支持VCR功能的扩展流合并算法VCRSM的设计与实现*)

单 炜 叶保留 戴 菡 陆桑璐 陈道蓄
(南京大学计算机系 南京210093)

摘 要 多播传输作为视频点播服务的一个有效方法在近十年来被广泛研究,许多算法都相对成熟。但由于VCR操作会使多播调度的复杂度增加,性能降低,因此对支持用户VCR操作的多播调度算法的研究很少。针对该问题,本文提出了一个支持VCR功能的零时延流合并调度算法VCRSM。VCRSM通过在客户端使用主动缓存技术满足部分VCR操作,并在服务器端对普通流合并算法进行扩展,使之能响应VCR请求流,并通过延迟请求提高流的共享度,优化服务器的性能。模拟实验证实,VCRSM能在不占用更多带宽的情况下,零延迟地满足用户的各种VCR操作。

关键词 多播,VCR操作,客户端主动缓存,扩展流合并算法

Design and Implementation: An Algorithm Supporting VCR Functions by Extended Steam Merging

SHAN Wei YE Bao-Liu DAI Han LU Sang-Lu CHEN Dao-Xu
(Department of Computer Sience and Technology, Nanjing University, Nanjing 210093)

Abstract As an effective means of VOD service, multicast has been researched for about ten years. Many algorithms of it have been matured. But there are only a few papers about multicast schedule supporting VCR functions. The reason is that supporting VCR functions will increase the complexity of schedule of the server, and will decrease its capability. In this paper, we present a multicast scheduling algorithm VCRSM provding VCR functions by extended stream merging. Active buffer management on client is used to partly satisfy the VCR actions of users. On server, original stream merging is modified to adapt to all VCR requests. We also put forward delay request to optimize the server's performance. When compared with other systems in simulation, VCRSM can satisfy all kinds of VCR functions with no delay, while occupying no more bandwidth than others.

Keywords Multicast, VCR actions, Active buffer management, Extended stream merging

1 引言

随着宽带网络的普及,人们对具有实时动态交互功能的视频服务的应用需求有增无减。同时,实时个性化视频服务进一步增强了对系统的带宽需求。因此,如何在有限带宽(服务器及网络带宽)资源条件下为更多客户提供个性化服务仍然是视频服务技术研究的一个主要难点和重点。

多播(multicast)传输为提高带宽利用率、提升系统服务能力提供了可行途径。近10年来,一些研究[1~9]试图在服务器端建立有效的多播调度算法来提高服务器端的带宽使用率,通过一定的调度策略来同步处理不同用户间的服务请求,实现流共享,但这些算法都假设影片播放过程中用户不会发出交互操作(如暂停、跳转、快进等)。由于用户VCR请求的异构性和异步性使服务器端对流的同步调度和维护变得异常复杂,许多服务商都使用单播技术来支持VCR操作。该方式对带宽要求随用户数线性增长,因而大大限制了系统的服务能力和可伸缩性。近年来,也有一些针对支持VCR功能的多播技术的研究[10~16],这些工作在现有多播调度算法基础上,借助客户端缓存或服务器端应急通道来支持VCR请求。但若完全依赖客户缓存,只能部分满足VCR请求,如小范围的跳转或快进;而在服务器端为每个VCR操作开辟应急通道,则会减少多播服务的可用带宽,牺牲服务器性能。

针对上述缺点,本文以分层多播流合并算法HMSM(Hierarchical Multicast Stream Merging)[8]为基础,通过将客户缓存和多播调度技术相结合,提出一种基于流合并技术扩展的VCR调度算法——VCRSM,使系统在保证实时满足用户VCR请求的同时,尽可能实现流合并和流共享,从而提高系统的服务能力。VCRSM的主要特点在于:首先,在客户端将主动缓存技术和VCR请求处理相结合,使客户缓存能满足部分VCR操作请求,并建立延迟请求机制优化服务器性能;其次,在服务器端,以多播方式发送所有数据,并在播放过程中将邻近流逐步合并,进而最大限度地实现流共享。

本文的内容组织如下:第2节介绍了相关研究,包括基本多播调度算法及现有支持VCR的多播调度算法;第3节详细描述了VCRSM算法的设计与实现;第4节对VCRSM进行了性能分析;最后对本文工作进行了总结。

2 相关研究

单播VOD系统的带宽需求随用户数的增加而线性增长,缺乏可扩展性。而研究表明,特定时段内大多数用户请求集中在少部分影片上,影片的访问率满足Zipf分布[4]。因此,可利用多播技术来缓解带宽瓶颈问题。多播算法可分为两类:

*)基金项目:国家高技术研究发展计划863项目(No.2001AA113050),国家重点基础研究发展规划973项目(No.2002CB312002)。单 炜 硕士研究生,研究方向:分布计算与并行计算;叶保留 博士研究生,研究方向:分布计算与并行计算;戴 菡 硕士研究生,研究方向:分布计算与并行计算。陆桑璐 教授,研究方向:分布计算与并行计算;陈道蓄 教授,博士生导师,研究方向:分布计算与并行计算。

一种是阶段性广播，另一种是调度多播。阶段性广播也称开环(open-loop)算法，由服务器为每个影片分配若干通道，分别循环播放该影片或其片段，其特点是服务器带宽消耗恒定，与请求到达率λ无关，扩展性很好。但在λ较小时，影片仍循环广播，因此对热门影片较为适用。典型的调度多播算法有金字塔广播(Pyramid Broadcast)[1]和摩天大楼广播(Skyscraper Broadcast)[2]。调度多播也称闭环(closed-loop)算法，数据发送由客户请求驱动。当且仅当用户提出影片服务请求时，服务器才会分配通道，并采取一定策略实现流共享。批处理方法[4]、动态摩天楼算法[3]，以及补丁流方式[5,6]都是调度多播。Mcache(Multicast with Cache)[7]将批处理技术引入补丁流算法。Derck Eager 等人在文[8]中分析了零延时服务模式下服务器带宽的理论下限，并集成动态摩天大楼、补丁等模式的优点，提出性能较为接近该理论下限的 HMSM 算法。HMSM 的主要思想将在本文3.1节介绍。

