基于缓存窗口和段补丁预取的流媒体动态调度算法

杨　戈[①②]　朱晓民[①]　廖建新[①]　黄　海[①]
[①](北京邮电大学网络与交换技术国家重点实验室　北京　100876)
[②](辽宁大学信息科学与技术学院　沈阳　110036)

摘　要：该文提出了一种新的基于缓存窗口和段补丁预取的移动流媒体动态调度算法，采用代理缓存窗口自适应伸缩和分段缓存补丁块方案，实现了移动流媒体对象在代理服务器中缓存的数据量和其流行度成正比的原则。仿真结果表明，对于客户请求到达速率的变化，该算法比传统算法具有更好的适应性，在最大缓存空间相同的情况下，能显著减少通过补丁通道传输的补丁数据，从而降低了服务器和骨干网络带宽的使用，能快速缓存媒体对象到缓存窗口，同时减少了代理服务器的缓存平均占有量。

关键词：移动流媒体；调度算法；代理缓存；段补丁预取
中图分类号：TP393　　**文献标识码**：A　　**文章编号**：1009-5890(2007)05-1198-04

A Dynamic Scheduling Algorithm for Streaming Media Based on the Cache Window and Segment Patch Pre-fetching

Yang Ge[①②]　Zhu Xiao-min[①]　Liao Jian-xin[①]　Huang Hai[①]
[①](*State Key Laboratory of Networking and Switching Technology*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)
[②](*College of Information Science and Technology*, *Liaoning University*, *Shenyang* 110036, *China*)

Abstract: A novel dynamic scheduling algorithm for mobile streaming media based on the cache window and segment patch pre-fetching is proposed. It employs the scheme that the cache window size can be increased or decreased adaptively according to the popularity of the requested object and the patch bytes that are segmented and cached. The principle is obeyed that the data cached for each mobile streaming media object are in proportion to their popularity at the proxy server. Simulation results show that this strategy is more adaptive than conventional algorithm for the variety of the request arrival rates. It can significantly reduce patching data through patching channel under the circumstance of the same maximum cache space. This can effectively reduce the server load and network bandwidth usage on backbone link. It can faster cache the media object than P·S·A algorithm. It reduces the average occupied cache space at the proxy.

Key words: Mobile streaming media; Scheduling algorithm; Proxy caching; Segment patch pre-fetching

1　引言

在移动通信网上，流媒体正受到越来越多的重视，在大规模的流媒体系统中，用户的点播往往集中于少数热门节目，这就使得合并用户服务、共享服务器和网络带宽等资源成为可能，于是流调度技术应运而生。

OBP(Optimized Batch Patching)+prefix&patch caching[1,2]算法将补丁数据也进行分段缓存，让不同批处理区间到达的客户可以共享一部分补丁块，但当客户对“热门”对象的访问请求强度很高时候，这个算法仍然需要消耗较高的骨干网络带宽。文献[3]提出的 P^3S^2A (Proxy-assisted Patch Prefetching and Service Scheduling Algorithm)调度算法，根据当前客户请求到达的分布状况，代理服务器为后续到达的客户请求进行补丁预取和缓存，但它对每个流媒体对象都进行缓存，对那些很少被访问的媒体对象进行全部或者部分缓存都将造成代理服务器缓存效率的下降，P^3S^2A算法将客户对媒体对象的访问等同起来，只是考虑客户对媒体对象的访问频率，当客户对媒体对象的访问时间是不同时，没有区别对待，另外对于较流行的流媒体对象缓存速度不够快，为了避免这些情况，增加较流行媒体对象的缓存空间，更好地区分不同流行度的流媒体对象，本文提出基于缓存窗口和段补丁预取的移动流媒体动态调度算法$DS·AM^2CWP^2$ (Dynamic Scheduling Algorithm for Mobile streaming Media based on the Cache Window and segment Patch Pre-fetching)。

