

# 一种基于 Sever 和 Proxy 流媒体流行性的 Caching 策略

谭 劲 余胜生 周敬利  
(华中科技大学 武汉430074)

## A Popularity-Based on Server-Proxy Caching Strategy for Streaming Media

TAN Jin YU Sheng-Sheng ZHOU Jing-Li  
(Huazhong University of Science and Technology, Wuhan 430074)  
jtan@wtwh.com.cn

**Abstract** It is expected that by 2003, continuous media will account for more than 50% of the data available on origin servers. This will provoke a significant change in Internet workload, due to the high bandwidth requirements and the long-lived nature of digital video, streaming server loads and network bandwidths are proving to be major limiting factors. Aiming at the characteristic of broadband network in a residential area, we propose a popularity-based on server-proxy caching strategy for streaming media. According to a streaming media popularity on streaming server and proxy, this strategy caches the content of this streaming media partially or completely, and plays an important role in decreasing server load, reducing the traffic from streaming server to proxy, and improving the start-up latency of the client.

**Keywords** Streaming server, Proxy caching, Streaming media, VOD, Popularity

## 1 概述

经过了2000、2001两年的社区宽带网建设的高速发展后,摆在中国 ISP 们面前的任务是如何在已建成的宽带网上开展增值服务,许多 ISP 尝试在宽带网上开展流媒体(Streaming Media)服务,如视频点播 VOD(Video On-Demand)系统。然而,流媒体对网络带宽和实时性的要求使得流服务器必须能够进行端对端(End-to-End)的拥塞控制和质量调整,由于 Internet 用户的集聚增加导致的拥塞网络和超负荷的流服务器使互连网上的视频流缺乏质量保证,用户必须忍受高启动延迟(Start-Up Latency)和不可预知的播放质量。为了提高整个流媒体系统的性能,在靠近用户的 Proxy 端缓存流媒体对象就变得越来越重要。

尽管 Proxy Caching 在传统的 Web 对象(文本与图像)中获得了巨大的成功(节省带宽、服务器负载均衡和改善感知的网络延迟),但代理服务器还没有有效地用于缓存(Caching)像音频和视频这样的流媒体,这是因为流媒体与传统的 Web 对象相比有自己内在的属性,主要表现在以下几个方面:

- 流数据通常在 UDP 之上传输,由应用程序或应用层协议负责拥塞控制。

- 产生的网络流量趋于突发(Burst),并对延迟相当敏感。

- 流媒体对象比传统的 Web 对象大几个数量级,潜在地增加了代理服务器 Cache 的存储需要,需要更复杂的缓存替换策略。

- 与其它 Web 对象相反,流媒体不需要一次性传送。服务器通常通过网络流水线传送数据给客户。

- 流媒体可以通过调整它们的质量来改变它们的大小。

- 流媒体文件遵循 WORM(Write-Once-Read-Many)原则,一旦一个流媒体文件在线后,十之八九不会改变,因此,流媒体缓存的一致性不是一个主要问题。

为了降低流服务器的负载,减少流服务器到社区代理服务器间的网络流量,提高服务质量(QoS),改善客户启动延迟(Start-up latency),我们结合社区宽带网结构和社区住户访问的特点,提出了一个基于流媒体在流服务器的流行性系数和在 Proxy 上的流行性系数的 Caching 策略。

## 2 社区流媒体服务特点

社区宽带网和社区住户播发流媒体具有以下特点:

- 社区宽带网带宽一般都是100M到楼,10M到户,通过一个代理服务器连接到 Internet。因此,社区内的用户到 Internet 的连接是同类的(Homogeneous),不用考虑代理服务器到客户端的质量调整。

- 流媒体的提供者为建设该社区宽带网的 ISP。因此便于流媒体服务器与社区代理服务器间的协调。

- 收费实行包月制,不用考虑按次计费(pay-per-view)。

- 住户上网时间分布明显,一般是在晚上和节假日,晚上21:00达到高峰。

图1描绘了为社区提供流媒体服务的整个网络结构。

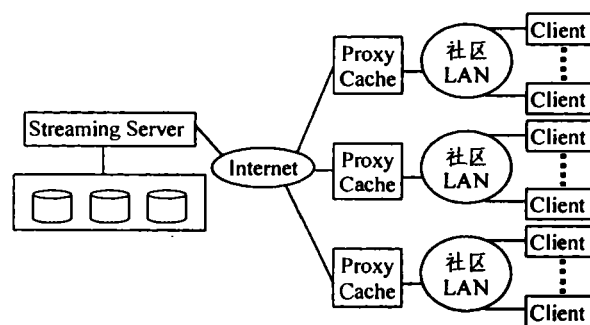


图1 社区流媒体服务的网络结构

谭 劲 博士生,主要研究方向是网络通信、多媒体技术、系统结构等。余胜生 教授,博导,主要研究方向是网络通信、多媒体技术、外存储技术等。周敬利 教授,博导,主要研究方向是网络通信、多媒体技术、外存储技术等。