由于上述多播方式缺乏对服务交互性和自主性的支持，一些研究采用客户端缓存技术或应急通道技术对基本多播调度算法进行调整，提出支持 VCR 的多播算法。文[10,11,13,14]采用客户端缓存技术对开环算法进行扩展，试图在不增加服务器带宽消耗的条件下支持 VCR 功能。文[10,11]让缓存决定数据接收源，以保证播放点尽量位于缓存中间，从而更好地满足用户的 VCR 请求。文[13]基于开环广播摩天大楼算法，但对影片分段方法作了改进，使更适合 VCR 操作需求。文[14]借助缓存技术及播放速率调整方法支持 VCR 功能，但仅考虑了快进操作。完全依赖客户端缓存的 VCR 操作受限于缓存大小及其内容。文[12,15,16]在系统中引入应急通道技术。文[12]在客户缓存不能满足 VCR 请求时，向服务器发送数据请求，服务器尽量将 VCR 调度到现存流中，并在失败时启动一个新的应急流。文[15,16]都基于批处理模式，在 SAM 协议[15]中，一旦客户发出 VCR 请求，服务器将此客户从组播流分离，并专门调度一个单播流来满足 VCR 交互服务，VCR 操作结束后，尽量把该用户重新合并到其它组播流。当服务器带宽满载而不能处理 VCR 请求时，请求将被延时，直到有满足该 VCR 操作的可用带宽。由于大量单播应急流的出现会严重影响系统的可扩展性，文[16]对 SAM 协议作了改进，将服务器总带宽按一定比例分配给组播流和交互流，避免两者相互影响而导致客户 QoS 降低。与本文工作相比，上述各算法都还存在初始响应延时。

3 扩展流合并算法 VCRSM 设计

3.1 基本流合并算法

分层多播流合并算法 HMSM[8]的主要思想是：在服务器端，所有流都以多播形式发送，用户都可随时加入任一现存流。为追赶在其之前的某个播放流(称为目标流)，用户在播放过程中最多可同时接收两个流。一旦追赶上当前目标流，将继续追赶该目标流的目标流，直至最终追赶上根流。追赶过程中，一旦两流合并，它们的客户从该时刻开始将同步接收数据。

文[9]针对如何确定目标流给出了一些 HMSM 变体算法，如 ERMT，CT，SRMT。其中 ERMT 算法更接近最优化，但其计算复杂度高，且要求所有现有流的未来状态都确定，不会再产生其他新流，而实际的 VCR 操作中，该条件很难满足。SRMT 和 CT 的计算较为简单，而且实验结果也不与 ERMT 存在明显差异。考虑到 SRMT 中的流更为稳定，目标流重定位的次数比 CT 少，因此，本文选择 SRMT 作为 VCRSM 在服务器端的基本流合并算法。

根据 SRMT 算法，服务器端维护系统内当前存在的所有多播流，并按开始时间排序。每当新增一个流 A 或有其它流与流 A 合并时，流 A 需要重新定位其目标流。定位时，流 A 由近至远遍历早期流，直到找到一个符合一定条件的早期流 P。为追上流 P，流 A 必须发送与流 P 当前播放点之间的缺失数据，且流 P 必须晚于流 A 结束。

以图1为例，客户 A,B,C,D 分别在 T_0，$T_{0.15}$，$T_{0.25}$，$T_{0.35}$ 时刻向同一部影片提出服务请求。服务器在上述时刻为每个客户请求分别调度一个组播流 S_A、S_B、S_C、S_D。根据 SRMT，客户 B 在接收 S_B 的同时，还接收 S_A，在 $T_{0.3}$ 时刻两流合并，此时 S_A 结束。图中虚线表示合并路线。当客户 C 到达时，A 正在接收数据 $D_{0.25}$，而 B 正在接收 $D_{0.1}$。此时，如果 C 同时接收 S_B，那么两流至少要在 $T_{0.35}$ 才能合并，而那时 S_B 已与 S_A 合并。因此，C 直接接收 S_A。如果没有其他后续客户，S_C 在 $T_{0.5}$ 与 S_A 合并。随后客户 D 到达时，S_D 将以 S_C 为目标流，并在 $T_{0.45}$ 合并。在追赶期间，客户 C 已经从 S_A 接收到 $D_{0.25}$～$D_{0.45}$，而合并时客户 D 并未接收到。由于此时 SA 的数据位置已到 $D_{0.45}$，因此，S_C 必须重新定位其目标流，使客户 D 接收到 $D_{0.25}$～$D_{0.45}$。

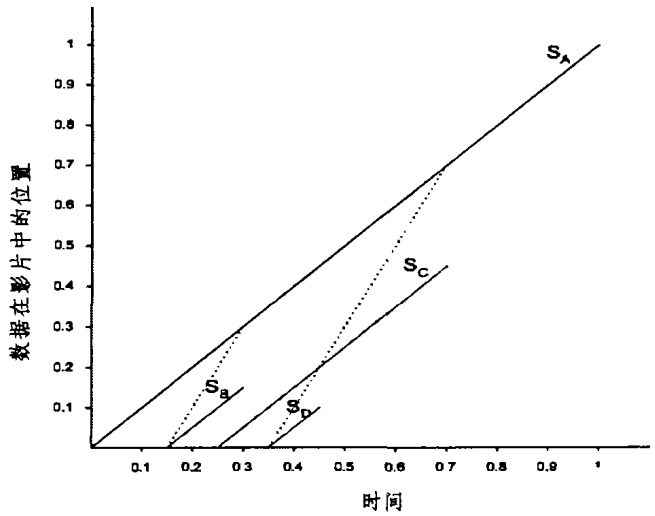


图1 基本流合并算法 SRMT

由于基本流合并算法只考虑了一个流从影片起始无间断播放的情况，因此不能提供 VCR 操作的响应。

3.2 VCR 操作定义

影片可看成是按一定时序关系排列的一系列数据帧。为便于描述，本文将客户当前正在观看的帧在影片中的位置称为播放点。VCR 操作可用二元组$(\Delta t,\Delta l)$描述，其中 $\Delta t\geqslant 0$，指 VCR 操作的持续时间，Δl 指 VCR 操作后播放点(称为目标点)与 VCR 操作前播放点(称为起始点)之间的相对距离。如果 $\Delta l>0$，则从起始点至目标点的移动方向与影片正常播放方向相同；如果 $\Delta l<0$，则相反。设视频文件的正常播放速率恒定为 b，则各种 VCR 操作可按如下定义：