2006-04-14 收到，2006-10-26 改回
国家杰出青年科学基金(60525110)，新世纪优秀人才支持计划(NCET-04-0111)，高等学校博士学科点专项科研基金资助课题(20030013006)，电子信息产业发展基金(基于 3G 的移动业务应用系统)和电子信息产业发展基金重点项目(下一代网络核心业务平台)资助课题

2 基于缓存窗口和段补丁预取的调度算法(DS3AM2 CWP2)

2.1 自适应窗口和段补丁预取

缓存在代理中的每个流媒体对象都要建立和保存一个叫媒体对象访问日志的数据结构，具体内容如下：T_{first} 为媒体对象第 1 次被访问的时间；T_{last} 为媒体对象最近一次被访问的时间，当媒体对象第 1 次被访问时，$T_{\text{first}}=T_{\text{last}}$；$T_{\text{sum}}$ 为媒体对象被访问的总长度(时间长度表示)；N为媒体对象被访问的次数；L_S 为每段段长(时间长度表示)，T为媒体对象总长度(时间长度表示)；B为每个数据块表示传输的最小数据单位(时间长度表示)；b为批处理间隔(时间长度表示) $b=mB(m=1,2,\cdots)$；P为媒体对象前缀部分(时间长度表示)。

在某时刻 t，$L_{\text{avg}}=T_{\text{sum}}/N$ 表示该时刻统计的每次平均访问媒体对象的长度(时间长度)，L_{avg} 可以隔一段时间进行统计一次，比如每隔 $n\bullet T$ 时间 $(n=1,2,\cdots)$ 统计 1 次。

设 $T=k\bullet b$，缓存窗口大小为 W，$W=m\bullet b(m=1,2,\cdots,k)$，$m$ 的初始值为 $\left\lfloor\frac{k}{2}\right\rfloor$，在代理中每次缓存补丁预取是以相应的数据段大小为单位，每个数据段包含 N_s 个 b，$L_s=N_s\bullet b$，设

$$N_s=\begin{cases}(i-1)i^{\left\lfloor\frac{s}{n+1}\right\rfloor-1}, & s>n\\ 1, & s=0,1,\cdots,n\end{cases}$$

本文中取 i=2，n=1，如图 1，则 $N_s=\begin{cases}2^{\left\lfloor\frac{s}{2}\right\rfloor-1}, & s\geq 2\\ 1, & s=0,1\end{cases}$

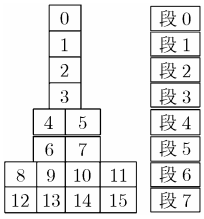

图 1 代理服务器中补丁预取的数据段

2.2 调度算法(DS^{3}AM^{2}CWP^2)

DS^{3}AM^{2}CWP^2 算法的基本思想是利用代理服务器在客户请求转交常规流数据时进行补丁预取并缓存，在缓存窗口 W 内，是否进行补丁预取取决于当前批处理区间内是否有客户请求到达以及代理是否已经缓存到 W，当客户播放完在代理缓存的媒体对象而且代理没有缓存到 W，这时要从源服务器中提取相应补丁块，每次代理服务器预取的补丁块的数量根据预先定义的数据段大小来定，当客户请求到达速率较高时，补丁预存能够很快达到缓存窗口，最大化地利用缓存，可以明显降低通过补丁通道从源服务器提取的补丁数量，从而降低骨干网络负担及源服务器并发流的个数。

调度算法具体过程：

(1) 假设第 1 个客户请求在 t_0 时刻到达，如图 2，这时代理中只有缓存客户请求对象的前缀部分，对于第 1 个客户和到达时刻 $t\in[t_0,t_1]$ 的客户请求如 A，代理服务器立即通过单播向每个客户传送媒体对象前缀部分 b，代理服务器将在 t_1 时刻请求源服务器通过单播信道开始传输常规流 T-b，同时代理开始预先分配一个长度为 L_0 的缓存空间，用于缓存即将到达的常规流的第一个数据段$[b,2b]$作为区间 $[t_1,t_2]$ 到达客户请求的补丁块，实现补丁预取，常规流数据到达代理后，代理通过组播通道向客户转交。