### 3 Caching 策略

随着流媒体在 Internet 上流行起来(预计到2003年,连续媒体将占源服务器上可用数据的50%),目前国外许多高校和科研机构正在致力于流媒体 Caching 的研究。这些研究的主要目的就是克服传统 Proxy Caching 的缺点,使其能有效地支持 Internet 上的连续媒体数据的传送。文[3]提出了缓存请求的对象的前缀(Prefix)部分策略,即在 Proxy 上存储流媒体的初始帧,一旦接收到对该流媒体的请求,Proxy 就用缓存的前缀数据提供给用户初始化,同时向流服务器请求其余的帧,该策略改善了启动延迟;文[4]提出了一种缓存分层编码(Layered-Encoded)的流媒体的策略,即缓存的数据量随着网络的流量而变化,网络的流量越大,缓存的数据也越多,播放的质量也越好。

由于考虑到 Proxy 的磁盘容量,目前提出的缓存策略都基于部分缓存,并且只考虑了 Proxy 服务器端流媒体的流行性(Popularity)。而硬盘以容量每年增加60%的速度发展<sup>[6]</sup>,320G 的 POP Server 的价格才服务器的一半,所以我们提出了基于流服务器(Streaming Server)和代理服务器上的流媒体的流行性相结合的部分缓存和完全缓存并存的缓存策略。为简单起见,作如下假定:

- 从 Proxy 服务器到客户端(宽带连接)有丰富的带宽供应。
- Proxy 服务器具有足够的存储容量 G(Byte)存储流服务器上大部分流文件的前面部分(假定前5秒,以 MPEG-1 为例,80G 的硬盘可以存储上万个流媒体对象的前面部分)和部分流行(受欢迎)流文件的整个内容。
- Proxy 上的硬盘(或 POP Server)具有无限的存储带宽。
- 从 Proxy 到用户的传输延迟远小于从 Streaming Server 到 Proxy 的传输延迟。

#### 3.1 策略

由于流媒体服务器和社区 Proxy 属同一 ISP 管理,我们提出了如下 Caching 策略。

所有的连续媒体对象都存储在流服务器上,每个流对象在流服务器上有一个流行(受欢迎)系数  $P_S$ ,该系数在流媒体对象在线时就已产生,用户通过 Proxy 服务器对某个流媒体的访问会改变该媒体在流服务器上的  $P_S$ 。当某社区的某个用户第一次访问该流媒体对象时,根据该对象  $P_S$  的确定是作部分(Partial)缓存还是作完全(Complete)缓存。对于已在 Proxy 上作部分缓存的流媒体,后续用户一旦提出对该流媒体的请求,Proxy 就用缓存的前缀数据提供给用户初始化,改善用户的启动延迟。访问结束后,改变 Proxy 服务器上的流行系数  $P_P$ ,并传给流服务以改变  $P_S$ ;对于已在 Proxy 上作完全缓存的流媒体,后续用户一旦提出对该流媒体的请求,就直接通过本地网络访问 Proxy,改善了用户的启动延迟,减少了从流服务器到 Proxy 服务器间的网络流量,降低了流服务器的负载。访问结束后,改变 Proxy 服务器上的流行系数  $P_P$ ,并传给流服务器以改变  $P_S$ 。这样一来, $P_P$  表示一个社区的流行性, $P_S$  表示所有社区的流行性。当 Proxy 的空间不够缓存一个流媒体的部分或一个完整的流媒体对象时,系数  $P_P$  作为缓存对象替换(Replacement)的一个重要参数。

