VBR 流磁盘 I/O 的优化调度及特定的缓冲计算

谢建国, 陈松乔, 陈建二

(中南大学 信息科学与工程学院, 湖南 长沙 410083)

摘要: VBR 流如视频流, 由于其位率的突发性和频繁的改变, 给传输系统(如网络和磁盘 I/O 端口) 带来了巨大的挑战. VBR 流的平滑调度和传输控制成为分布式多媒体应用领域内研究的热点. 对于存储的视频, 在连续传输中, 预缓冲是平滑 VBR 特性非常有效和实用的方法. 作者在基于为网络传输而考虑的 VBR 流平滑算法的基础上, 讨论了考虑物理磁盘块存取特性的 VBR 流二次平滑调度及缓冲计算方法. 首先, 给出了一个在最小缓冲尺寸下的磁盘 I/O 调度规划算法 A; 然后, 计算了在最大存取单元(为某种需要而设定的) 不超过某一特定值时的最低缓冲需求, 及在该最低缓冲需求下的一个复杂度为线性时间的磁盘 I/O 调度规划算法 C. 分析结果表明, 所设计的算法能 100% 地利用磁盘空间.

关键词: 变位率; 缓冲; 平滑; 算法

中图分类号: TP37 **文献标识码**: A **文章编号**: 1005-9792(2001)02-0204-05

分布式多媒体应用如 VOD、远距离学习与合作、视频会议等, 它们都需要存储的视频通过高速网实时传输, 以及这些存储视频连续回放才能实现. 然而, 由于压缩的视频数据在连续传输或回放过程中呈现出强的变位率(variable-bit-rate, VBR) 特性很明显, 其峰值速率往往是其平均值的好几倍, 且变化频繁, 这种不稳定性使视频的存储管理及网上传输复杂化, 对网络传输的服务质量 QoS 提出了巨大的挑战. 适合传输 CBR 的 2 种网络服务模型(确定性的保证服务[1] 和再协商式的 CBR 服务(RCBR)[2]) 用来传输 VBR 流(如视频流), 其服务质量难以保证. 由于多媒体应用的需求和为了提供满意的服务质量, 在网络及多媒体领域内 VBR 流平滑技术的研究已成为热门课题[3~12]. 目前, 平滑与控制 VBR 流的技术有: 一是多流复用技术, 分为时间复用[1] 和统计复用[2,7]; 二是预缓冲技术, 即利用客户端的缓冲区[3] 或沿途中接点的输入输出缓冲区[5], 预送数据、平滑变位率, 缩减峰值数据的要求; 三是其它技术, 包括反馈控制[11]、线性预测[12] 等预测技术, 预测 VBR 流的传输特性, 以缩减峰值率带宽需求, 调整编码参数[10~12], 减少峰值率, 有时综合复用和预缓冲技术来平滑峰值率的带宽需求[7].

作者利用预缓冲技术, 讨论兼顾存储系统的块传输特性, 利用服务器本地的缓冲区平滑存储的视频数据[3]. 给出了缓冲计算和相应情况下存取调度的算法及存储块划分策略. 这种算法的结果, 既考虑了 VBR 流的网上传输特性, 又兼顾了磁盘存取特性, 极大地利用了磁盘空间, 优化了存取效率.

1 VBR 流的 CBR 方式传输

文献[3, 5] 利用流传输路径上的缓冲区, 将压缩的 VBR 视频流分成许多 CBR 段, 再分别在确定性的保证服务和 RCBR 服务 2 种网络 QoS 服务模式下传输, 取得了令人满意的效果. 以文献[3] 为例, 它假定在客户的播放区有一尺寸为 b 的缓冲区用来预存数据, 如图 1 所示.