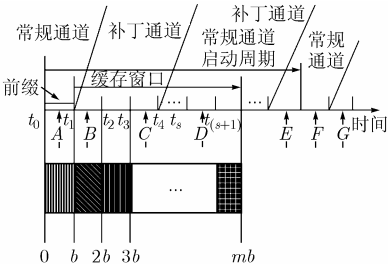

图 2 DS^{3}AM^{2}CWP^{2}算法示意图

(2) 当到达时刻 $t\in[t_1,t_2]$ 的客户请求如 B，代理服务同样立即通过单播向每个客户传送媒体对象前缀部分 b，代理服务器在 t_2 时刻开始预先分配一个长度为 L_1 的缓存空间，用于缓存$[2b,\ 3b]$，作为下一个区间 $[t_2,t_3]$ 内到达的客户请求的补丁，同时代理服务器在 t_2 时刻通过补丁通道向客户传输补丁数据，通过组播向客户转交常规流，系统运行到 t_3 时刻，代理服务器已经缓存 2 个长度 L_0+L_1 的补丁段。

(3) 同理，若在整个缓存窗口中没有出现空区间，当到达时刻 $t\in[t_S,t_{S+1}]$ 的客户请求如 D，代理服务同样立即通过单播向每个客户传送媒体对象前缀部分 b，代理服务在 t_{s+1} 时刻通过补丁通道向客户传输补丁数据，代理服务器在 $1+N_0+N_1+N_2+N_3+\cdots+2^{\left\lfloor\frac{s-1}{2}\right\rfloor-1}$ 时刻开始预分配一个长度为 L_s 的缓存空间，用于缓存 $\left[\left(1+N_0+N_1+N_2+N_3+\cdots+2^{\left\lfloor\frac{s-1}{2}\right\rfloor-1}\right)b,\ \left(1+N_0+N_1+N_2+N_3+\cdots+2^{\left\lfloor\frac{s-1}{2}\right\rfloor-1}+2^{\left\lfloor\frac{s}{2}\right\rfloor-1}\right)b\right]$，作为以后区间到达的客户请求的补丁，客户需要在 t_{s+1} 时刻加入补丁通道，在 $t_{\left(1+N_0+N_1+N_2+N_3+\cdots+2^{\left\lfloor\frac{s-1}{2}\right\rfloor-1}\right)}$ 时刻加入常规通道。

(4) 若在某个时间间隔内没有客户请求到达，则该区间为空区间，这时代理暂停分配缓存空间，同时缓存窗口减小一个 b。此后，如果在后续的时间间隔内有客户请求到达，则代理根据实际情况决定分配缓存的大小以及是否向服务器要求补丁服务和加入常规通道的时间，具体见第 3 节算法分析。

(5) 若客户到达的区间前面有多个连续空区间，则代理

有可能缺失一定的补丁数据，如在$[t_6, t_7]$区间到达的客户前面连续有两个空区间，则代理在t_7时刻需要分配 3 个数据块缓存$[5b, 7b]$，$[7b, 9b]$，$[9b, 13b]$，其中$[5b, 7b]$的数据块需要代理通过一个单播通道从源服务器中重新获取，称为补丁服务，$[t_6, t_7]$到达的客户请求将在t_7时间加入常规流中。

(6) 重复上面(3)–(6)步，直到缓存窗口边界。经过上面(1)–(7)步后，如果在代理中已经缓存媒体对象的一部分，大小等于前面一个服务周期结束时缓存窗口的最终长度，后来的客户请求将开始一个新的服务周期，首先由代理向客户提供服务，在服务到缓存窗口，需要代理通过一个单播通道从源服务器中获取，媒体对象剩余部分(T-W)。

(7) 经过一段时间，本文取经过该媒体对象的播放时间，可设代理缓 $W=\left(\left\lfloor\frac{\text{Lavg}}{b}\right\rfloor+1\right)b$ 。