#### 3.2 流服务器端流行系数 $P_S$

当第一个用户通过 Proxy 访问一个流媒体对象时,Proxy

怎么能事先知道是作部分缓存还是作完全缓存?如果对于一个很流行的流媒体,第一次就作完全缓存将会大大地改善用户的启动延迟,减少从流服务器到 Proxy 服务器间的网络流量,降低流服务器的负载。在现实中,要准确地确定一个流媒体在流服务器上的流行性系数是很困难的,但有一点是肯定的,即所有的流媒体对象的流行性是不相同的。研究表明典型的流媒体流行性模型是遵循 Zipf 规律分布的:

$$Z(i) = C/i, C = 1 / \left( \sum_{i=1}^N 1/i \right) \quad i \in \{1, \dots, N\} \quad (1)$$

然而,Zipf 规律并没有直接告诉我们在 N 个流媒体对象中哪个最流行。一般说来,一个流媒体的流行性除了与它的内在属性(故事情节、演员、导演等)有关外,还与它的外在属性(广告宣传、在线时间等)有关,并随着时间而变化的。我们在这里定义的流行系数  $P_S$  的主要目的并不是要准确地求出一个流媒体相对于其它 N-1 个流媒体的流行性系数的值,而是在用户访问某个流媒体对象时告诉 Proxy 是作部分缓存还是作完全缓存。因此,根据经验和感受(例如所谓的大片、新片)是可以预见该媒体在线(Online)时是需要作完全缓存还是作部分缓存的,流媒体服务器需要给所有的流媒体对象定义一个流行性域值  $C_{Threshold}$ 。在一个流媒体刚在线时,管理员根据他或她自己的经验和感受给该媒体确定一个初值 W:

$$W = C_{Partial} \text{ 或 } C_{Complete}$$

其中  $C_{Partial}$  表示作部分缓存( $C_{Partial} \leq C_{Threshold}$ ),表示完全缓存( $C_{Complete} > C_{Threshold}$ ),调整  $C_{Partial}$  或  $C_{Complete}$  的大小,会改变流媒体在服务器上标明作部分缓存或完全缓存的持续时间。

给定流服务器上的某个流媒体,在线(Online)时间为  $T_0$ ,当前有 M 个 Proxy,用户通过第 i 个 Proxy 访问该媒体对象时,访问时间为  $T_T(i)$ 。流服务器根据下面的公式计算出当前  $P_S$  的值,Proxy 得到该值后,确定是作部分缓存还是作完全缓存。访问结束后,就在 Proxy 端形成该媒体的流行系数  $P_P$ ,并将该系数传送到流服务器端调整该媒体的流行性系数  $P_S$ , $P_S$  的计算公式为:

$$P_S = \sum_{i=1}^M P_P(i) * 2^{-(T_T(i) - T_0)/\lambda} + W \quad (2)$$

其中, $P_P(i)$  为该媒体在第 i 个 Proxy 上的流行系数, $i \in \{1, \dots, M\}$ , $\lambda$  为控制流行性系数  $P_S$  衰减速率的系数。在流媒体刚在线时  $P_S = W$ 。 $P_P(i)$  可以为负数,也可以为正数,当  $P_S \leq C_{Threshold}$  时,Proxy 作部分缓存;当  $P_S > C_{Threshold}$  时,Proxy 作完全缓存。

#### 3.3 Proxy 端流行系数 $P_P$

对于一个已部分缓存或完全缓存在 Proxy 上的流媒体,我们用访问度  $A_D$  (Access Degree) 来表示该流媒体在 Proxy 端的流行系数  $P_P$ ,访问度  $A_D$  为用户观看流媒体的长度与流媒体的实际长度的比率,它比访问频率更能准确地反映出流媒体的流行性。假定社区内有 U 个用户观看了该媒体,当第 i 个用户观看该媒体时,该媒体的访问度为:

$$A_D(i) = \frac{\text{PlaybackTime}}{\text{StreamLength}} \quad (3)$$

当用户没有看该流媒体时(也许看了流媒体内容的文字介绍)时, $A_D(i) = 0$ 。当用户观看完整个流媒体时, $A_D(i) = 1$ 。因此, $0 \leq A_D(i) \leq 1$ 。

假定  $T_C$  为最后一次部分或完全缓存该媒体的时间, $T_T(i)$  为第 i 个用户访问的时间。则 Proxy 端流行系数  $P_P$  为:

$$\text{完全缓存时, } P_P = \sum_{i=1}^U A_D(i) - \lambda_c * \sum_{i=1}^U \text{Log}(T_T(i) - T_C) \quad (4)$$

$$部分缓存时, P_F = \sum_{i=1}^U A_D(i) - \lambda_p * \sum_{i=1}^U \text{Log}(T_T(i) - T_C) \quad (5)$$

