使用字节命中率来衡量代理缓存的性能。字节命中率主要反映代理缓存在节省骨干网络带宽中所起的作用。相比而言,在异构环境下人们更希望知道具有不同接入带宽的用户对于流媒体内容播放质量的平均满意程度。文[10]对用户观看流媒体时的主观满意程度进行了调研,发现传统的峰值信噪比并不适合作为衡量标准。

本文的主要工作包括:(1)提出用质量命中率来表示流媒

体代理缓存在满足用户观看质量方面的贡献;(2)提出了一种使所有用户的平均质量命中率最大化的流媒体代理缓存布局方法。这种基于质量命中率的流媒体代理缓存布局方法不仅考虑了流媒体内容流行度的分布,而且考虑了异构环境下的不同接入带宽的用户对观看质量的要求。

后文的组织结构是:第2节简要介绍了本文所研究的场景和问题的形成,第3节提出了基于质量命中率的流媒体代理缓存布局方法,第4节对该布局方法进行了仿真,并与其他方法进行了比较。最后小结全文。

## 2 问题的提出

一个典型的视频流媒体内容分发系统的拓扑结构通常由一个源服务器、部署在网络边缘的代理缓存和位于其后的用户组成,如图1所示。图1中的甲、乙、丙用于区分异构环境中具有不同接入带宽的三类用户。

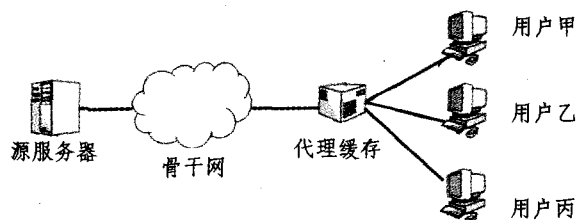


图1 一个典型的视频流媒体内容分发系统的拓扑结构

一般情况下,源服务器存储了具有最高播放质量的所有流媒体内容。代理缓存则总是试图尽可能多地存储具有较高

<sup>\*</sup>湖北省自然科学基金(2005ABA264),国家自然科学基金(No. 60302004)。王泰 博士研究生,研究方向为多媒体分发技术;杜旭 副教授,研究方向为多媒体网络通信;程文青 教授,研究方向为多媒体网络通信;杨宗凯 博士生导师,教授,研究方向为多媒体网络通信。

流行度的流媒体片段,以使得大多数用户的请求能够直接从代理缓存获得响应,而不必增加源服务器和骨干网的工作负荷。由于代理缓存的存储空间通常十分有限,对于那些没有命中的请求,代理缓存负责将其转发给源服务器,由源服务器将代理缓存未能存储的那一部分内容通过代理缓存传送给用户。

在异构环境中,由于不同种类的接入用户对流媒体的播放质量具有不同的要求,因此某段流媒体内容需要由时间长度和播放质量这两个属性来描述。不论是时间长度还是播放质量,只要没有满足用户的请求,都需要由源服务器来补足缺额。

记第  $k$  个影片的第  $j$  个片段为  $Seg(k, j)$ , 它的时间长度为  $l(k, j)$  秒。在源服务器中,它能提供的最高传输速率为  $R_{original}(k, j)$  bps。在代理缓存中,它能提供的最大传输速率为  $R_{ached}(k, j)$  bps。如果  $Seg(k, j)$  不在代理缓存中,相应的  $R_{ached}(k, j)$  为 0。

假设异构环境中有  $N$  种不同的接入带宽,由小到大记为  $C(1), C(2), \dots, C(i), \dots, C(N)$ 。据此可以将用户分为  $N$  类。当第  $i$  类用户请求  $Seg(k, j)$  时,设这类用户能够请求的速率为  $R_{request}(i, k, j)$  bps。由于用户的请求受其接入带宽的限制,当  $R_{original}(k, j)$  超过  $C(i)$  时,第  $i$  类用户能够请求的速率只能达到  $C(i)$ , 即(1)式。

$$R_{request}(i, k, j) = \min(C(i), R_{original}(k, j)) \quad (1)$$

类似地,当代理缓存收到来自第  $i$  类用户的请求后,在缓存片段所能提供的最大传输速率  $R_{ached}(k, j)$  和用户接入带宽  $C(i)$  的限制下,代理缓存能向第  $i$  类用户实际传送的可用速率  $R_{available}(i, k, j)$  bps 由(2)式给出。

