

流媒体服务中基于分布式代理的缓存数据放置策略

郭攀红¹ 杨 扬¹ 李新友²

(北京科技大学信息工程学院 北京 100083)¹ (国家信息中心 北京 100045)²

摘 要 随着高速宽带接入技术的发展,流媒体技术的研究得到了迅速的发展,并具有广阔的应用前景。流媒体代理技术作为减轻服务器的访问负载、提高用户的访问响应速度的重要手段,已成为流媒体研究领域中的研究热点之一。针对流媒体服务中的分布式代理服务器系统,提出了一种优化的缓存数据放置策略。其主要思想是将缓存数据放入某个特定的代理服务器中,使得今后访问该数据的网络传输开销最小。仿真实验表明,所提出的算法比传统的缓存数据放置算法能获得更小的传输开销和更好的可扩展性。

关键词 分布式代理,流媒体服务,缓存技术,数据放置,算法分析

Data Placement Scheme for Distributed Caching System in Media Streaming Service

GUO Pan-hong¹ YANG yang¹ LI Xin-you²

(School of Information Engineering, University of Science and Technology Beijing, Beijing 100083, China)¹

(State Information Center, Beijing 100045, China)²

Abstract With the improvement of the high-bandwidth access technologies, multimedia streaming has experienced quickly development, and has wide application prospect. As an important solution for reducing the load of server, improving the response time for customers, the steaming proxy technique has become one of the most popular research fields in the streaming research community. This paper presented an optimized data placement scheme for distributed caching system. The idea of the proposed scheme is to put the cache data into a particular proxy server, in order to minimize the transmission cost of data accessing in future. The simulation results demonstrate that our proposed algorithm achieves smaller transmission cost and has better scalability comparing with traditional cache data placement scheme.

Keywords Distributed proxy, Media streaming service, Cache technology, Data placement, Analysis of algorithm

1 引言

流媒体服务具有低延迟、低抖动、低丢失率、高实时性的特点,这就需要有高速、稳定的网络来保证流媒体内容能快速、稳定地被传输到用户端并进行播放。而 Internet“尽力而为”的服务模式无法满足视频流庞大的数据量和突发性的特点。目前比较通用的解决方案是引入代理服务器来解决这一问题。通过广泛分布在互联网各处的代理服务器节点,并通过代理缓存技术使流媒体内容被分布到离用户最近的地方,使得数据传输距离大大缩短,流量负荷被分流,从而缓解了对带宽的需求^[1,2]。

图 1 描述了在 Internet 网络中“服务器-代理-客户机”的体系结构。流媒体代理服务器通常位于 Internet 网络边缘,一端支持 Internet 用户,另一端连接 Internet 中的视频服务器。在用户端,Internet 用户通过 LAN, X-DSL 等与代理服务器相连,由于距离较近,传输链路可保障媒体流的服务质量(QoS)。代理服务器的另一端与视频服务器相连。一般地,代理服务器与流媒体服务器通过 Internet 连接且距离较远。代理服务器实际上是位于客户与服务器之间的一个应用层网

关,同时扮演客户端(对服务器)和服务器(对用户)的双重角色。基于流媒体服务的代理技术是流媒体研究领域中的重要课题。随着流媒体技术在 Internet 和无线网络环境中的高速发展,对流媒体代理服务器的研究也正在深入^[3]。

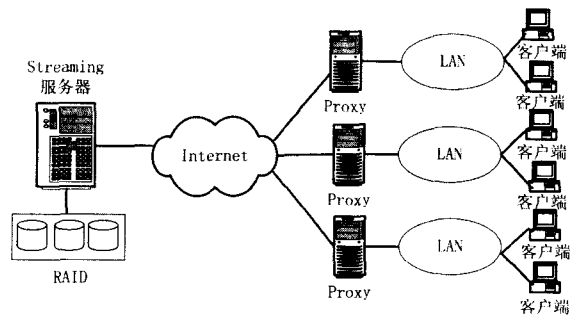


图 1 “服务器-代理-客户机”体系结构连接图

目前提出的各种代理缓存技术尽管有效地降低了流服务器和网络传输的负载,但仍不能满足含有大量用户群时对媒体请求的服务质量。这是由于:其一,单个代理的有限存储空间限制了整个系统为用户提供的媒体缓存空间;其二,媒体数据的传输通常要为每一个用户请求专门预留一定可持续的网