3 DS³AM²CWP²算法分析

为了分析简单，假设到达的客户其w_1==2，这时通过补丁通道传输的补丁数量要比标准 DS³AM²CWP² 算法多，设这种情况是伪 DS³AM²CWP² 算法。为了便于比较，在本试验中，假设在每个时间间隔中都有客户，媒体对象播放的持续时间长度 T=90min，前缀长度 P=1min，批处理间隔 b=1min，常规通道启动周期为 W+P，媒体播放速率为 r=1.5Mbps (MPEG-1)，u是需要启动补丁通道重传的部分如图 3，R表示源服务器输出链路的平均传输带宽(即骨干链路的平均传输速率)，骨干链路的归一化带宽(服务器平均并发流个数)为 $R/r=[u+(T-P)\}/(W+P+1/\lambda))$如图 4，代理服务器的缓存平均占用量为 S，即 $S=P\bullet r+(u+u_1)\bullet r$，如图 5，其中$u_1$表示从常规通道中预取的补丁，$u_1=b+(W-b)\bullet(1-p)$ 。

图 3 中(a)，(b)分别显示 OBP+prefix&patch caching 算法，P³S²A 算法和伪 DS³AM²CWP² 算法在不同客户请求强度下通过补丁通道传输补丁数量的比较，前两种算法都随 W 变大而增加，OBP+prefix&patch caching 算法增加更快。变化最慢的是伪 DS³AM²CWP² 算法。当 $W\le 3b$ 时，DS³AM²CWP² 算法等同于 P³S²A 算法，这时无论 λ 如何变化，这两个算法要传输的补丁数据量都相同。当 $W>3b$ 时，伪 DS³AM²CWP² 算法都是 3 种算法中传输最少的补丁数据，当 $\lambda\ge 8$ 时，P³S²A 算法和伪 DS³AM²CWP² 算法都趋于稳定，需要传输非常少的补丁数据，OBP+prefix&patch caching 算法变化不大。同理标准 DS³AM²CWP² 算法也满足这个趋势。

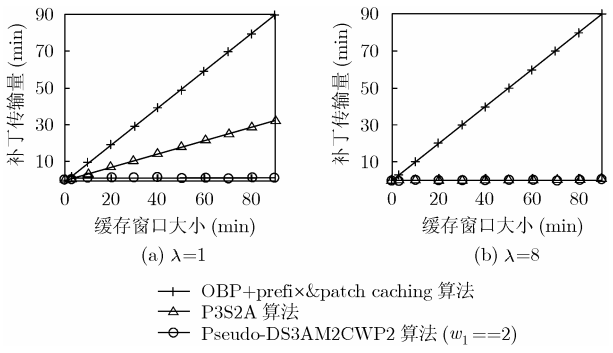

图 3 补丁传输量和窗口 W 的关系

图 4 中(a)，(b)分别显示 OBP+prefix&patch caching 算法，P³S²A 算法和伪 DS³AM²CWP² 算法在不同客户请求强度下消耗的骨干链路带宽，3 种算法的骨干链路消耗的带宽随缓存窗口增大而减少，显然伪 DS³AM²CWP² 算法减少的更快，OBP+prefix&patch caching 算法减少的最慢。同理标准 DS³AM²CWP² 算法也满足这个趋势。