$$R_{available}(i, k, j) = \min(R_{ached}(k, j), C(i)) \quad (2)$$

由(1)和(2)式我们给出质量命中率的定义。

**定义** 当代理缓存响应第  $i$  类用户对  $Seg(k, j)$  的请求时,代理缓存的质量命中率  $QHR(i, k, j)$  为它向第  $i$  类用户实际传送的可用速率  $R_{available}(i, k, j)$  与第  $i$  类用户能够请求的速率  $R_{request}(i, k, j)$  的比值,如(3)式所示。

$$QHR(i, k, j) = \frac{\min(R_{ached}(k, j), C(i))}{\min(C(i), R_{original}(k, j))} \quad (3)$$

根据上述定义,我们可以推出当代理缓存响应全体用户对所有片段请求时的平均质量命中率。

设  $Seg(k, j)$  的流行度为  $Pop(k, j)$ , 所有片段的总流行度为  $T_{pop}$ 。源服务器上共有  $T_{dip}$  部流媒体影片,第  $k$  个影片被分成  $S(k)$  个片段。第  $i$  类用户的数目占全体用户总数的比例为  $P(i)$ 。那么,所有用户请求所有内容片段时,代理缓存的平均质量命中率  $\overline{QHR}$  如(4)式所示。

$$\overline{QHR} = \frac{1}{T_{pop}} \sum_{k=1}^{T_{dip}} \sum_{j=1}^{S(k)} QHR(i, k, j) Pop(k, j) P(i) \quad (4)$$

将(1)、(2)、(3)式代入(4)式,得

$$\overline{QHR} = \frac{1}{T_{pop}} \sum_{k=1}^{T_{dip}} \sum_{j=1}^{S(k)} \frac{\min(R_{ached}(k, j), C(i))}{\min(C(i), R_{original}(k, j))} Pop(k, j) P(i) \quad (5)$$

假设代理缓存的磁盘容量为  $Cache$  bits。为了使(5)式中的  $\overline{QHR}$  最大化,需要在(6)、(7)式的约束下求出  $R_{ached}(k, j)$ 。

$$\sum_{k=1}^{T_{dip}} \sum_{j=1}^{S(k)} l(k, j) R_{ached}(k, j) \leq Cache \quad (6)$$

$$R_{ached}(k, j) \leq R_{original}(k, j) \quad (7)$$

### 3 异构环境下的代理缓存布局方法

在本文所研究的场景中,相同内容的流媒体片段只存储

一份,因此,如何确定  $R_{ached}(k, j)$  成为解决代理缓存最优化布局问题的关键。(5)式中  $R_{ached}(k, j)$  都是以片段为考察对象,为简洁起见,用一个全局标识符  $m$  来表示一个片段,设总共有  $n$  个片段。令

$$G_m(R_{ached}(m)) = \sum_{i=1}^N \frac{\min(R_{ached}(m), C(i))}{\min(C(i), R_{original}(m))} \frac{Pop(m)P(i)}{T_{pop}} \quad (8)$$

则(5)式可以简化为

$$\overline{QHR}(R_{ached}(m)) = \sum_{m=1}^n G_m(R_{ached}(m)) \quad (9)$$

我们采用动态规划的方法来递推地求出使  $\overline{QHR}$  最大化的解  $R_{ached}(m)$ 。为此,需要引入一些辅助变量来描述。设  $x(m)$  是分配给从第  $m$  个片段到第  $n$  个片段的缓存空间,  $u(m)$  是分配给第  $m$  个片段的缓存空间。第  $m$  个片段所占据的缓存空间  $u(m)$  与其能提供的最大传送速率  $R_{ached}(m)$  和时间长度  $l(m)$  的关系如(10)式所示。

$$u(m) = R_{ached}(m) l(m) \quad (10)$$

根据  $x(m)$  的定义,  $x(m+1)$  与  $x(m)$ 、 $u(m)$  的关系如(11)式所示。

$$x(m+1) = x(m) - u(m) \quad (11)$$

设  $f_m(x(m))$  是将容量为  $x(m)$  的缓存空间分配给从第  $m$  个片段到第  $n$  个片段时  $\overline{QHR}$  的最大值,那么  $f_m(x(m))$  可以用(12)、(13)式递推地计算出来,其中  $m = n-1, \dots, 1$ 。

$$f_n(x(n)) = \max_{0 \leq R_{ached}(n) \leq \min(x(n)/l(n), R_{original}(n))} G_n(R_{ached}(n)) \quad (12)$$

$$f_m(x(m)) = \max_{0 \leq R_{ached}(m) \leq \min(x(m)/l(m), R_{original}(m))} [G_m(R_{ached}(m)) + f_{m+1}(x(m+1))] \quad (13)$$