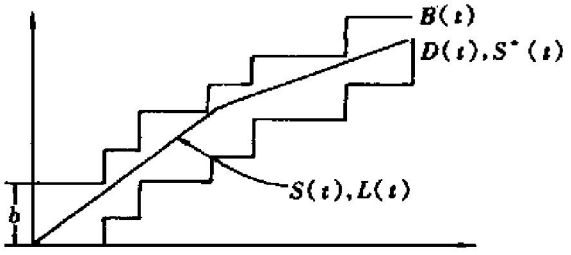


图 1 传输规划图

图 1 的横轴表示传输时间, 以视频帧为单位, 纵轴表示数据总量. $D(t)$ 表示在 $[1, t]$ 时间区间中播

收稿日期: 2000-10-07
基金项目: 国家杰出青年科学基金资助项目(69928201); "长江学者奖励计划" 基金资助项目
作者简介: 谢建国(1964-), 男, 湖南祁阳人, 中南大学副教授, 博士研究生, 从事多媒体、网络优化及算法研究.

过消耗的数据总量; $B(t)$ 表示在 $[1, t]$ 过程中缓冲区能容纳数据的总量, 条件是缓冲区数据既不溢出也不饥饿. $S(t)$ 是一条介于 $B(t)$ 和 $D(t)$ 之间可行的、由视频服务器发送的数据传输规划, 它由 1 组 CBR 段构成, 且必须满足条件: $D(t) \leqslant S(t) \leqslant B(t)$. 对于给定的 b, 在 $B(t)$ 和 $D(t)$ 之间可以找出许多可行的 $S(t)$, 但其中存在最优的 $S(t)$, 用 $S^*(t)$ 表示, 文献[3]给出了寻找算法和证明. $S^*(t)$ 具有 2 条特性: 一是每一个 CBR 段尽可能长; 二是当从一个 CBR 段必须过渡到另一个 CBR 段时, 其改变点应尽可能早, 这样可以保证传输率的过渡变化尽可能小, 从全局看 $S^*(t)$ 显得尽可能平滑. $S^*(t)$ 的这些特性在网络传输服务中能得到较好的 QoS 保证.

2　问题的提出

$S^*(t)$ 的计算是应网络的传输模式 QoS 而提出的, 忽略了块存取特性的磁盘系统. 视频服务器以 $S^*(t)$ 方式向子网发送数据, 存储子系统有 2 种途径支持它: 一是同步方式, 以 $S^*(t)$ 方式从磁盘读取同样多的数据; 二是异步方式, 磁盘以自己的方式读出数据, 预置于缓冲区, 确保以 $S^*(t)$ 方式发送数据, 不出现数据饥饿现象.

对于能同时支持上千路流并发存取的网上视频服务器, 典型的调度方式是周期地为每一路流存取一个固定长度的视频数据块(通常是 1 个逻辑块只需 1 次寻找定位). 而磁盘是块设备, 错误校正信息以物理块为单位, 每次以整数个物理块读写为有效. 为实现同步方式, 若在 1 个调度周期内, $S^*(t)$ 方式需要的数据块长度不是物理块的整倍数, 而以整倍数存放(不足部分用非有效字符填充, 不允许跨块存储) 及整倍数块读出, 这无疑浪费了磁盘 I/O 的有效带宽, 导致同时支持存取的并发流数下降. 若以有效数据整倍数存放及整倍数块读出, 如异步方式, 多出的数据置于缓冲区, 留到下一个周期发出. 这种方式充分利用了磁盘 I/O 的有效带宽, 但需要内存支持. 由于千路流的存在, 内存资源就显得极为有限, 需要仔细地规划.

3　基于 I/O 特性的平滑优化

二次平滑是建立在 $S^*(t)$ 的基础之上, 考虑磁盘存取特性而提出的, 其目的是充分利用磁盘 I/O 带宽和磁盘存储空间. 显然, 若存取速率满足了 $S^*(t)$, 则也就满足了 $D(t)$.

3.1　符号的定义与说明

存储块　存储的视频以存储块的方式安置在磁盘上. 这里的存储块指的是逻辑块, 它不同于视频文件常用的 CTL(Constant Time Length) 或 CDL(Constant Data Length) 存储方式, 用 PCTL(Pseudo-Constant Time Length) 表示, 指在 1 个调度周期内为某一路流计划存取 1 个存储块的长度. 它要求是物理块的整数倍, 在存取中 1 次磁头定位, 不存在块内定位问题. CTL 和 PCTL 都是变长存储块.

设 T 表示视频服务器完成 1 个调度周期所需的时间, 在 1 个调度周期内为每一路流存取 1 个存储块 PCTL.