1. 向前跳转/向后跳转：$\Delta t=0,\Delta l\neq 0$。
2. 快进/快退：$\Delta t>0,|\Delta l|/\Delta t>b$。
3. 慢进/慢退：$\Delta t>0,|\Delta l|/\Delta t<b$。
4. 暂停：$\Delta t>0,\Delta l=0$。
5. 正常播放：$\Delta t>0,\Delta l/\Delta t=b$；倒放：$\Delta t>0,\Delta l/\Delta t=-b$。

为便于分析，假设系统中每个客户的带宽都大于等于$2b$，快进和快退操作的加速因子为2，即速率为$2b$。4.3节将讨论上述条件不成立时的情形。

根据上述定义，在基于VCRSM算法的VOD的系统中，服务器端的流包括四种：1)正常流：速率为b的流；2)倒放流：速率为$-b$；3)快进流：速率为$2b$；4)快退流：速率为$-2b$。其中，倒放流和快退流统称为回退流，倒放流、快进流和快速流都属于特殊流。

3.3 客户端主动缓存

在不支持VCR操作的多播系统中，客户端缓存主要用于缓存"未来"数据。事实上，客户缓存数据还可在一定程度上满足VCR请求。VCRSM引入客户端主动缓存技术，使客户缓存不仅缓存未来数据，还可根据需要缓存已播放数据，以便为快进、倒退等VCR操作服务。主动缓存还可以根据当前缓存情况自行决定是否继续接收流及是否向服务器发送数据请求(包括立即请求和延迟请求)。

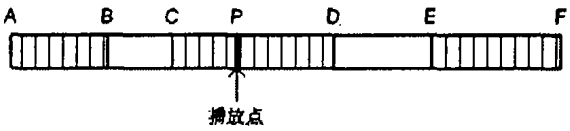


图2 缓存中数据示例

VCRSM中的客户端缓存可以看成是一个滑动窗口，其中数据按在影片中的先后顺序从左向右存放，如图2所示。缓存数据不一定连续，图2阴影部分表示缓存中已有数据(如A～B)，而空白部分表示未缓存帧(如B～D)，称为断点。在从播放点向右(左)搜索过程中遇到的断点的起始位置称为断点起始(如C,D)。若没有遇到断点，则将窗口最右(左)端当作断点起始。图3显示了各种VCR操作过程中缓存数据的变化情况。

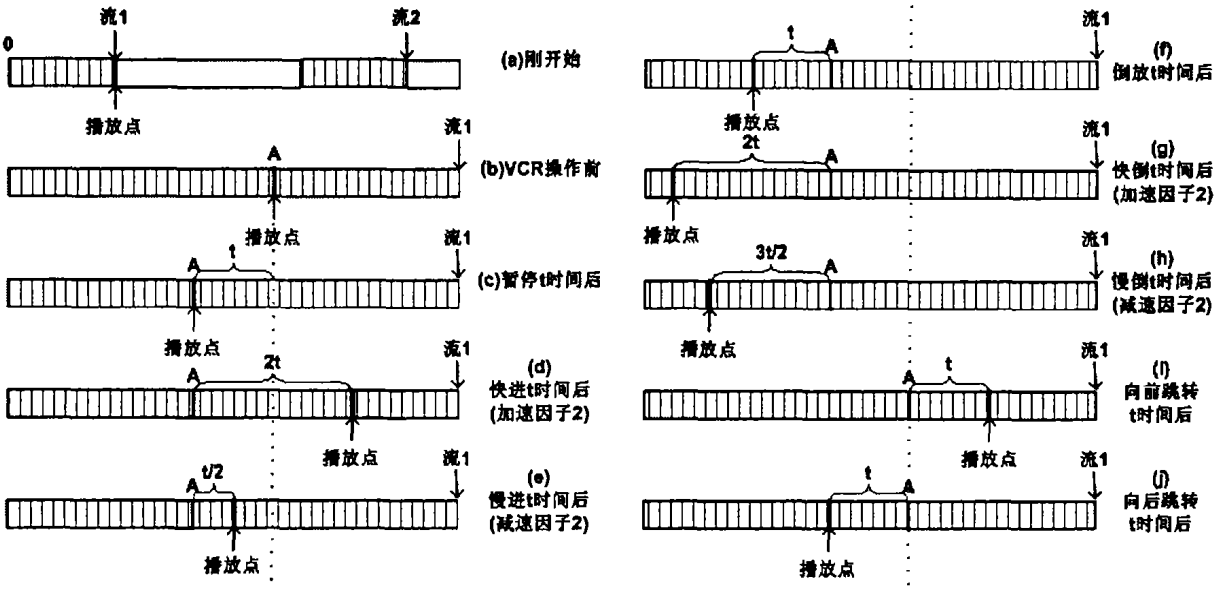


图3 各种VCR操作过程中的缓存数据情况

3.3.1 *正常播放* 用户刚发出影片服务请求时，缓存空间没有任何数据。为支持实时服务，服务器立即为其开辟一个新流流1满足观看需求，同时为用户调度另一个目标流流2。影片播放期间，客户在缓存中缓存流1和流2的数据。开始阶段，两流之间存在一个断点(如图3(a)所示)；但随时间推移，客户端缓存逐渐被填满(如图3(b)所示)，此时用户被合并到一个根流，只需要接收该流。此后，播放速率与传输速率相等，所以如果不发生VCR操作，播放点和接收点的相对位置将维持不变。当客户缓存溢满后，系统按LILO方式丢弃已播放数据。

3.3.2 *暂停操作* 暂停操作使播放点维持不动，但若缓存未满，会继续接收流。一旦缓存溢满，将丢弃最左端的帧。此时，播放点会相对逐渐左移，如图3(b)(c)所示。若播放点移至缓存最左端时用户仍处于暂停状态，用户必须停止接收流，退出多播组。用户取消暂停操作后将继续正常播放。