到稿日期:2008-12-22 返修日期:2009-03-10 本文受 863 国家重点基金项目(2007AA01Z234),国家自然科学基金(60873192)资助。

郭攀红(1976-),女,博士,主要研究方向为资源发现、P2P, E-mail: guopan hong@126.com; 杨 扬(1955-),男,博士生导师,主要研究方向为多媒体通信、图像语音识别; 李新友(1962-),男,博士,教授,主要研究方向为网络安全、图像处理。

络带宽,对代理服务器提出了较高的带宽 I/O 要求,使得代理能够同时提供的用户服务数有限。为解决这些问题,研究者们提出了将多个代理看成逻辑上的整体使其相互协作,共同向用户提供流媒体服务。采用相互协作的分布式代理技术往往能给用户提供更高质量、更小的服务延迟和更大的代理缓存空间^[4]。

在流媒体代理服务器中,缓存管理策略是整个系统的核心,也是影响整个系统性能的关键。为提高分布式代理中整个缓存模块所带来的系统性能,提出了一种优化的缓存数据放置策略(Optimized Cache data Allocation,OCA)。OCA 策略区别于传统的缓存数据放置策略,它能有效地将多个媒体数据段分布到多个代理上,目的是减少代理之间的数据传输代价,从而减少整个分布式系统中访问媒体数据的传输开销。通过理论分析与仿真实验表明,该策略能够减小数据的传输开销,提高系统的健壮性和可扩展性。

2 问题的提出

代理缓存技术的主要目标是降低网络传输的负载;分布式代理技术通过多个代理之间的相互协作,提供逻辑上更大的缓存空间,从而提高缓存字节命中率及服务的可扩展性。然而,传统的分布式代理技术未考虑缓存媒体的放置问题,即当一个用户通过本地代理发出请求时,本地代理将该请求转发到源服务器并获得相应数据。通常该代理将反馈的媒体数据保存到本地缓存以供本地客户端以及与其它代理连接的客户端再次访问时使用。当分布式系统中的其它代理访问该缓存媒体对象时,系统会认为远程命中,而直接通过该代理提供服务。其过程如图 2 所示。

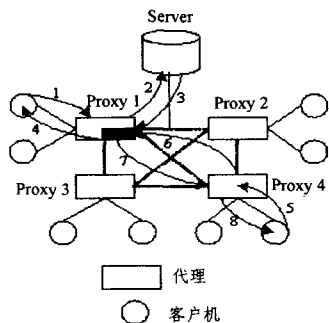


图 2 传统的分布式代理缓存协作过程

①客户端向代理 1 发出请求;②代理 1 转发请求给服务器;③代理 1 取回响应的媒体数据并将数据放置在本地缓存中;④将媒体数据反馈给客户端;⑤另一客户端向代理 4 发出同一媒体请求;⑥代理 4 通过缓存查找协议得知请求数据代理 1 命中,于是向代理 1 请求数据;⑦代理 1 将缓存中的媒体数据反馈给代理 4;⑧代理 4 转发反馈的数据给相应的客户端。

这种方案的缺点是没有考虑将获得的缓存数据如何优化地放置到整个分布式代理系统中,以减少缓存数据在各个代理之间的传输开销。在以往研究中,由于“代理-服务器”之间的网络传输开销要远大于“代理-代理”之间的开销,因此缓存数据在代理之间的传输开销容易被忽视。事实上随着系统中代理服务器数量以及整个缓存容量的不断扩大,“代理-代理”之间的传输开销是不容忽视的问题。文献^[5]中给出了分布式代理中在不同的缓存大小的情况下“代理-服务器”、“代理-