以图 1 为例. 横轴表示某 VBR 流数据传输时间 t, 以调度周期 T 为单位; 纵轴表示数据总量; b 表示服务器在服务器端为该流分配的缓冲区尺寸; N 表示该流完成数据传输(或完成播放) 所需的总周期数, $t \in \{1, \cdots, N\}$.

$s^*(t)$ 表示视频服务器在 t 时刻或某个调度周期向该流客户播放器发出的数据长度.

对于 $S^*(t)$, t 以调度周期 T 作为计数单位. 表示在 $[1, t]$ 时间区间内视频服务器向该流客户播放器发出的数据总量, 有 $S^*(t) = \sum_{i=1}^{t} s^*(i)$.

$B(t)$ 表示缓冲区 b 在 $[1, t]$ 时间区间内能接受的数据总量, 有 $B(t) = S^*(t-1) + b$, $B(1) = b$.

$l(t)$ 表示视频服务器在 t 时刻或某个调度周期检索的数据长度.

$L(t)$ 表示在 $[1, t]$ 时间内视频服务器为该流存取的数据总量, 有 $L(t) = \sum_{i=1}^{t} l(i)$. 用 N 维向量 $\boldsymbol{L} = [l(1), \cdots, l(N)]$ 表示 1 个可行的存取规划, 或 1 组存储块 PCTL; $\boldsymbol{S}^* = [s^*(1), \cdots, s^*(N)]$ 表示 1 个给定的网络调度. 它们满足条件 $S^*(t) \leqslant L(t) \leqslant B(t)$, 同时 $l(i)$ 是磁盘基本输入输出物理块的整数倍, $i \in \{1, \cdots, N\}$.

先寻找 1 个可行的存取规划 $\boldsymbol{L}$, 基于下面 2 个基本点: 一是 $l(i)$ 是磁盘基本输入输出物理块的整数倍, $i \in \{1, \cdots, N\}$; 二是缓冲区尺寸 b 尽可能小. 用 $\Delta L = \sum_{t=1}^{N} (l(t) - \bar{l})^2 / N$ (其中, $\bar{l} = \sum_{t=1}^{N} l(t)/N$) 来度量 $\boldsymbol{L}$ 的平滑性能. 平滑性好对于存

储管理特别是接纳计算十分有利.

3.2 最小缓冲需求的存取规划算法

算法的基本思想是: 取缓冲区尺寸 $b=p$, 其中 p 表示基本输入输出物理块的尺寸(对于大型磁盘组通常是 2 kB, 4 kB, 6 kB 等, 这视系统而定), 它是最小的需求, 后面将给出证明. 对于 $i\in\{1, \cdots, N\}$, $l(i)$取 $s^*(i)/p$ 的上整或下整, 保证公式 $S^*(t)\leqslant L(t)\leqslant B(t)$成立. 下面给出一个简单规划算法 A.

算法 A

输入: $s^*(1), \cdots, s^*(N)$和 p.

输出: $l(1), \cdots, l(N)$一组存储块长度.

$b=p$, $\Delta b=0$, $\Delta s=0$; // Δb 表示缓冲区中数据的节余量.

For $i=1$ To N

If ($\Delta s=$ REM ($s^*(i)/p$) $=0$) // 若 $s^*(i)$是 p 的整数倍.

$l(i)=s^*(i)$;

Else If ($\Delta b\geqslant\Delta s$)

$l(i)=s^*(i)-\Delta s$, $\Delta b=\Delta b-\Delta s$;

Else $l(i)=s^*(i)+(p-\Delta s)$, $\Delta b=\Delta b+(p-\Delta s)$;

End For;

算法 A 能在 n 步内完成, 时间复杂度为 $O(n)$, 它使用了最小缓冲区, 但平滑性能不如 S^*, 最大的 $l(i)$不小于最大的 $s^*(i)$, 最小的 $l(i)$不大于最小的 $s^*(i)$.

3.3 缓冲区 b 的计算

这里的缓冲区是指因 2 次平滑而引入的, 由于磁盘 I/O 和网口 I/O 传输速率的差异以及系统因其它目的而引入的缓冲区不计其内.