3.3.3 *快进操作* 快进使播放速度增加到流接收速度的两倍，因此播放点与接收点之间的距离会逐渐缩小，如图3(b)、(d)。一旦播放点追上接收点，缓存必须向服务器发送立即请求，要求服务器以速率$2b$发送播放点之后的数据，用户边接收边播放，直到快进操作停止为止。

快进流的消耗带宽为$2b$，由于系统假设客户带宽上限为$2b$，因此快进流之间无法合并，对服务带宽的消耗更为严重。为减少快进流申请概率，当客户发出快进操作时，主动缓存会检查缓存现有数据，如果播放点后有断点，且该断点不再被填充，主动缓存将向服务器发送一个延迟请求，要求服务器在播放到达断点之前尽早发送断点数据，从而降低快进流的请求概率。

3.3.4 *慢进操作* 慢进操作情况与暂停操作类似，慢进状态下，播放速率小于接收速率，播放点逐渐左移且与接收点的相对距离逐渐增大，如图3(b)、(e)所示。一旦播放点移到缓存最左端，缓存就必须退出多播组，停止流的接收。

如果慢进阶段播放点以后存在断点，且不在接收流，缓存会立即向服务器发送延迟请求。如果服务器没有响应，播放点会前进至断点，缓存现有数据将不能满足用户请求，会向服务器发送立即请求。由于正常流的速率大于慢进播放速率，因此可以用正常流来响应慢进请求，而不必服务器专门提供慢进流的服务。

在慢进过程中，受缓存大小限制，缓存溢满会导致流接收停止，而缓存数据缺失又要求向服务器发出请求，对这两种相异操作需要有一个折衷处理。可为缓存中的断点起始和播放点之间的距离设置一个阈值来权衡两操作。

3.3.5 *倒放、快倒、慢倒操作* 图3(b)、(f)、(g)、(h)分

别显示了这几种操作中缓存数据的变化情况。当发生倒放、快倒、慢倒操作时，缓存会从播放点开始向左搜索断点(包括缓存的最左端)。通常情况下，此时缓存往往正在接收播放点以后的数据流，故缓存会通过发出立即请求或延迟请求提出数据请求。一旦接收到服务器响应，客户端就必须停止当前接收流，转而加入服务器返回的组播地址接收数据。

一般情形下，客户总会在一段时间后从回退状态恢复到正常播放状态，因此可以设定一个阈值，使得仅当断点起始与播放点间距小于该阈值时，才发送延迟请求。这样可避免频繁发送不必要的延迟请求而导致用户离开原多播组后又申请相同数据，从而减少无谓的流合并开销和服务器调度开销。阈值可以是一个固定值，也可由服务器根据当前状态(例如当前负载，当前回退流的数量大小，甚至可以是服务器端对于客户VCR行为的统计数据)来决定。

3.3.6 *向前跳转、向后跳转操作* 跳转操作使用户在瞬间从当前播放点向前或向后跳转到任一位置(如图3(b)、(i)、(j)所示)，并以该位置为当前播放点继续正常播放。

3.3.7 *VCR操作结束后的正常播放* 返回正常播放状态后，缓存首先停止一切特殊流的接收。然后，以当前播放点为起点，向右搜索断点。此时可能会出现以下情况：1)因跳转或是快进操作导致断点起始即为当前点。在这种情况下，缓存必须向服务器发起立即请求。2)断点位于当前播放点之后且不再被填充。此时缓存向服务器发送延迟请求，要求在播放到达断点之前开始传送断点数据。3)断点正在被填充或者已到达影片末尾。该情形表明客户端已经缓存了用户所需要的全部数据，不必向服务器发送请求。

3.3.8 *冗余数据的处理* 当缓存中存在当前接收的流数据时，会出现数据冗余，有两种处理方法：1)停止接收流，离开组播组；2)丢弃缓存中数据，继续监听流。

如果正在接收的流为特殊流，应停止接收该流。因为通常情况下，特殊流大多只为单个用户服务(如快进流和快退流)或是在少量用户间共享(如倒放流)，离开这些流往往意味着服务器可以停止发送这些流，从而减轻服务器的负担。

如果正在接收的流是正常流，从表面上看，停止接收流可能会减轻服务器的负载，但实际上却可能会增加服务器的负载。这是因为，正常流通常被多个用户共享，一个用户离开不会导致该流停止，而对离开用户而言，如果缓存中存在断点，则在随后某个时候会再次向服务器发送请求，因而反而可能会导致服务器负载增加。我们的实验也证实了这一点。

3.4 服务器端的流合并处理

SRMT仅考虑了用户以恒定速率请求完整流时的流合并优化问题。而用户的VCR操作的请求流可从影片的任一位置开始，而且请求流的速率也不恒等于正常播放速率 b。因此，为支持VCR功能，必须对SRMT进行扩展。本文针对VCR操作特性，对基本流合并算法做了如下扩展：

1. 在服务器端维护正常流、快进流、倒放流、快退流四种流的列表。

2. 由于VCR请求流不一定从影片开始位置0开始，因此这里引入虚拟开始时间概念。假设某个流A的速率为 r(当为回退流时，$r<0$)，流的实际开始时间为 T_{start}，起始位置为 a，那么，虚拟开始时间 VT_{start} 可定义为：

$$VT_{start}=T_{start}-a/r$$

如果在坐标轴上以时间为 x 轴，以数据位置为 y 轴，则一个流或其延长线与 x 轴的交点即为其虚拟开始时间。以图4为例，流 A、B、C、D 的虚拟开始时间分别为 T_A、T_B、T_C、T_D。