$\lambda_c$  和  $\lambda_p$  为 Proxy 端控制该媒体完全缓存和部分缓存的流行性系数  $P_F$  随着时间衰减的系数。如果  $\lambda_c$  大于  $\lambda_p$ , 将使部分缓存优先于完全缓存, 反之将使完全缓存优先于部分缓存, 可以根据 Proxy 端的磁盘容量改变  $\lambda_c, \lambda_p$  的值来调整缓存策略。当某个流媒体由部分缓存转为完全缓存或由完全缓存转为部分缓存时, 以前所有用户对该媒体的访问度  $A_D$  清零,  $T_C$  也随着变化, 也就是重新形成该媒体的 Proxy 端的  $P_F$ 。

### 3.4 替换算法

当 Proxy 的空间不够缓存一个流媒体的部分或一个完整的流媒体对象时, 就要进行完全缓存对象的替换 (Caching Replacement), 即释放掉流行性差 ( $P_F$  的值最小) 的部分缓存或完全缓存的流媒体占用的空间, 以便部分或完全缓存新访问的流媒体。在释放已缓存的流媒体时, 应遵守以下规则:

- 在释放完全缓存媒体占用的空间时, 并不是完全释放该媒体 (应保留该媒体的前面部分, 假定前5秒), 该媒体相当于从完全缓存转到部分缓存。

- 在释放  $P_F$  值最小的部分或完全缓存的流媒体之前, 如果该流媒体正在被用户访问, 就不能释放。

- 释放时应将  $P_F$  值传回流服务器端, 以便改变流服务器端流行系数  $P_S$ 。

- 如果部分缓存和完全缓存的两个流媒体具有相同的最低  $P_F$  值, 在对缓存的流媒体的  $P_F$  值进行升序排序时, 完全缓存的  $P_F$  值应排在前面, 即先释放完全缓存的媒体。

下面给出了缓存和释放已缓存的对象的伪代码:

```

Caching-Process(Stream i)
{
  GetFromServer P_S(i);
  if (P_S(i) > C_Threshold)
  if NotCompleteCached(i) // 如果该媒体未作完全缓存
  if EnoughFreeCompleteSpace(i)
    CompleteCaching(i)
  Else
    if ReleaseSpace(CompleteSpace(i)) CompleteCaching(i)
else
  if NotPartialCached(i) // 如果该媒体未作部分缓存
  if EnoughFreePartialSpace(i)
    PartialCaching(i)
  Else
    if ReleaseSpace(PartialSpace(i)) PartialCaching(i)
}
ReleaseSpace(Space s)
{
  Sort P_F;
  s0=0;
  j=0;
  while (s0 < s)
  {
    if NotAccessedCachedStream(j) // 如果缓存的媒体没有被访问
    {
      SendbackToServer P_F(j);
      s0 = s0 + CachedStream(j);
      Evict(CachedStream(j));
    }
    j++;
  }
  return s0;
}
    
```

## 4 性能分析

一般说来, 用命中率 HR (Hit Ratio) 和位命中率 BHR (Byte Hit Ratio) 两个参数能很好地反映传统缓存系统的性能。由于流媒体比其它 Web 对象 (如文本和图像) 大几个数量级, 流媒体缓存系统中有部分缓存和完全缓存的对象, HR 不能很好地反映出其性能, 而我们设计的缓存策略的主要目的

是改善用户的启动延迟, 减少从流服务器到 Proxy 服务器间的网络流量, 降低流服务器的负载。因此, 我们用 BHR 和流量减少率 TRR (Traffic Reduction Ratio) 两个参数来衡量我们提出的缓存策略的性能, 这里:

$$BHR(i) = \frac{CachedLength(i)}{StreamLength(i)} \quad i \in \{1, \dots, N\}, N \text{ 为媒体数} \quad (6)$$

对于 Proxy 而言, 我们用  $T_{Out}$  表示流出 (从 Proxy 到客户) 的总流量,  $T_{In}$  表示流进 (从服务器到 Proxy) 的总流量, 于是:

$$TRR = \frac{T_{Out} - T_{In}}{T_{Out}} \quad (7)$$

TRR 越大, 说明降低的流服务器负载和节省的网络流量也越大。

我们用图2所示的实验模型和表1所示的参数作模拟实验。BHR 和 TRR 如图3和图4所示。

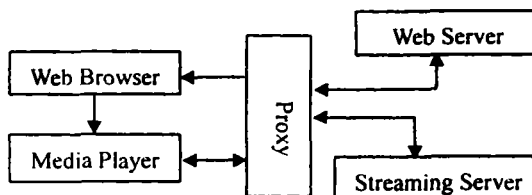


图2 实验模型

表1 实验参数

媒体数	17
媒体长度(秒)	153-3120
Proxy 硬盘容量(G)	9
部分缓存的长度	前5秒
LAN(即 Proxy)数	3
Proxy 与 Client 间的传输延迟(ms)	100
Proxy 与 Server 间的传输延迟(ms)	250