代理”和“代理-客户端”三者的网络传输开销的关系,如图 3 所示。

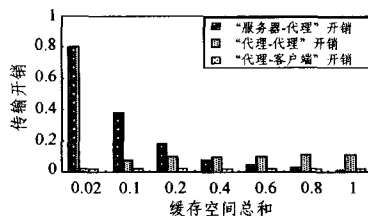


图 3 “代理-服务器”、“代理-代理”和“代理-客户端”三者的网络传输开销

图中看到,当整个代理缓存容量占服务器存储的媒体节目总容量的 40% 时,“代理-服务器”之间的开销已经小于“代理-代理”之间的开销值,也就是说,此时优化缓存数据在代理之间的传输开销已经是不容忽视的问题。同时可以得出,这种传统的缓存放置策略可扩展性并不好,因为随着代理数目的增加,总体缓存容量必然增大,当增加到 0.4 倍于服务器的媒体存储容量时,由于缓存命中率已经很高,继续减少“代理-服务器”之间的传输开销作用已不大。相反,若某些代理之间的网络传输状况并不好,代理数目的增加会使得整个系统的网络传输代价反而增加,因此需要考虑一种新型的缓存数据放置算法来减少数据在代理服务器之间的传输开销。

3 优化的缓存数据放置算法(OCA)

传统模式的分布式代理之间缓存数据的放置并不能相互协作。我们通过 P2P 灵活的协议,将 P2P 网络中数据的放置与定位功能引入进来,设计一套合理的缓存数据放置算法,有效地将多个媒体数据段分布到多个代理上,以减少代理之间的数据传输代价,从而减少整个分布式系统中访问媒体数据的传输开销。

3.1 缓存数据分段

在我们讨论的系统中,缓存的媒体数据分布在多个代理上,且假设每个代理的缓存空间有限,媒体数据仅能部分地缓存到代理上,只有源服务器上保存有完整的媒体节目内容。该系统中每个代理仅缓存媒体流的前缀部分,因此本文讨论的缓存放置实际上是前缀缓存的数据放置问题。至于前缀缓存的大小,我们将基于访问数据量的媒体流行度衡量标准引入进来,且流行度越高,前缀缓存的大小就越大。同时把前缀缓存分割成若干片断(Segments),基于分段的前缀缓存技术能够更有效地利用缓存空间^[6-8],缓存的后缀(Suffix)部分仍保留在服务器端。基于分段的前缀缓存示意图如图 4 所示。

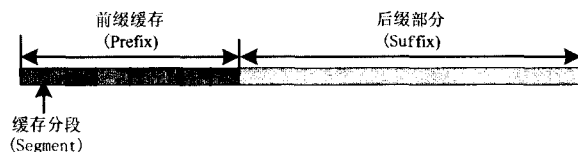


图 4 基于分段的前缀缓存

3.2 参数描述

在我们描述的基于 P2P 模式的分布式代理系统中,必须定义一套 P2P 的应用层协议,这些协议根据本文所提出的优化的缓存数据放置算法实现数据的分发、定位、查找和数据取回。该算法描述了在已知系统中媒体节目的访问模式、代理

之间以及代理与服务器之间的传输开销、给定代理的缓存磁盘空间等参数的情况下,如何将分段的前缀缓存数据分配到各个代理缓存上,使得用户访问媒体节目所需的传输开销总和最小。

假设系统中总共有 K 个代理,标记为 $P_i, i=1,2,\dots,K$ 。代理 P_i 连接有 W_i 个客户端。假设每个客户端有相同的媒体请求到达率 λ ,则通过代理 P_i 转发的媒体请求的到达率 λ_i 为 $\sum_{k=1}^{W_i} \lambda$,整个分布式代理系统的媒体请求到达率 λ_{total} 为 $\sum_{i=1}^K \lambda_i$ 。假设系统中源服务器上存储有 Q 个定码率的媒体节目。对于媒体节目 U_m 的回放时间长度为 T_m ,码率为 r_m , $m=1,2,\dots,Q$ 。节目 U_m 的大小为 $B_m=r_m * T_m$ 。由于已知对媒体节目的选择服从 Zipf 分布^[9],假定分布式系统中收到某个用户请求后,该请求访问节目 m 的概率为 μ^m ,而通过代理 P_i 访问该节目 m 的概率为 μ_i^m ,则有:

$$\mu_i^m = \mu^m \times \frac{\lambda_i}{\lambda_{total}} = \frac{\mu^m * W_i}{\sum_{j=1}^K W_j} \quad \text{s. t. : } \sum_{m=1}^Q \mu_i^m = 1 \quad (1)$$