3. 服务器将四种流列表中的流分别按虚拟时间进行排序。同类流之间可以采用基本流合并算法进行合并，但用虚拟时间代替实际时间。

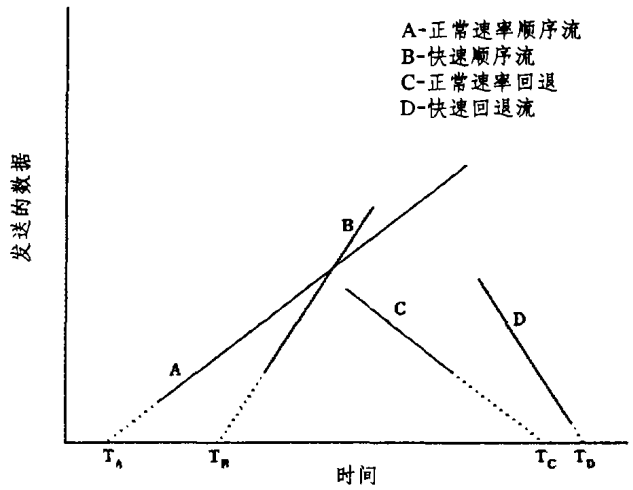


图4 虚拟开始时间

3.4.1 *立即请求响应* 立即请求要求服务器立即为其调度所需流，并确定目标流。对于正常流的请求，可采用类似于SRMT的算法在正常流列表中查找目标流，但需用虚拟开始时间代替实际开始时间(见算法1(a))。

对于快进流和快倒流，由于客户端带宽上限正好等于快速流速率，服务器无法再对这些流进行合并。因此其性能相当于为每个快进和快倒请求调度一个单播流，直到VCR操作结束。当客户端带宽大于快速流速率时，可以用扩展的SRMT算法来进行快速流的调度与合并。

算法1 服务器对立即请求的响应调度算法

```
void targetCommonStream(Stream * a){
    Stream *p = a->previous; //按虚拟时间顺序向前搜索
    while (p != NULL){
    if (a->virttime - p->virttime < p->endtime - nowtime())
    break; //能在p结束以前追上p
    p = p->previous;
    }
    a->target = p;
        //修改流a的结束时间
    if (p != NULL)
      a->endtime = nowtime()+ (a->virttime - p->virttime);
      else a->endtime = a->virttime + movielen;
}
```

(a)正常流的目标流定位

```
void targetRewindStream(Stream * a){
    //在回退流中搜索其候选目标流
    Stream *p = a->previous;
    double endtime1;
    while (p != NULL){
    endtime1 = nowtime + (a->virttime - p->virttime);
        //追上时间
    if (endtime1 < p->endtime)break;//在p结束前追上
    p = p->previous;
    }
    //在正常流中搜索其候选目标
    Stream *q = commonStreams;
    double endtime2;
    while (q != NULL){
    if (q->virttime > x->virttime){q = NULL; break;}
    endtime2 = (a->virttime + q->virttime)/ 2; //相遇时间
    if (endtime2 > nowtime())
        if (endtime2 < q->endtime)break;//在p结束前相遇
    q = q->next;
    }
    //判断选择哪个候选目标流
    if ((p == NULL)&& (q == NULL)){
        a->target = NULL;
        a->endtime = a->virttime;
    }else if (q == NULL){
```

```
        a->target = p;
        a->endtime = endtime1;
      }else if (p == NULL){
            a->target = q;
            a->endtime = endtime2;
      }else {
      if (endtime1 <= endtime2){
            a->target = p;
            a->endtime = endtime1;
      }else {
            a->target = q;
            a->endtime = endtime2;
      }
    }
}
```

(b)倒放流的目标流定位

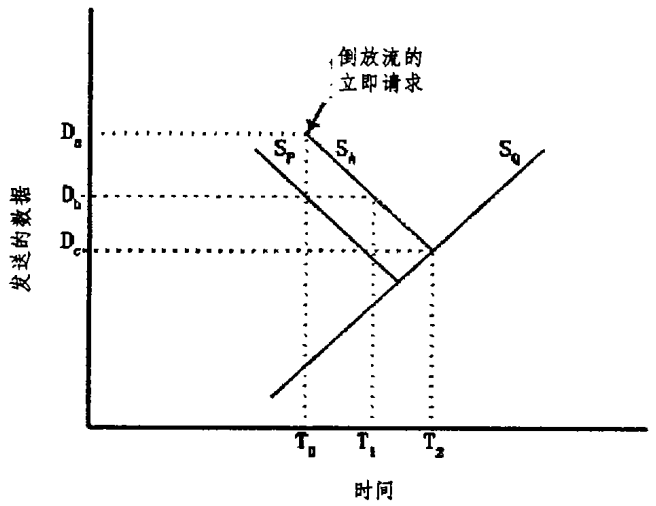


图5 倒放流的目标流定位

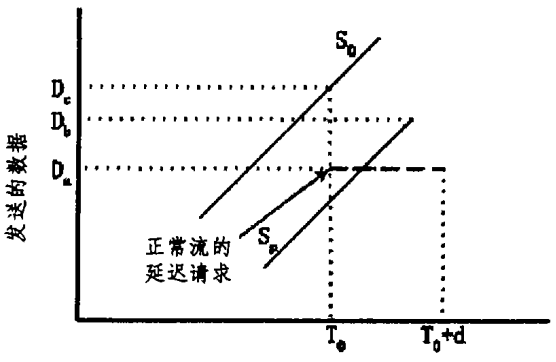
对于倒放流，服务器即调度一个倒放流。由于倒放流速率与正常流速率相同，因此，除可用扩展的 SRMT 算法从倒放流列表中选择以外，还可从正常播放流列表中为其选择目标流，从而减少服务器端带宽消耗(见算法1(b))。以图5为例，假设在 T_0时刻，客户 A 发送倒放流的立即请求，请求数据的起始点为 D_a，服务器立即为其调度一个倒放流 S_A，同时选择倒放流 S_P、正常流 S_Q 作为其候选目标流，其结束时间分别为 T_1 和 T_2。由于 T_1较小，因此将选择流 S_P 作为目标流。在实际系统中，由于回退操作概率小，因而回退流数目远少于正常流数目，而且其持续时间也相对较短，因此很少出现以回退流作为目标流的情形。我们的实验也证实了这一点。

3.4.2 延迟请求响应 延迟请求表明客户不急需请求数据，因此，可尽量选择一个能在延迟时间内满足请求的现存流。这样，既能使用户提前接收所需数据，又不会增加服务器负载。具体算法见算法2。