由  $f_m(x(m))$  的定义可知,  $f_1(x(1))$  就是将代理缓存的全部空间都分配给所有  $n$  个片段时  $\overline{QHR}$  的最大值。一旦求出了  $\overline{QHR}$  的最大值,采用与(12)、(13)式所示的递推方向相反的步骤就可以求出相应的  $R_{ached}(m)$ , 从而获得在异构环境下使质量命中率最大化的代理缓存布局。在用计算机对上述过程进行计算时,需要将  $x(m)$  和  $u(m)$  的变化量进行离散化处理。设在每次循环中,它们变化的步长值为  $unit$ 。具体的过程用伪码表示如表 1。

令步长值  $unit$  与  $Cache$  的比值为  $step$ , 简称为步长比例,则上述方法的时间复杂度为  $O(n \lceil 1/step \rceil^2)$ 。

## 4 性能仿真

本节的 4.1 小节介绍异构环境下流媒体代理缓存仿真环境的建立,同时给出了用基于质量命中率的代理缓存布局方法(Layout for Quality Hit Ratio, 简称为 LQHR) 求出的布局结果。在 4.2 小节中,我们将 LQHR 与文[5]中采用的质量恒定缓存布局(Layout for Constant Quality, LCQ) 和文[10]提到的质量分层缓存布局(Layout for Layered Quality, LLQ) 进行了比较。仿真结果表明, LQHR 在质量命中率和请求延迟率等方面均优于 LCQ 和 LLQ。

### 4.1 仿真环境

设在异构环境中有 5 类不同的用户,他们的接入带宽  $C(i)$  及所占比例  $P(i)$  如表 1 所示。 $C(i)$  呈等比增长,可以覆盖一个较大的带宽范围<sup>[11]</sup>。 $P(i)$  的设置与文[11]相同,用于区分以窄带用户为主 S-Narrow、均匀分布 Uniform 和宽带用户为主 S-Wide 这 3 种不同的异构环境。

表1 基于质量命中率的流媒体代理缓存布局方法(Layout for QHR, LQHR)

```

输入: 片段流行度  $Pop(m)$ , 接入带宽分布  $C(i)$ , 异构用户比例  $P(i)$ , 代理缓存容量  $Cache$ 
01: 根据(12)式, 初始化  $f_n(x(n))$ ;
02: for  $m=(n-1): (-1): 1$ 
03: { for  $x(m)=0: unit: Cache$ 
04:   {  $f_m(x(m)) \leftarrow -\infty$ ;
05:     for  $u(m)=0: unit: \min(x(m), R_{original}(m) l(m))$ 
06:       { 分别根据(10)和(11)计算  $R_{cached}(m)$  和  $x(m+1)$ ;
07:          $tmp \leftarrow G_m(R_{cached}(m)) + f_{m+1}(x(m+1))$ ;
08:         if  $tmp > f_m(x(m))$  then  $f_m(x(m)) \leftarrow tmp$ ; } }
09:  $\overline{QHR} \leftarrow \max_{x(1)=0: unit: Cache} (f_1(x(1))), maximum \leftarrow \overline{QHR}$ ;
10: 将  $f_{n+1}(x(m+1))$  置零, 其中  $x(m+1)=0: unit: Cache$ 
11: for  $m=1: 1: n$ 
12: { for  $x(m)=0: unit: Cache$ 
13:   { for  $u(m)=0: unit: \min(x(m), R_{original}(m) l(m))$ 
14:     { if  $maximum == G_m(u(m)/l(m)) + f_{m+1}(x(m+1))$ 
15:        $R_{cached}(m) \leftarrow u(m)/l(m)$ ;
16:        $maximum \leftarrow f_{m+1}(x(m) - u(m))$ ; } } }
输出:  $\overline{QHR}, R_{cached}(m), m=1, 2, \dots, n$ 
    
```