假定代理 i 的缓存容量为 c_i ,整个系统的缓存容量为 C_P ,且整个系统的缓存容量应小于所有媒体节目所占的存储容量,即:

$$C_P = \sum_{i=1}^K c_i, \text{ s. t. : } C_P \leq \sum_{m=1}^Q B_m \quad (2)$$

假定代理与服务器端之间传输一个字节数据的传输开销为 $\pi^{S \rightarrow P}$,代理服务器 i 与代理服务器 j 之间传输一个字节数据的传输开销为 $\pi_{i,j}^{P \rightarrow P}$,代理 i 至它的客户机传输一个字节所需的传输开销为 $\pi_i^{P \rightarrow C}$ 。

定义媒体节目 m 的前缀缓存大小为 β_m ($\beta_m \leq B_m$),同时将节目的前缀缓存分成若干段,每一段为缓存的最小存储单元。假定节目 m 有若干段前缀缓存存储于代理 i 上,且占用代理 i 的缓存空间大小定义为 s_i^m ,则: $\beta_m = \sum_{i=1}^K s_i^m$ 。其中,各 s_i^m 的分配大小及在代理中的分布位置由本文下述的优化缓存放置算法(OCA)决定。相关参数定义如表 1 所列。

表 1 参数描述

参数	参数描述
K	系统中代理服务器的个数
W_i	代理 i 连接的客户机数目
λ_i	通过代理 i 发送请求的到达率
Q	源服务器上存储的媒体节目个数
m	媒体节目 m 的大小
μ^m	用户发出请求后访问节目 m 的概率
μ_i^m	代理 i 发出请求访问媒体节目 m 的概率
c_i	代理 i 的缓存容量
C_P	整个分布式代理系统的缓存容量
$\pi^{S \rightarrow P}$	代理与服务器之间传输一个字节数据的传输开销
$\pi_{i,j}^{P \rightarrow P}$	代理服务器 i 与代理服务器 j 之间传输一个字节数据的传输开销
$\pi_i^{P \rightarrow C}$	代理 i 至它的客户机传输一个字节数据的传输开销
β_m	媒体节目 m 的前缀缓存大小
s_i^m	媒体节目 m 分配在代理 i 上的前缀缓存大小

3.3 算法描述

对于各媒体的前缀缓存在各个代理上占用的缓存空间集合 $\{s_i^m\}$,优化的缓存数据放置算法描述为,在满足式(3)和式(4)条件下:

$$s_i^m \geq 0, i \in [1 \dots H] \quad (3)$$

$$\sum_{m=1}^Q s_i^m \leq c_i, \sum_{i=1}^K s_i^m \leq \beta_m \quad (4)$$

求得 $\min Cost(\{s_i^m\})$ 的值。其中 $Cost(\{s_i^m\})$ 为在给定的

一组 $\{s_i^m\}$ 集合下,用户请求媒体数据所需的总体网络传输开销函数。式(4)表示集合元素 s_i^m 的限制条件,其总和不得超过代理的缓存容量及媒体的前缀缓存的大小。

要求得最小化的 $Cost(\{s_i^m\})$ 的值,我们做如下分析:首先可以把系统中多个代理看成一个整体并有一个集成的缓存,把缓存在多个代理间的数据的传输开销看成这个集成的代理的内部开销;令 $C^m(\beta_m)$ 为媒体节目 m 的前缀缓存大小为 β_m 时,请求该媒体所需的传输开销;则 $Cost(\{s_i^m\})$ 的值可表达为 $\sum_{m=1}^Q C^m(\beta_m)$ 。当媒体未缓存而前缀缓存的大小为 0 时,传输开销定义为 $C^m(0)$ 。要使缓存后媒体的传输开销最小,该问题实际可看成与未缓存时相比所节省的开销最大化。即:

$$\text{MAX}(\sum_{m=1}^Q [C^m(0) - C^m(\beta_m)]) \quad (5)$$

我们可求得 $C^m(0)$ 与 $C^m(\beta_m)$ 的表达式:

$$C^m(0) = \lambda * \mu_m * B_m (\pi^{S \rightarrow P} + \pi^{P \rightarrow C}) \quad (6)$$

$$C^m(\beta_m) = \lambda * \mu_m * (\pi^{P \rightarrow C} \beta_m + (\pi^{S \rightarrow P} + \pi^{P \rightarrow C})(B_m - \beta_m) + \pi^m(\beta_m)) \quad (7)$$

式(6)与式(7)中 $\pi^{S \rightarrow P}$ 与 $\pi^{P \rightarrow C}$ 分别代表服务器到代理、代理到客户端之间传送一个字节数据所需的传输开销。 $\pi^m(\beta_m)$ 表示大小为 β_m 的前缀缓存数据在代理内部之间传输的开销。由于 $\pi^{S \rightarrow P}$, $\pi^{P \rightarrow C}$ 及 β_m 的值已由系统决定,对于前缀缓存的优化放置问题,实际上应考虑如何优化 $\pi^m(\beta_m)$,即优化前缀缓存数据在代理内部之间的传输开销。

对于给定的集合 $\{s_i^m\}$,定义 $\delta(m, i, s_i^m)$ 为大小为 s_i^m 的前缀缓存在代理内部之间传输所需的传输代价,则优化目标为:

$$\min(\sum_{m=1}^Q \sum_{i=1}^K \delta(m, i, s_i^m)) \quad (8)$$

$$\text{s. t. : } \sum_{i=1}^K s_i^m \leq \beta_m; \sum_{m=1}^Q s_i^m \leq c_i \quad (9)$$

而 $\delta(m, i, s_i^m)$ 可以通过式(10)得到:

$$\delta(m, i, s_i^m) = \sum_{j=1}^K s_i^m (\pi_{i,j}^{P \rightarrow P} + \pi_j^{P \rightarrow C}) \lambda_j \mu_j^m \quad (10)$$

定义 $\delta'(m, i)$ 为:

$$\delta'(m, i) = \sum_{j=1}^K (\pi_{i,j}^{P \rightarrow P} + \pi_j^{P \rightarrow C}) \lambda_j \mu_j^m \quad (11)$$

则缓存数据放置策略的优化目标可改写成 $\min(\sum_{m=1}^Q \sum_{i=1}^K \delta'(m, i) * s_i^m)$ 。另外,由于 $\delta'(m, i)$ 与 s_i^m 相互独立,且 $\delta'(m, i)$ 可看作代理 i 缓存的媒体节目 m 为一个字节数据时在代理内部之间的传输开销,因此最终优化目标为最小化 $\sum_{m=1}^Q \sum_{i=1}^K \delta'(m, i)$ 。对于如何得到 $\min(\sum_{m=1}^Q \sum_{i=1}^K \delta'(m, i))$ 的值的的问题,可以通过优化的贪婪算法得出最优解。

首先定义矩阵:

$$\begin{bmatrix} \delta'(1,1) & \delta'(1,2) & \dots & \dots & \delta'(1,K) \\ \delta'(2,1) & \delta'(2,2) & \dots & \dots & \delta'(2,K) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \delta'(m,i) & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta'(Q,1) & \delta'(Q,2) & \dots & \dots & \delta'(Q,K) \end{bmatrix}$$

根据式(11)计算出矩阵中每个元素的值。然后对于每个代理,计算它的 $\sum_{m=1}^Q \delta'(m, i)$ 的值,并按升序排序;对于每个媒体节目,计算它的 $\sum_{i=1}^K \delta'(m, i)$ 的值,并按降序排序;然后依次将

算得的开销值最小的代理尽可能多地缓存开销值为最大的媒体节目,同时满足式(9)所需的条件,从而得出最优解。该算法描述如下。

算法描述:optimal cache allocation algorithm(input:系统中各参数的值;output: $\delta'(m, i)$ 的排序结果)