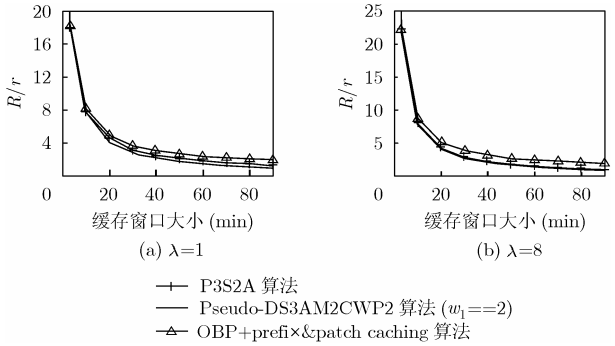

图 4 骨干链路归一化带宽和窗口 W 关系

图 5(a)，(b)分别是 3 种算法在不同客户请求强度下，所需要的代理平均缓存空间的对比，平均缓存空间随 W 线性增长，伪 DS³AM²CWP² 算法增长的最慢，OBP+ prefix&patch caching 算法和 P³S²A 算法比较接近，当 λ =8 时，3 者已经基本重合。因为标准 DS³AM²CWP² 算法比伪 DS³AM²CWP² 算法的 u 小，而在同样条件下 u_1 相同，所以标准 DS³AM²CWP² 算法必然在伪 DS³AM²CWP² 算法曲线下面。

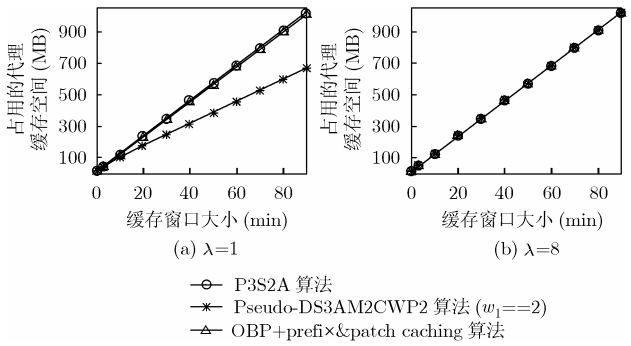

图 5 代理服务器的缓存平均占有量的对比

4 结束语

本文根据用户对流媒体对象的访问行为，提出了 DS³AM²CWP² 调度算法，每次分配不同大小的缓存段，实现基于常规流的补丁预取和缓存，同时隔一段时间，更新一次缓存窗口大小，根据媒体流行度动态决定其最大缓存大小，尽量拖延了加入常规流的时间，这样不但降低了补丁块传送的数量，从而降低了骨干网络带宽的消耗和流媒体服务

器的负担，而且在少量客户请求时，能够节省代理缓存。实验表明，$DS^3AM^2CWP^2$ 算法比 P^3S^2A 算法和 OBP+prefix&patch caching 算法具有更好的性能，但是在客户请求相对较少的情况下，$DS^3AM^2CWP^2$ 算法有可能在某一时刻比 P^3S^2A 算法和 OBP+prefix&patch caching 算法占有多一些的代理缓存，但经过一段时间的缓存窗口自动调整，这种情况会有很大改善，考虑到目前的缓存成本，$DS^3AM^2CWP^2$ 算法还是具有非常大的实际用途。另外，移动网络用户也非常需要流媒体系统提供 VCR-like 功能，下一步是如何在 $DS^3AM^2CWP^2$ 算法的基础上实现 VCR-like 功能。

参 考 文 献

[1] Verscheure O, Venkatramani C, Frossard P, and Amini L. Joint server scheduling and proxy caching for video delivery[J]. *Computer Communications*, 2002, 25(4): 413–423.

[2] Frossard P and Verscheure O. Batch patch caching for streaming media[J]. *IEEE Communications Letters*, 2002, 6(4): 159–161.

[3] 覃少华，李子木，蔡青松，胡建平. 基于代理缓存的流媒体动态调度算法研究[J]. 计算机学报，2005, 28(2): 185–194.
Qin Shao-hua, Li Zi-mu, and Cai Qing-song, and Hu Jian-ping. Study on dynamic scheduling algorithms for streaming media based on proxy caching[J]. *Chinese Journal of Computers*, 2005, 28(2): 185–194.

杨　戈：　男，1974 年生，讲师，博士生，主要研究方向为下一代网络、流媒体.

朱晓民：　男，1974 年生，副研究员，博士，主要研究方向为智能网、下一代网络、3G 核心网.

廖建新：　男，1965 年生，教授，博士生导师，主要研究方向为下一代网络、智能网.

黄　海：　男，1979 年生，博士生，研究方向为下一代网络.