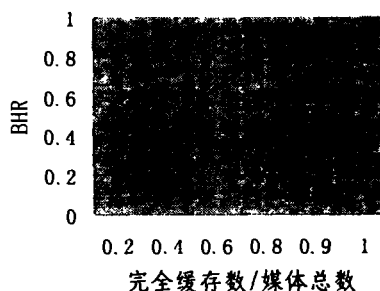


图3 BHR

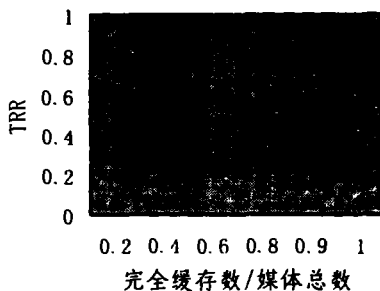


图4 TRR

(下转第101页)

数,  $C$  是要划分的类数,  $x_k$  是第  $k$  个数据样本,  $v_i$  是第  $i$  个类的类中心矢量,  $u_{ik}$  是第  $k$  个数据样本属于第  $i$  个类的隶属程度,  $m$  是一个大于 1 的常数。给定类数  $C$  和初始隶属度矩阵  $\{u_{ik}\}_{n \times c}$  (或初始中心矩阵  $\{v_i\}_{c \times c}$ )。FCM 算法通过迭代式 (2)、(3), 收敛到目标函数的一个局部极小点或一个鞍点。

$$u_{ik} = \left[ \frac{1}{d_{ik}} \right]^{\frac{2}{m-1}} / \sum_{j=1}^c \left[ \frac{1}{d_{ij}} \right]^{\frac{2}{m-1}}, k=1, 2, \dots, n, i=1, 2, \dots, C \quad (2)$$

$$v_i = \sum_{k=1}^n u_{ik}^m x_k / \sum_{k=1}^n u_{ik}^m, i=1, 2, \dots, C \quad (3)$$

其中  $d_{ik} = \|x_k - v_i\|_A$  是样本点到类中心的距离。

使用该聚类方法, 可以将相似的个体用户兴趣归类, 便于新的用户继承已有类别用户的兴趣特征。

聚类的过程将兴趣类似用户的兴趣进行了综合, 扩充了个人兴趣表述中的向量空间的维度。个人兴趣继承群体兴趣后, 个人兴趣描述向量将根据群体兴趣进行修订。通过这种聚类、继承的运算后, 用户的兴趣相互补充, 完成了概念扩充的过程, 使得查全率增高。

#### 2.4 网页推荐及用户反馈

设用户兴趣度为  $I$ , 定义一个阈值  $I_k$ , 对文档集中  $Sim(V, U) > I_k$  的文档, 则由系统自动推荐给用户。

用户可以对过滤后文档集中的文档按照与自己信息需求的相关程度打分, 分数可以分为几个级别 (例如可以设为 3 个级别, 也可以分得更多更细), 最低为 0, 最高为 1。设文档为  $D$ , 兴趣度为  $I$ , 反馈信息形式可以表示为:  $\{(D_1, I_1), (D_2, I_2), \dots, (D_n, I_n)\}$ 。根据用户的反馈, 需要及时调整用户模型, 必要时调整群体用户模型。

用户反馈是提高查准率的重要手段, 该模型中能够很方便地实现这个反馈过程。这里用户反馈的兴趣使用关键词描述方法, 因此反馈信息可以方便地用于修改关键词特征向量的参数。

**结论** 本文提出了利用向量空间模型 (VSM) 和 FCM 聚类算法聚类生成群体用户兴趣和个体用户兴趣在服务器端进行网页推荐的结构模型。在已经进行的小样本的试验中, 初步

地证明了该方法有一定的可行性, 并且得到了较好的试验结果。这个方法能生成具有概念扩充能力的检索和过滤命令, 从而更好地帮助用户完成其特定信息的获取。这个结构模型不仅可以应用在单个网站中, 提供用户个性化浏览页面; 也可以应用在企业的信息采集系统中, 提供一种概念扩充的检索方法。这种结构模型的优点在于其在用户的反馈和浏览中进行概念的扩充和学习, 比一般的概念词典扩充方法更灵活, 检索效果更好。