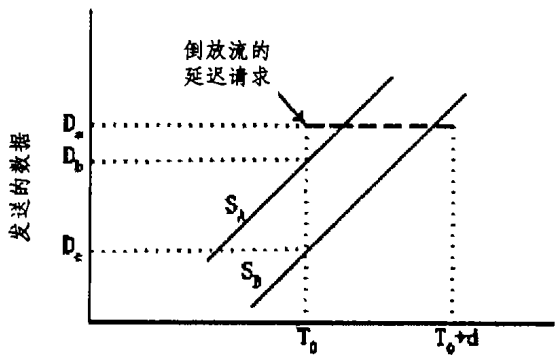
算法2 服务器对延迟请求的响应调度算法

```
void answerDelayreq(double reqpos,double delay,double rate){
    double reqVirttime1 = nowtime()- reqpos;
    dobule reqVirttime2 = reqVirttime1 + delay;
    Stream * p = commonStreams;
    if (rate > 0){ //正常流的延迟请求
        while (p != NULL){
            if (p->virttime > reqVirttime2)
                {p = NULL; break;}//p 晚于最大延迟
            if (p->virttime > reqVirttime1)
                if (p->endtime - p->virttime > reqpos)
                    break; //p 能在延迟之内到达请求点
            p = p->next;
        }
    }else { //倒流的延迟请求
        Stream * lastp = NULL;
        while (p != NULL){
            if (p->virtime > reqVirttime2)
                {p = NULL; break;}//p 晚于最大延迟
            if (p->virttime > reqVirtime1)
                if (p->endtime - p->virttime > reqpos)
                    lastp = p;//p 在延迟之内到达请求点
            p = p->next;//选最晚到达请求点的流
        }
        p = lastp;
    }
    answer(p);
}
```

以图6为例说明服务器对延迟请求的调度算法。图6(a)中，客户发送一个正常流的延时请求，请求发送时间为 T_0，请求允许延时为 d，请求的起始位置为 D_a。服务器从点(T_0，D_a)开始向右搜索正常速率的顺序流，查找从 T_0～T_0+d 时段内尽早经过 D_a 的流，即坐标轴上与虚线段相交且虚拟开始时间最小的正常流。一旦有符合条件的流，则将其地址发送给客户，以便客户加入该多播组。图6(a)中的流 S_P 为满足条件的一个流。由于重定位目标流可能会延长该流的播放时间，因此新客户加入 S_P 后，不必为 S_P 重定位目标流。图6(a)中，流 S_Q 为 S_P 的目标流，如果 S_P 没有重新定位目标流，客户将错过 D_b～D_c 段的数据。这种情况不会导致异常，因为用户在随后可发现该断点，并通过发送立即请求或延迟请求来弥补缺失数据。



(a) 正常流的延迟请求



(a) 倒放流的延迟请求

图6 服务器对延迟请求的响应

对于回退流的延迟请求，假设发送请求的时间为 T_0，许可延迟为 d，请求的起始位置为 D_a。此时服务器不必立即发送一个回退流，只要查找一个 T_0～T_0+d 时段内能到达 D_a 的流。回退流的延迟请求和顺序延迟请求有所不同，它需要的

是 D_a 以前的数据，这就意味着在规定时间 d 内向客户发送了尽量多的 D_a 以前的数据，所以必须在满足条件的多个流内选择一个尽量晚到达 D_a 的流。图6(b)中，流 S_A 和 S_B 都能满足在 d 时间内向客户发送 D_a 以前的数据的请求，但显而易见，S_B 发送了更多满足客户需要的数据。

4 模拟结果

4.1 环境设置

我们在 ns2[17] 中对 VCRSM 进行了模拟。模拟实验假设服务器端有 n 部影片，每部影片长度 l 为100分钟，用户加入是一个泊松过程，每个客户都有固定缓存空间 B，用户对影片的访问概率满足 Zipf 分布[4]。

图7给出了发生每一个特定 VCR 操作的概率以及平均持续时间的模型。在该模型中，一共有10种状态，表示10种 VCR 操作。假设 VCR 操作的持续时间满足平均值为 τ_i 的指数分布，在向前跳转和向后跳转两个状态中，平均持续时间表示客户从当前点向前或向后跳转的偏移量。

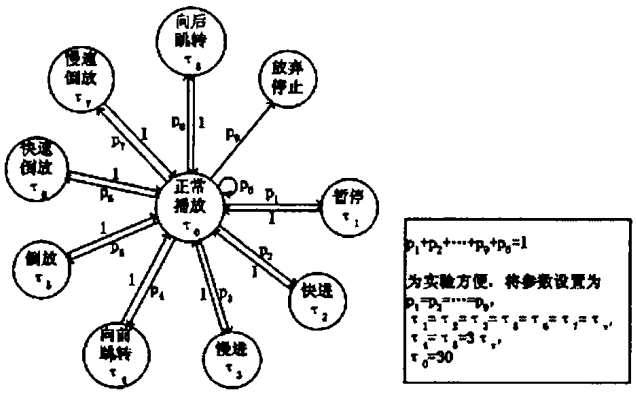


图7 用户 VCR 操作模型

4.2 各参数对服务器带宽的影响

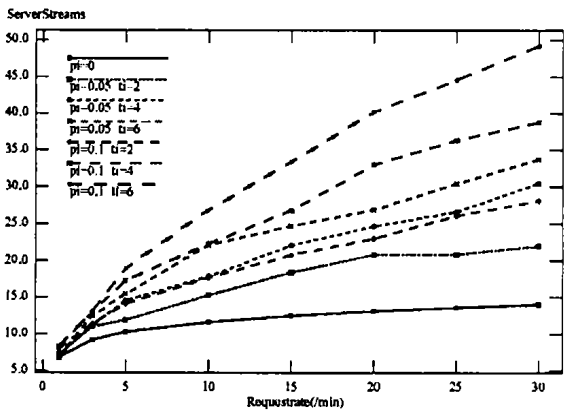


图8 用户 VCR 操作频度对服务器带宽的影响

图8显示了客户的 VCR 操作频度对服务器带宽的影响，这里的 VCR 操作频度与 τ_i 及 p_i 有关，p_i 越大，p_0越小，用户的 VCR 操作频度就越大；$\tau_i(i\neq 0)$越大，τ_0越小，用户的 VCR 操作频度就越大。实验表明，τ_0和 τ_i 对服务器带宽的影响可用 τ_i/τ_0代替，即当 τ_i/τ_0一定时，τ_i 和 τ_0的变化对服务器带宽的影响不大。为了方便实验并不失一般性，我们未考虑用户对不同 VCR 操作偏好的差异，即 $p_i=p_j(i,j\neq 0)$；同时，除跳转外其余 VCR 操作的持续时间相同，跳转操作的偏移量是其它 VCR 操作持续时间的3倍，假设客户缓存容量 B 为50分钟。通过对 p_i、τ_i/τ_0设置不同组合(见图8)，考查不同的操作频度对服务器带宽的影响。其中，当 $p_i=0$时，无 VCR 操作，τ_i 与 τ_0的设置无意义，VCRSM 退化到 SRMT。从图8中可以看出，即使 VCR 操作频度相当大，服务器带宽仍然相对较小，远小于单播系统中的服务器带宽 $l\lambda$。