1. 根据式(11)计算出 $\delta'(m, i)$ 每个元素的值;
2. 将 $\sum_{m=1}^K \delta'(m, i)$ 计算结果按升序排序;排序后的结果保存为链表,记为 $\text{Link}(i)$, $I_{\min} = \text{head}(\text{Link}(i))$;
3. 将 $\sum_{i=1}^K \delta'(m, i)$ 计算结果按降序排序;排序后的链表记为 $\text{Link}(m)$, $M_{\max} = \text{head}(\text{Link}(m))$;
4. 尽可能多地将媒体节目 M_{\max} 的前缀缓存段放置到代理 I_{\min} 中。
5. if 代理 I_{\min} 仍有缓存空间剩余 then $M_{\max} = \text{next}(\text{Link}(m), M_{\max})$;
else if 对媒体 M_{\max} 的缓存未全部完成 then $I_{\min} = \text{next}(\text{Link}(i), I_{\min})$;
end if
6. if 所有媒体的前缀缓存段分配完成 then 返回;
else goto step 4;
end if

4 性能分析

本节给出了优化的前缀缓存分布式放置策略的性能分析。研究的重点在于通过引入这种相互协作的代理缓存数据放置机制来减少对整个系统的网络传输负载的影响,同时还考察了该系统的健壮性和可扩展性。通过将该算法与传统的分布式代理缓存放置策略相比较,来衡量这种优化的缓存数据放置算法的性能结果。

在这组实验中,采用了泊松分布来仿真多个用户对系统的访问。实验环境参数设置如下:系统中服务器存储有 150 个播放时间为 30 分钟至 60 分钟的定码率的媒体对象,码率为 512kbps。媒体节目的长度服从均匀分布,平均长度为 45 分钟。由于假定这 150 个节目的流行度服从倾斜因子为 0.271 的 Zipf 分布^[10,11],因此预先已知每个节目的访问概率,即 μ^m 值。且假设我们所研究的是一个比较稳定的已知的分布式代理系统,即我们预先已知每个代理连接的客户端数、各代理之间的传输开销及代理至客户端的传输开销。也就是说我们可得出 $\pi_{ij}^{P \rightarrow P}$, $\pi_{ij}^{P \rightarrow C}$, λ_i , μ_i^m 这些参数的值,并根据式(11)算出 $\delta'(m, i)$ 的值,然后按照前文描述的 optimal cache allocation(OCA)算法将前缀缓存分段放置到各代理中。

4.1 减少传输开销

提出优化的缓存数据放置策略最主要的目的就是减少用户请求时媒体在网络中的传输开销。在该实验中测试了算法对减少网络传输的影响,并与传统的分布式代理缓存放置策略进行了比较。

实验假定系统中有 6 个代理,代理所连接的每个客户端有相同的请求到达率,假定为 20/分钟。则通过代理 i 向服务器发送请求的到达率 λ_i 为 $20 \times W_i$ 。每个代理有相同的 W_i 值,每个代理连接的客户端的数目为 15 个。

图 5 描述了随着整个代理的总体缓存容量增加时传输负载的变化曲线。在实验中,我们使用标准化后的度量值,网络负载以系统未设置缓存时为标准,总体缓存容量以整个媒体节目所占存储空间为标准。即当网络负载值为 1 时,表示代理缓存容量为 0 时传输开销的值;当代理缓存容量为 1 时,表

示代理缓存容量的总和与服务器存储的所有节目所占存储空间相等。很显然,随着整个缓存容量的增加,传输开销逐渐减少。当缓存容量小于 0.4 时,OCA 算法与传统分布式算法相差并不大。然而当缓存容量继续增大时 OCA 算法的传输开销基本上保持线性减少,而传统模式减小的幅度很小,这说明 OCA 算法更适合缓存容量比较大的情况。产生此结果的原因是当缓存容量小于 0.4 时,传输开销主要在服务器端与代理之间,两种算法都可以明显地降低服务器与代理之间的传输开销。当缓存容量大于 0.4 时,代理之间的传输开销对整个开销的影响显露出来,此时 OCA 算法能获得更小的传输开销。