可以看出 b 存在 1 个取值范围是$[p, b_{\max}]$, 其中取 $b_{\max}$时, 有 $l(1)=l(2)=\cdots=l(N)=\overline{l}$, 大于 $b_{\max}$ 的取值失去其物理意义.

3.3.1 b 的最小取值

让 $s^*(i)=n_1p+\Delta s$, $l(i)=n_2p$, 分 3 种情况进行讨论.

a. $l(i)=s^*(i)$, 第 i 个周期没有多余的数据要驻留 b 中, b 的最小取值可以为 0.

b. $l(i)>s^*(i)$, 第 i 个周期有多余的$l(i)-s^*(i)$数据要驻留 b 中, 在最小情况下取 $n_2=n_1+1$, 且 $\Delta b=0$, 至少有 $p-\Delta s$ 余量, 当 $\Delta s\rightarrow 0$ 时, 有 $p-\Delta s\rightarrow p$, 所以 b 不可以小于p.

c. $l(i)<s^*(i)$, 第 i 个周期没有多余的数据要驻留 b 中, 但需要在第 i 个周期以前缓冲区中至少有$s^*(i)-l(i)$的余量. 在最小情况下取 $n_2=n_1$, 因有$\Delta s<p$, 当 $\Delta s\rightarrow p$ 时, 为保证缓冲区能足以提供 Δs 的数据量, 所以 p 是最低的需求值.

除对所有的 i 有$l(i)=s^*(i)$以外, p 是最低的需求值, 也是充分可行的, 算法 A 说明了这一点. 由第 5 步和第 7 步知 $\Delta b=\Delta b+(p-\Delta s)=p-(\Delta s-\Delta b)<p$, 由第 3 步知 $\Delta s<p$, p 足够容纳 Δs 的不足数据量.

3.3.2 b 的一般取值问题

引入 b 的目的有 2 个: 首先是为了调整 $l(i)$, 使之是 p 的整数倍, 算法 A 及上面的证明说明了 b 的最小取值为p; 其次是 $\boldsymbol{L}$ 的平滑性问题, 很明显, b 取值越大, $\boldsymbol{L}$ 的平滑性越好, 对于给定的 S^*, 理论上 b 存在 1 个最大值 $b_{\max}$, 使得 $l(1)=l(2)=\cdots=l(N)=\overline{l}$, 此时 $\Delta L=0$. 作者设计了 1 个求 $b_{\max}$的算法, 经计算, 对于以 MPEG-1 编码的视频流 $b_{\max}$在几 M 以上, 以 MPEG-2 编码的视频流 $b_{\max}$在 10 M 以上. 对于能同时支持上千路流的 Internet 服务器. 难以提供如此大的 $b_{\max}$. 因此, 这里对 $b_{\max}$不作进一步讨论.

图 2 纵轴表示一个单位时间的数据量, 横轴表示整数序列 1, …, N 中的一段. 矩形部分由一组 $s^*(1), s^*(2), \cdots, s^*(N)$等离散点构成, 直线表示在 $b_{\max}$支持下最平滑的存取规划. 两线的交点用 t_1, t_2, …等表示.

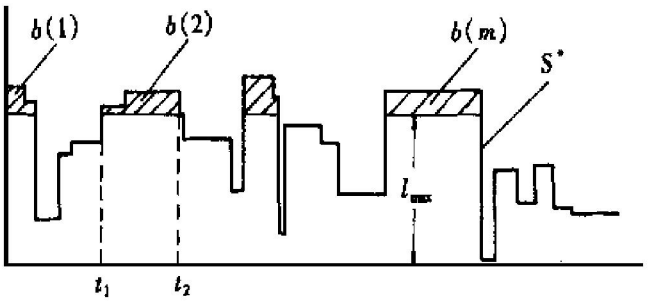


图 2 $l_{\max}$与 S^* 的关系

3.3.3 带限制条件 $l_{\max}$下 b_{req}的取值

在上面的讨论中作者提出 2 个目标性能: $l(i)$应是磁盘物理块的整数倍和平滑性能参数 ΔL, 但有另一个参数 $l_{\max}$(表示 $\boldsymbol{L}$ 中最长的块) 没有讨论. 可以看出, ΔL 除了和算法有关外, 本质上取决于 b 的大小, b 越大, ΔL 性能越好, 同时 $l_{\max}$值也相应地就小. 由于服务器的资源有限, b 的取值受到限制; 另一方面, 在存储管理设计中, 常要求 $l_{\max}$(是 p 的倍数) 不超过某一值时, 对磁盘空间的有效利用特别是多流情况下的存取接纳计算十分重要(如以 $l_{\max}$为所