在 VCRSM 中，客户端的主动缓存起了很大作用。图9分析了客户端缓存大小对服务器带宽的影响。当缓存大小大于等于影片长度时，服务器带宽完全不受客户缓存影响，即不存在因客户缓存限制导致服务器负担加重的情况。从图9中可以看出，当客户端缓存相对较小时，服务器端需分配的流相对较多。随着客户端缓存的增大，服务器端带宽逐渐减小，直到客户端缓存达到影片大小的一半时，服务器端带宽基本上不再受客户缓存大小的影响。

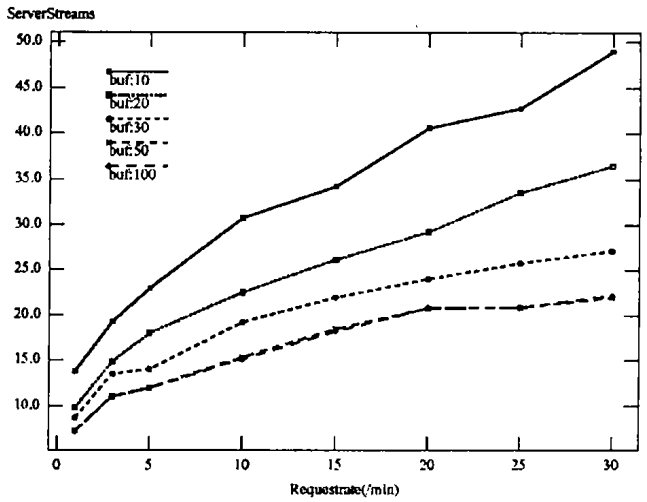


图9 客户端缓存大小对服务器带宽的影响

4.3 和其他调度算法的比较

如2.2.2节中所述，现有支持 VCR 功能的多播调度算法可以分为开环和闭环两类。相对于开环算法，VCRSM 具有零时延的优点。开环算法的带宽消耗衡定，虽然具有良好的可扩展性，但在客户到达率较低的情况下，带宽使用率较低。此外，开环算法不能保证 VCR 操作过程中影片的精确定位，从而降低了客户端的 QoS。

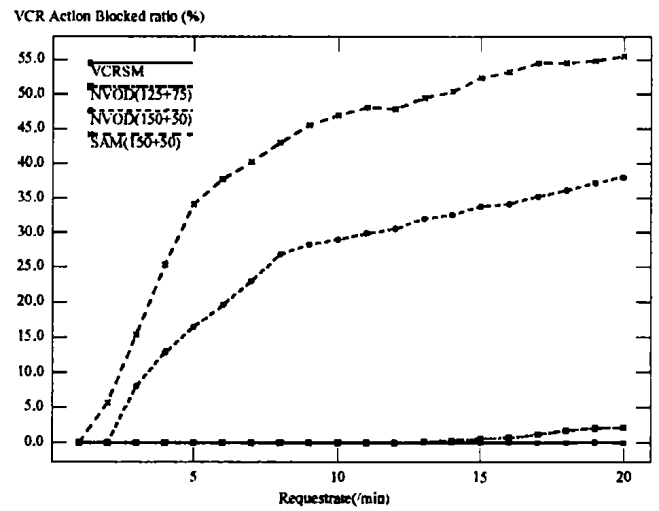


图10 不同系统 VCR 操作阻塞率的比较

图10给出了在相同 VCR 频度下 VCRSM 与闭环算法 SAM[15] 及 NVOD[16] 的比较。在服务器端有10部影片，带宽限制为200个通道。当 λ 增大到20时，VCRSM 的 VCR 操作阻塞率仍约为0%，而在 NVOD(125,75) 的 VCR 操作阻塞率为

2.5%，NVOD(150,50)和 SAM(150,25)的 VCR 操作失败率则要远远高于前两者，达到35%以上。并且，VCRSM 是零时延的，不存在客户反悔的情况；而 NVOD 和 SAM 是基于批处理的，客户的平均等待时间为0.4分钟，客户反悔率达50%以上。

在实验中还发现，当有 VCR 操作时，服务器端的带宽随着 λ 近似线性增大。这是因为，实验中假设快进和快退操作播放速率恰好为客户端的带宽，因此对于快速流，服务器无法再进行合并，相当于单播流的性能。但如果客户端带宽大于快速播放速率，快速流之间也可合并，故当 λ 趋向于无穷大时，相应带宽不会线性增长。

结论 本文结合客户端主动缓存技术对基本流合并策略进行扩展，提出了一个支持 VCR 功能的多播调度算法 VCRSM。VCRSM 在客户端通过主动缓存机制一方面预取数据，同时提供局部 VCR 操作提供支持，并使用延迟请求为流合并和服务器带宽优化创造条件。在服务器端，将流调度策略和 VCR 请求相结合，在提供实时响应的基础上通过优化流合并来提高流的共享度，减少服务器带宽消耗。

与已有 VOD 系统相比，VCRSM 的主要优点是：真正支持实时交互式流媒体服务，对所有 VCR 功能提供即时服务，没有延时(初始延时和 VCR 操作延时)；具有良好的可扩展性，实验表明，在一定磁盘空间支持下，VCRSM 的服务器端带宽消耗受用户规模及 VCR 操作频度影响相对较小。

参 考 文 献

1 Viswanathan S, Imielinski T. Pyramid Broadcasting for Video On Demand Service. In: Proc. SPIE Multimedia Computing and Networking Conf. (MMCN '95), 1995. 66～77