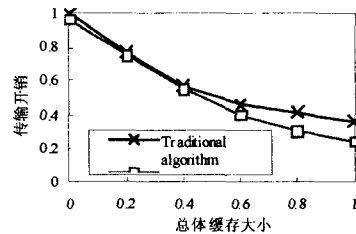


图 5 不同缓存大小对传输开销的影响

4.2 系统健壮性

基于对等网络(Peer to Peer, P2P)的流媒体系统是当前针对流媒体传输技术的另一研究方向^[12,13]。文献[14]对基于 P2P 网络的 Web 缓存对象的共享系统进行了研究,本文是针对流媒体对象的缓存系统。为了验证所提出的分布式代理服务器结构的优越性,对两种不同结构的服务健壮性(Robustness)作了比较。

实验中将 OCA 算法移植到客户端运行,将每个客户端请求得到的数据直接缓存到客户端的本地缓存,将代理服务器的缓存容量设为 0,并采用 OCA 算法进行缓存分配。这样该系统实际上就是一个基于客户端的 P2P 流媒体共享系统,因为当客户端请求的数据不在本地缓存时,OCA 算法将试图从其他客户端缓存中寻找数据,从而实现流媒体数据在客户端之间的共享。我们知道,在流媒体系统中客户端是不稳定的节点。其原因有可能是客户机关机,退出系统,或客户机失效,死机等等。实验中分别测试了分布式代理系统与基于客户端的 P2P 系统在不同的客户端失效概率情况下的传输开销。从图 6 可以看到,随着客户端失效概率的增加,分布式代理系统的传输开销基本保持不变,而基于客户端的 P2P 系统传输开销呈线性增加。当失效概率为 1 时,基于客户端的 P2P 系统的传输开销也为 1(即没有任何缓存时系统的开销值)。这是因为此时所有的客户端均失效,已没有客户端来提供缓存空间。实验结果说明了与基于客户端的 P2P 系统相比,分布式代理系统由于设置了一些专用于提供缓存服务的代理服务器,因此提高了其提供服务的健壮性。

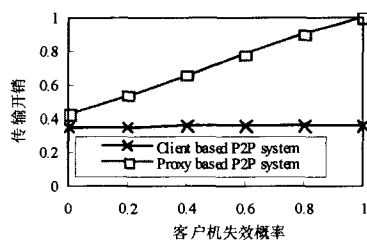


图 6 不同客户机失效概率下的传输开销(缓存大小:0.6)

4.3 可扩展性

为了验证所提出的系统的可扩展性,我们需研究该系统在代理数目及客户机数目不断增大时对数据传输开销的影响。图7给出了在不同的代理数目及每个代理所连接的客户机数目的情况下请求数据的总体传输开销。其中图7(a)为采用了OCA算法的系统的传输开销图,图7(b)为采用传统的分布式代理缓存放置算法的传输开销图。在这组实验中,可规定每个代理的缓存容量 c_i 为0.04(即缓存容量占服务器所存储媒体空间总和的4%),所有客户端的请求到达率均为20/minute。从图中可看到,随着代理数目的增加,两种算法的网络传输开销都会减少,而代理数目从1增加到8时,传输开销减少得十分明显,说明分布式的协作代理缓存技术是十分有必要的。随着客户机的增加,两种算法的传输开销都相应增大,这是由于客户端的增加会带来更多的媒体请求从而增大网络传输负载。另外,当代理数目增加到10以上时,若采用传统的代理缓存放置算法,增加代理数目对减少网络传输开销已不起作用。事实上,代理数目的增加可能反而会增加缓存数据在代理之间的传输开销。若采用优化的缓存数据放置算法,代理数目的增加可以使得整个系统的传输开销继续减少。从增加客户机数目这一变化趋势上看,传统的代理缓存放置算法比采用OCA算法的系统传输开销有更为明显的上升趋势。这说明了采用OCA算法的分布式代理系统有更好的可扩展性。