有的存储块分配同样长度的磁盘空间). 所以, 考虑在 l_{max}受限制时 b 的取值有着重要的意义. 下面给出在已知 s^* 和 l_{max}的条件下计算最小需求 b_{req}的算法 B.

算法 B

输入: $s^*(1), s^*(2), \cdots, s^*(N)$和 l_{max}.

输出: b_{req}.

计算各峰值超过 l_{max}的矩形中多出的部分$b(m)$及个数 M; {如图 3 中的阴影区, 不考虑谷值}

让 $m = M$;

Do

$m = m - 1$; {从最后开始计算}

If $b(m)$和 $b(m+1)$之间所夹的区域其数据量 $\sum(l_{max} - s^*(i))$小于 $b(m+1)$

让 $b(m) = b(m) + [b(m+1) - \sum(l_{max} - s^*(i))]$;

Until $m = 0$;

取 $b_M = \max\{b(m)\}$;

输出 $b_{req} = ([b_M/p] + 1)p$ 和 b_M;

结束算法;

算法 B 在 N 的线性时间内可以结束. 第 5 步中的加 1 运算是为了保证在整数块存取下 b_{req}中有多余 b_M 的数据.

3.3.4　在限制条件 l_{max}, b_{req}下的调度规划

在已知 S^*, l_{max}和最小需求 b_{req}的条件下, 若不考虑平滑性能作为评估指标, 则可以找出许多可行的规划 $\boldsymbol{L} = [l(1), \cdots, l(N)]$. 下面给出时间复杂度为 $O(n)$的一个 I/O 调度规划 ——算法 C.

算法 C

基本思想: 首先总是以 l_{max}作为调度单元, 多余的数据进缓冲区, 直到缓冲区满. 然后, 逐步按步长 p 减小调度单元, 使之满足一定的条件. 用 $\sum b$ 跟踪缓冲区中数据量的变化.

输入: $s^*(1), s^*(2), \cdots, s^*(N)$, l_{max}, b_{req}及 $b(m)$中的最大者 b_M.

输出: $l(1), \cdots, l(N)$.

$\sum b = 0$; 初始化 $\sum b$ 等.

For $i = 0$ To $N - 1$

$m = 0$

Do (T)

If $\sum b + (l_{max} - mp) - s^*(i) \leqslant b_{req}$

$l(i) = l_{max} - mp$, $\sum p = \sum p + l(i) - s^*(i)$, break;

Else $m++$;

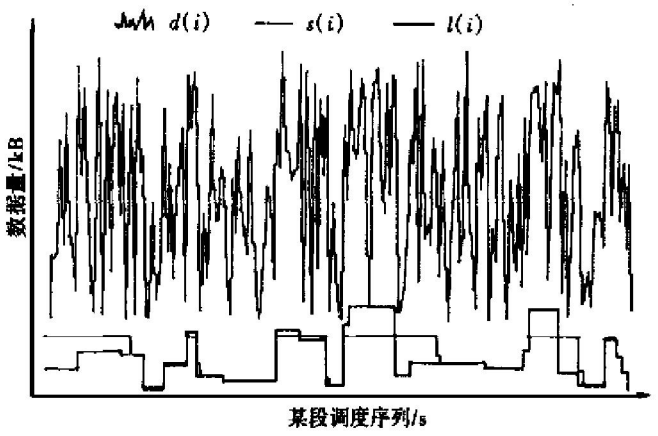
End For;

输出 $l(1), \cdots, l(N)$;

算法 C 中有 2 个循环: FOR 和 DO. FOR 循环和问题的规模有关; DO 循环和问题规模无关. 根据文献[3] 中实验数据, S^* 中最长和最短的块, 相差不超过 10 倍, 再加上 l_{max}的限制, m 和问题规模无关, 可视为常数. 所以, 算法 C 的时间复杂度为 $O(Cn)$, 等同于 $O(n)$, 在线性时间内可以完成.