表2 异构环境的参数设置

	P(1)	P(2)	P(3)	P(4)	P(5)
C(i) (bps)	64	128	256	512	1024
S-Narrow	0.50	0.20	0.15	0.10	0.05
Uniform	0.20	0.20	0.20	0.20	0.20
S-Wide	0.05	0.10	0.15	0.20	0.50

文[1]发现一部流媒体影片的内容流行度分布类型随影片类型的不同而不同。在商业娱乐类的热门影片中, 比较容易观察到其内容流行度从头至尾递减的现象; 而在教学类的热门影片中, 比较容易观察到其内容流行度从头至尾高低起伏的现象, 显然后者更具有普遍意义。我们用 Zipf-like 类型的内容流行度曲线在同一部影片内的周期性再现来模拟内容流行度从头至尾高低起伏的情况。同时, 不失一般性, 这些 Zipf-like 曲线的参数  $\theta$  在 0~1 之间随机生成。

为了详细地分析异构环境和内容流行度分布对代理缓存布局的影响, 在本小节中只考虑源服务器上只有一部影片的情况(4.2节考虑多部影片的情况)。该影片在源服务器中能提供的最大传送速率是 256bps, 片长为 3000s。以代理缓存总容量为该影片的 1/2 为例, 图1、图2和图3分别示出了同一部影片在不同的异构环境和内容流行度分布下在代理缓存中能提供的最大传送速率, 也就是它的布局示意图。为便于比较, 对这3幅图的纵坐标进行了归一化处理, 也就是说 Rate Cached 是  $R_{cached}(m)$  与  $R_{original}(m)$  的比值, Segment Popularity 是  $Pop(m)$  与  $\max_{m=1, 2, \dots, n} Pop(m)$  的比值。这部影片被分成 100 个片段( $n=100$ ), 每个片段的时间长度为 30s。按自然顺序从片头到片尾对每个片段进行编号(片段序号): 1, 2, ..., 100。每 10 个片段为一个周期, 对应于一种 Zipf-like 分布。不同周期内的 Zipf-like 分布的参数不同。step 的值取为 0.005。

对比图1、图2、图3可知, 在相同的内容流行度分布下, 从 S-Narrow 到 S-Wide, 随着宽带用户逐渐增多, 越来越多的低流行度的片段被挤出代理缓存(Rate Cached 为 0), 节省出来的空间使得越来越多的高流行度的片段能提供越来越大的传送速率, 以满足越来越高的质量请求。而在同一种异构环境中, 流行度比较接近的邻近片段能提供相近的传送速率, 例如编号为 71 到 80 的片段。而流行度相差比较悬殊的邻近片段能提供的传送速率的差别也往往较大, 例如编号为 1 到 10 的片段。