#### 参考文献

- 1 Setal L. Searching the World Wide Web. Science, 1998, 280 (5360): 98~100
- 2 Lawrence S, Giles C L. Accessibility of information on the Web. Nature, 1999, 400: 107~109
- 3 <http://www.google.com>
- 4 Belkin N J, Croft W B. Information filtering and information retrieval: Two sides of the same coin. Communication of ACM, 1992, 35(12): 29~38
- 5 林鸿飞. 基于 Web 的信息过滤机制. 计算机工程与应用, 2002(2)
- 6 Pazzani M, Billsus D. Learning and revising user profiles: The Identification of Interesting Web Sites. Machine Learning, 1997, 27: 313~331
- 7 Letsche T A, Berry M W. Large-Scale Information Retrieval with Latent Semantic Indexing. Information Sciences, 1997
- 8 周水庚, 关侗红, 胡运发. 隐含语义索引及其在中文文本处理中的应用研究. 小型微型计算机系统, 2001, 22(2)
- 9 牛伟霞, 张永奎. 潜在语义索引方法在信息过滤中的应用. 计算机工程与应用, 2001, 9
- 10 Moldovan D I, Mihalcea R. Using WordNet and Lexical Operators to Improve Internet Searches. INTERNET COMPUTING, Jan.-Feb. 2000
- 11 Tu H-C, Hsiang J. An architecture and category knowledge for intelligent information retrieval agents Decision Support Systems, 2000, 28: 255~268

(上接第 72 页)

通过模拟实验我们发现: 如果一个媒体在线时, 能够确定其流行系数, 将大大地改善整个流媒体系统的性能, 它可以从完全缓存变成部分缓存, 也可以从部分缓存变成完全缓存, 由该媒体的访问度和访问的时间确定,  $\lambda$ ,  $\lambda_c$  或  $\lambda_p$  对流行性系数的影响很大, 它们的取值与用户访问模式与分布的关系我们还要作深入的研究。

**结束语** Proxy 缓存是一种由成本效益 (cost-effective) 改善互连网上流媒体服务系统性能的解决方案。通过在 Proxy 上缓存经常被访问的数据, 将会极大地减少用户感知的延迟、服务器负载和网络流量。然而, 传统的 Web Caching 技术不能很好地适用于流媒体数据, 我们针对社区宽带网的特点, 提出的一种基于 Server 和 Proxy 流行性的流媒体 Caching 策略, 利用流媒体在流服务器的流行性系数  $P_s$  和在 Proxy 上的流行性系数  $P_p$ , 在 Proxy 端对流媒体实行部分缓存或完全缓存, 改善了整个流媒体系统的性能。特别是在服务器端如果能准确地给出  $P_s$  的初值, 就更能体现出该策略的优越性。由于 Proxy 与流服务器之间的传输路径可能要经过好几个不同的 ISP, 我们下一步的工作就是探讨和解决流媒体

在这段路径上的包丢失的恢复和质量调整给已缓存的数据量带来变化的问题。

#### 参考文献

- 1 吴国勇, 邱学刚, 等. 网络视频 流媒体技术与应用. 北京邮电大学出版社, 2001
- 2 徐光怪, 史元春译. 多媒体系统设计. 电子工业出版社, 1998
- 3 Jrefod S S, Towsley D. Proxy Prefix Caching for Multimedia Streams. In: Proc. IEEE Infocom, March 1999
- 4 Rejaie R, et al. Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet. In: Proc. IEEE Infocom, March 2000
- 5 Reisslein M, Hartanto F, Ross K W. Interactive Video Streaming With Proxy Servers. In: Proc. IEEE Infocom, April 2000
- 6 Gibson G A, Vitter J S, Wilkes J. Storage and I/O Issues in Large-Scale Computing. In: ACM Workshop on Strategic Directions in Computing Research, ACM Computing Surveys, 1996. <http://www.medg.lcs.mit.edu/doyle/sdcr>.
- 7 Wang Y, Zhang Z-L, Du D, Su D. A Network-Conscious Approach to End-to-End Video Delivery over Wide Area Networks Using Proxy Servers. In: Proc. IEEE Infocom, April 1998