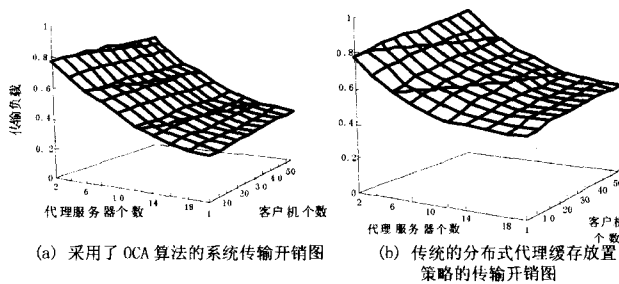


图7 采用不同的分布式缓存放置策略下的系统可扩展性图

结束语 本文提出了一种优化的缓存数据放置算法,其目的是使得在流媒体服务中,缓存数据在分布式代理之间以及代理与客户机之间的传输开销最小。理论分析及仿真实验表明,本文所提出的放置策略比未考虑优化的传统放置策略有更小的数据传输开销,并为分布式系统提供了更好的可扩展性。另外,我们验证了与当前的P2P流媒体传输技术相比,在网络节点不稳定的情况下,采用分布式代理服务器的流媒体传输技术有更高的服务可靠性。在今后的研究中,我们将本文的分布式代理缓存数据放置算法应用到实际流媒体网络中来测试命中率及数据传输开销。

参考文献

- [1] Acharya S. Techniques for improving multimedia communication over wide area networks[D]. Cornell University, NY, January 1999
- [2] Chen S, Shen B, Zhang X, et al. Designs of high quality streaming proxy systems[C]// Proc. IEEE INFOCOM. 2004; 1512-1521
- [3] 余江,刘威,王泰. 流媒体代理缓存技术研究[J]. 计算机科学, 2006,33(1)
- [4] Hoffmann M, Eugene T S Ng, Guo K, et al. Caching Techniques for Streaming Multimedia over the Internet[R]. Bell Laboratories, Carnegie Mellon University, April 1999
- [5] Alan TS I, Liu Jiangchuan, Lui J C S. COPACC: A Cooperative Proxy-Client Caching System for On-Demand Media Streaming [C]// Proc. of IFIP International Conference on Networking. Waterloo, ON, Canada, 2005; 1383-1387
- [6] Eager D L, Ferris M C, Vernon M K. Optimized Regional Caching for On-demand Data Delivery[C]// Proc. of IS&T/SPIE Multimedia Computing and Networking. San Jose, CA, Jan. 1999; 301-316
- [7] Jin S, Bestavros A, Iyenger A. Accelerating Internet streaming media delivery using network-aware partial caching[C]// Proc. of IEEE ICDCS'02. Vienna, Austria, July 2002; 153-160
- [8] Song Jia. Segment-based proxy caching for distributed cooperative media content servers[J]. ACM SIGOPS Operating Systems Review, 2005, 39(1): 22-33
- [9] Zipf G. Human Behavior and Principle of Least Effort; An Introduction to Human Ecology[M]. USA Massachusetts: Addison Wesley press, Cambridge, 1949
- [10] Dan A, Sitaram D, Shahabuddin P. Scheduling Policies for an On-Demand Video Server with Batching[C]// Proc. of the ACM Multimedia. New York: ACM Press, 1998; 391-398
- [11] Aggarwal B, Wolf J L, Yu P S. On Optimal Batching Policies for Video-on-Demand Storage Servers[C]// Proc. of the 1996 Intl. Conf. on Multimedia Computing and Systems. Hiroshima, Japan, 1996; 253-258
- [12] Zhang M, Xiong Yongqiang, Zhang Q, et al. Optimizing the Throughput of Data-Driven Peer-to-Peer Streaming[J]. IEEE Transactions on Parallel and Distributed Systems, 2009, 20(1): 97-110
- [13] 刘亚杰, 窦文华. 一种P2P环境下的VoD流媒体服务体系[J]. 软件学报, 2006, 17(4): 876-884
- [14] 凌波, 王晓宇, 周傲英, 等. 一种基于Peer-to-Peer的Web缓存共享系统研究[J]. 计算机学报, 2005, 28(2): 107-178