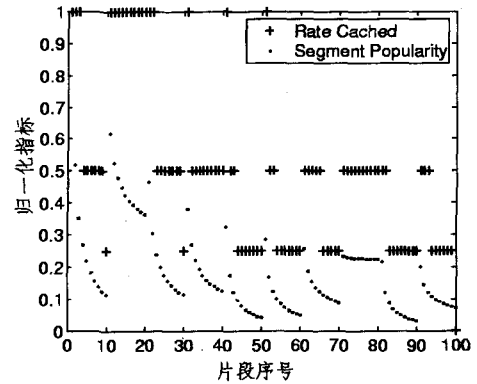


图1 S-Narrow下的布局图

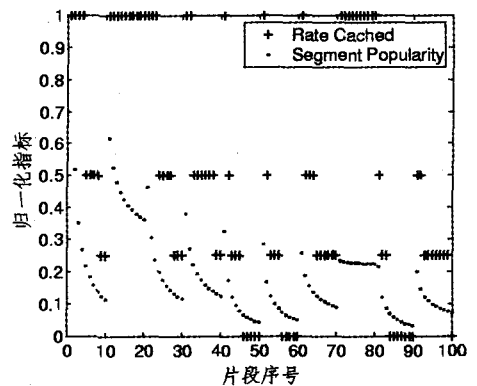


图2 Uniform下的布局

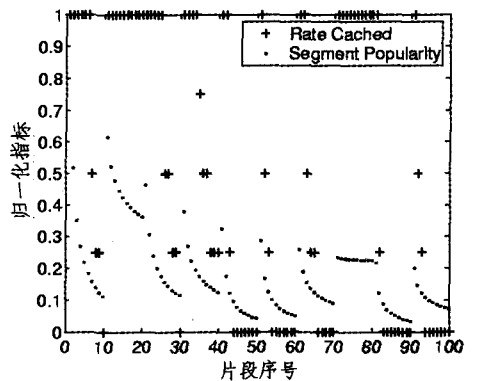


图3 S-Wide下的布局

以上分析表明, 在一定的代理缓存总容量的约束条件下, 流媒体片段所能提供的传送速率主要受用户成分(窄带用户

为主还是宽带用户为主)的影响;宽带用户越多,对传送速率的要求越高。除此以外,相邻片段所能提供的传送速率之间的差别还受到内容流行度的影响:相邻片段的内容流行度差别越大,说明用户对流媒体内容感兴趣的程度越不一致,因而也就不必要提供相同的传输质量。这一结果与我们的期望是一致的。

#### 4.2 性能比较

在本小节中,我们将 LQHR 与文[5]中采用的质量恒定缓存布局(Layout for Constant Quality, LCQ)和文[10]提到的质量分层缓存布局(Layout for Layered Quality, LLQ)进行了比较。为公平起见,我们将 10 部片长为 3000s 的 CBR 流媒体影片划分成相同时间长度(300s)的片段(共计 100 个片段),在源服务器上能提供的最大传送速率为 512 kbps。假设它们的内容流行度分布是 Zipf-like 类型,每一部影片的内容流行度 Zipf-like 的参数  $\theta$  在 0~1 之间随机产生。图 4,5,6 示出了在不同的异构环境中 LQHR,LCQ 和 LLQ 所能达到的平均质量命中率  $\overline{QHR}$ 。

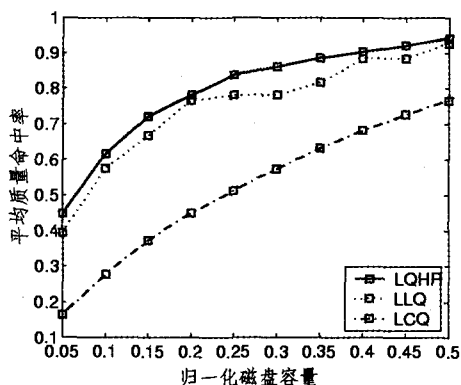


图 4 S-Narrow 下的  $\overline{QHR}$

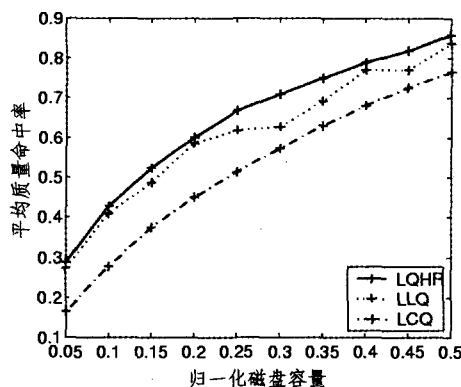


图 5 Uniform 下的  $\overline{QHR}$

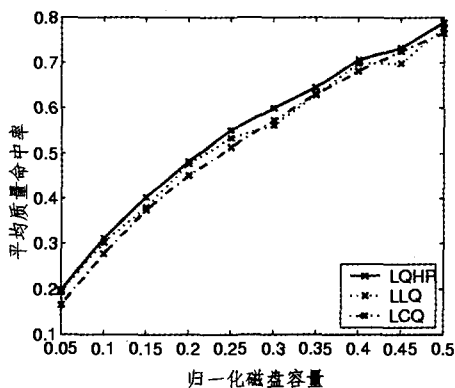


图 6 S-Wide 下的  $\overline{QHR}$

如图 4,5 和图 6 所示,在 S-Narrow、Uniform 和 S-Wide 3 种不同的异构环境下,我们的布局方法 LQHR 所获得的  $\overline{QHR}$  与 LLQ(LCQ)相比,最大超额分别达到了 8.0%(33.8%), 8.1%(15.0%)和 3.7%(2.6%)。在 S-Wide 这种异构环境下,3 条  $\overline{QHR}$  的性能曲线十分接近,这意味着 3 种方法所获得布局结果十分相近。这是因为在 S-Wide 异构环境中,为了满足以宽带接入为主的用户群对传送质量的高要求,每种布局方法都试图给高流行度片段以较高的传送速率。所不同的是 LCQ 只能给每个流行度片段分配固定的传送速率,也就是该片段在源服务器中能提供的最大传送速率,而 LLQ 只能在有限的 4 种传送速率中选择比较合适的一种。但是,在基于质量命中率的代理缓存布局方法中,LQHR 能够以 0.5% 的步长比例(step)精细地搜索最优解,因而所获得的质量命中率是三者中最高的。