2 Hua K A, Shen S. Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems. In: Proc. of ACM Sigcomm'97, 1997

3 Eager D L, Vernon M K. Dynamic Skyscraper Broadcasts for Video-on-Demand. In: Proc. 4th Intl. Workshop on Multimedia Information Systems, Sept. 1998

4 Dan A, Sitaram D, Shahabuddin P. Scheduling Policies for an On-Demand Video Server with Batching. In: Proc. of ACM Multimedia, Oct. 1994. 15～23

5 Gao Lixin, Towsley Don. Supplying Instantaneous Video-on-Demand Services Using Controlled Multicast. In: Proc. IEEE ICMCS'99, Florence, Italy, June 1999

6 Gao Linxin, Zhang Zhi-Li, Towsley D. Catching and Selective Catching: Efficient Latency Reduction Techniques for Delivering Continuous Multimedia Streams. In: Proc. of the Seventh ACM Intl. Conf. on Multimedia (Part 1), 1999. 203～206

7 Ramesh S, Rhee I, Guo K. Multicast with Cache (Mcache): An Adaptive Zero-Delay Video-on-Demand Service. In: Proc. of IEEE INFOCOM, 2001

8 Eager D, Vernon M, Zaborjan J. Minimizing Bandwidth Requirements for On-Demand Data Delivery. In: Proc. of the ninth annual ACM-SIAM symposium on Discrete algorithms, 1998. 11～20

9 Eager D, Vernon M, Zahorjan J. Optimal and Efficient Merging Schedules for Video-on-Demand Servers. In: Proc. 7th ACM Int'l. Multimedia Conf. (ACM MULTIMEDIA '99), Nov. 1999. 199～202

10 Kwon J B, Yeom H Y. Providing VCR Functionality in Staggered Video Broadcasting. Trans. on Consumer Electronics (SCI), 2002, 48(1): 41～48

11 *Fei Zongming, Kamel I, Mukherjee S, Ammar M H. Providing* Interactive Functions for Staggered Multicast Near Video-On-Demand Systems. In: Proc. of the IEEE Intl. Conf. on Multimedia Computing and Systems, 1999

12 Almeroth K C, Ammar M H. On the Use of Multicast Delivery to Provide a Scalable and Interactive Video-on-Demand Service. IEEE Journal of Selected Areas in Communications, Aug. 1996, 14

13 Fei Zongming, Ammar M H, Kamel I, Mukherjee S. Providing Interactive Functions through Active Client-Buffer Management in Partitioned Video Multicast VOD Systems. In: Proc. of First Intl. Workshop on Networked Group Communications (NGC'99), Nov. 1999. 152～169

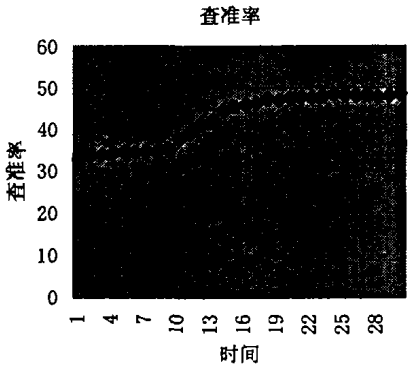
14 Biersack E W, Jean-Marie A, Nain P. Open-loop Video Distribution with Support of VCR Functionality. In: Proc. of the Performance 2002 Conf. Roma, Italy, 2002

15 Liao Wanjiun, Li V O K. The Split and Merge Protocol for Interactive Video-on-Demand. IEEE MultiMedia, 1997, 4(4): 51～62

16 Abram-Profeta E L, Shin K G. Providing Unrestricted VCR Functions in Multicast Video-on-Demand Servers. In: Proc. of IEEE Intl. Conf. on Multimedia Computing and Systems (ICMCS'98), Austin, Texas, 1998

17 The ns Manual. July 23 2003. http://www.isi.edu/nsnam/ns/

(上接第78页)



由此可见，该系统具有较高查全率及查准率，并能够较快地适应用户兴趣的变化。

结束语 本文提出了一种动态跟踪用户兴趣的推荐系统。该系统从语义角度上对用户兴趣细化分类，形成多个个性化向量，并从语义角度上计算了用户兴趣与文档间的相关度，从而能够实时检测用户兴趣及用户兴趣的变化。经实验证明，所构造的用户兴趣模型对用户作出的推荐较为全面和准确，而且在用户兴趣发生变化的情况下能快速、准确地对用户兴趣模型作出调整。我们要作的下一步工作是：进一步细化用户行为对其兴趣模型的影响。比如鼠标滚动、页面保存、添加到收藏夹等等。

参 考 文 献

1 Bonino D, Corno F. A Real-Time Evolutionary Algorithm for Web Prediction WI-2003. In: the 2003 IEEE/WIC Intl. Conf. on Web Intelligence, Halifax, Canada, Oct. 2003

2 Claypool M, Brown D. Inferring User Interest. IEEE Internet Computing, Nov./Dec. 2001

3 Callan J, Lu Z, Crolt W. Searching distributed collections with inference networks. In: Proc. of ACM SIGIR Conf. Seattle, 1995. 21～28

4 Campos L M, Huete J F. Building Bayesion Network-Based Information Retrieval Systems. In: Proc. of the 11 Intl. Workshop on Database and Expert Systems Applications(DEXA'00)

5 Campos L M, Huete J F. Document Instantiation for Relevance Feedback in the Bayesian Network Retrieval Model. ACM SIGIR' 01 Workshop on Mathematical/Formal Methods in IR, New Orleans, Louisians, 2001

6 Rbeiso B, Munts R. A belief network model for IR. In: Proc. of 19th ACM-SIGIR Conf., 1996

7 Kelly D, Teevan J. Implicit feedback for inferring user preference: a bibliography. SIGIR Forum, 2003, 37(2): 18～28

8 Han Jiawei, Kamber M. Data Mining: Concepts and Techniques. The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers, ISBN 1-55860-489-8, Aug. 2000. 550

9 Balabanovic M, Shoham Y, Yun Y. An Adaptive Agent for Automated Web Browsing. Journal of Visual Communication and Image Representation, 1995, 6(4)