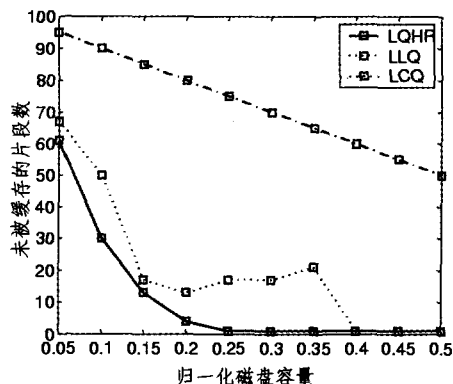


图 7 S-Narrow 下未被缓存的片段数

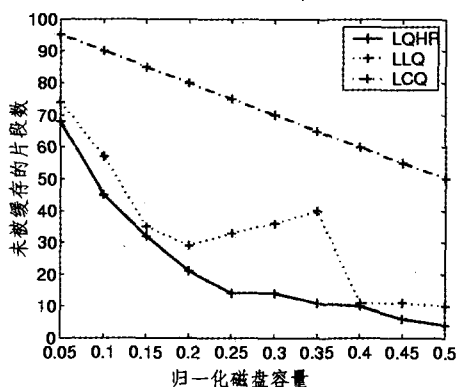


图 8 Uniform 下未被缓存的片段数

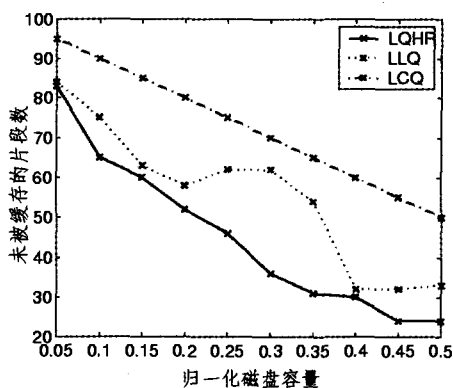


图 9 S-Wide 下未被缓存的片段数

图 7,8,9 示出了在不同的异构环境中 LQHR,LCQ 和 LLQ 的性能比较。(下转第 144 页)

表 2 未做数据预处理的数据

age	Symp1	Symp2	Symp3	Smk1	SmkT	Drk1	hth	fHis1	fHis2	Pres	Chek1
48			2	1		1	1		2	112/57	1.40(1)
74	2			1		1	1			135/65	8.8(2)
92	2			1		1	1			181/92	168(3)
72	2			2	60		1			105/78	36(4)
51		2		1		2	1			90/65	
51		2		1		1	1			170/98	176(3)

表 3 预处理后的数据

Age0	Symp0	Smk0	hth	fHis0	Pres0	Chek1
2	1	1	1	1	1	2(1)
1	2	1	1	2	1	2(2)
1	2	1	1	2	3	3(3)
1	2	2	1	2	1	3(4)
2	2	2	1	2	1	1
2	2	1	1	2	3	3(3)

参考文献

1 万志红, 范千云. 电子病历系统[J] 中国医药卫生信息, 2002, 22

(1):246~250

2 刘安滨. 再谈电子病历[J]. 中国医院管理, 2003, 23(4):37~38  
 3 李少龙, 吴一龙, 汪建平. 立体化电子病历与询证医学的研究. 询证医学, 2003, 3(2):74~77  
 4 Basili R, Paziienza M T. Lexical Acquisition and Information Extraction [C]. Information Etraction: A Multidisciplinary Approach to an Emergin Information Technology, 1997, 1299:44~72  
 5 Han Jiawei, Kamber M. Data Mining: Concepts and Techniques [M]. USA: Morgan Kaufmann Publishers, 2001. 70~95

(上接第 125 页)

LLQ 这 3 种代理缓存布局方法所不能容纳的流媒体片段数目, 或称为“未被缓存的片段数”。

在一定的存储容量下, 不能容纳的流媒体片段越少, 说明代理缓存的利用率越高, 代理缓存能够直接响应来自用户请求的机会越大, 而片段的播放时延也越短。如图 7, 8 和图 9 所示, 在 S-Narrow、Uniform 和 S-Wide3 种不同的异构环境下, 我们的布局方法 LQHR 所容纳的流媒体片段数目与 LLQ(LCQ)相比, 最大超额分别达到了 20(76), 29(61)和 26(34)。LLQ 的性能曲线在上述 3 幅图中并不是始终随着磁盘空间的增加而降低的。这是因为: LLQ 这种布局方法中所能选择的 4 种传送速率之间的差额是呈指数率增长的, 当归一化磁盘容量在 0.2 到 0.3 之间变化时, 相同的一块存储空间分配给已有的缓存片段来增加其传送速率比分配给一个新的片段更能满足用户对观看质量的请求。

虽然 LQHR 采用了比 LLQ 更细致的搜索步长来搜索合适的传送速率, 但是在 S-Narrow、Uniform 和 S-Wide 这 3 种不同的异构环境下, 在代理缓存的布局结果中出现的不同的传送速率的平均值分别为 4.0, 4.3 和 4.7, 并不会为现有的代理缓存系统增加很多的系统开销。

小结 在网络边缘对流媒体的热点片段进行缓存是流媒体分发研究领域的一个热点。异构环境下不同种类的接入用户对流媒体的播放质量具有不同的要求。为了使代理缓存最大限度地满足全体用户对收看质量的请求, 本文首先提出了质量命中率的定义来衡量代理缓存对提高用户观看质量的贡献, 然后提出了一种基于质量命中率的流媒体代理缓存布局方法。仿真结果表明, 该布局方法不仅能使全体用户的平均质量命中率最大, 而且有效地利用了代理缓存的磁盘空间, 显著降低了片段的播放时延。

后续工作将在此基础上, 围绕流媒体代理缓存在无线蜂

窝网中的应用展开。

参考文献

1 Costa C, Cunha I, Borges A, et al. Analyzing client interactivity in streaming media. In: Proc. 13th Int World Wide Web Conf., New York, NY, May 2004. 534~543  
 2 Sen S, Rexford J, Towsley D. Proxy prefix caching for multimedia streams. In: Proc. IEEE INFOCOM, Mar. 1999. 1310~1319  
 3 Chae Y, Go K, Buddhikot M M, et al. Silo, rainbow and caching token: schemes for scalable tolerant streaming caching. IEEE Journal on Selected Areas in Communications, 2002, 20(7):1328~1344  
 4 Chen S, Shen B, Wee S, et al. Adaptive and lazy segmentation based proxy caching for streaming media delivery. In: Proc. of 13th Int Workshop on NOSSDAV, Monterey, CA, 2003. 22~31  
 5 Wu K, Yu P S, Wolf J L. Segmentation of multimedia streams for proxy caching. IEEE Transactions on Multimedia, 2004, 6(5):770~780  
 6 Liu W, Chou C T, Yang Z K, et al. Popularity-wise proxy caching for interactive streaming media. In: Proc. 29th IEEE LCN, 2004. 250~257  
 7 Rejaie R, Yu H, Handley M, et al. Multimedia proxy caching mechanism for quality adaptive streaming applications in the Internet. In: Proc. IEEE INFOCOM, Mar. 2000. 980~989  
 8 Kangashariju J, Hartanto F, Reisslein M, et al. Distributing layered encoded video through caches. IEEE Transactions on Computers, June 2002, 51(6):622~636  
 9 Tang X, Zhang F, Chanson S T. Streaming media caching algorithms for transcoding proxies. In: Proc. 31st Int Conf. on Parallel Processing, Aug. 2002. 287~295  
 10 Zine M, Schmitt J, Steinmetz R. Layer-encoded video in scalable adaptive streaming. IEEE Transactions on Multimedia, Feb. 2005, 7(1):75~84  
 11 Liu J, Chu X, Xu J. Proxy cache management for fine-grained scalable video streaming. In: IEEE INFOCOM, Mar. 2004. 1490~1500