4　结果与分析

实验中, 服务器的调度周期取为 1 s, 基本磁盘块取 2 kB. 提取了卡拉 OK 中的一段数据作为数据源, 其中 $d(i)$的取值范围是 0~ 198 kB, 平均速率为 93 kB/s; 应用文献[3] 中的平滑算法, 在 b 为 300 kB 时, 得到的 $s(i)$为 9~ 131 kB; 在 l_{max}为 110 kB 时, 应用算法 B 得到需要的最小缓冲取是 486 kB, 应用算法 C, 得到的调度规划结果如图 3 所示.



$d(i)$ —客户区数据消耗; $s(i)$ —服务器网络传输规划; $l(i)$ —磁盘 I/O 调度块规划

图 3　实验结果比较

经过算法 C 后, 磁盘的利用率为 100%. 就平滑性而言, 算法 C 不是一个好的算法. 从图 3 可以看出, 在缓冲区为空时它总是以最大的尺寸作为调度块, 否则, 在最小尺寸的 2 块之间跳转, 中间块极少, 但它的时间复杂度是线性的. 若考虑平滑性能, 则算法必须考虑回溯, 其时间复杂度一般在 $O(n^2)$ 以上.

参考文献:

[1] Wrege D, Knightly E, Liebeherr J, *et al*. Deterministic delay bounds for VBR video in packet-switching networks: fundamental limits and practical tradeoffs[J]. IEEE/ACM Trans Networking, 1996, 4(3): 352-362.

[2] Grossglauser M, Keshav S, Tse D N C. RCBR: a simple and efficient service for multiple time-scale traffic[J]. IEEE/ACM Trans on Networking, 1997, 5(6): 741-755.

[3] Salehi J D, Zhang Zhi-li, Kurose J, *et al*. Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing[J]. IEEE/ACM Trans on Networking, 1998, 6(4): 397-410.

[4] Duffield N G, Ramakrishnan K K, Reibman A R. SAVE: an algorithm for smoothed adaptive video over explicit rate networks[J]. IEEE/ACM Trans on Networking, 1998, 6(6): 717-728.

[5] Rexford J, Towsley D. Smoothing variable-bit-rate video in an internetwork[J]. IEEE/ACM Trans on Networking, 1999, 7(2): 202-215.

[6] Charny A, Ramakrishnan K K, Lauck A. Time scale analysis and scalability issues for explicit rate allocation in ATM networks[J]. IEEE/ACM Trans on Networking, 1996, 4(2): 569-581.

[7] Zhang Zhi-li, Kurose J, Salehi J, *et al*. Smoothing, statistical multiplexing and call admission control for stored video[J]. IEEE J Select Areas Commun., 1997, 15(8): 1148-1166.

[8] Zhang Zhi-li, Hui J. Applying traffic smoothing techniques for quality of service control in VBR video transmissions[J]. Computer Commun., 1998, 21(4): 375-389.

[9] Rosario J M D, Fox G C. Constant bit rate transmission of variable bit continuous media in video-on-demand servers[J]. Multimedia Tools Application, 1996, 1(2): 215-232.

[10] Yau D K Y, Lam S S. Adaptive rate-controlled scheduling for multimedia applications[J]. IEEE/ACM Trans on Networking, 1997, 5(4): 475-487.

[11] Kanakia H, Mishra P P, Reibman A R. An adaptive congestion control scheme for real time packet video transport[J]. IEEE/ACM Trans on Networking, 1995, 3(6): 671-682.

[12] Adas A M. Using adaptive linear prediction to support real-time VBR video under RCBR network service model[J]. IEEE/ACM Trans on Networking, 1998, 6(5): 635-644.

The optimizing schedule of the VBR stream disk I/O and buffer computing in special situation

XIE Jian-guo, CHEN Song-qiao, CHEN Jian-er

(College of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract: VBR compressed video can exhibit significant multiple-time-scale bit-rate variation, and this gives a challenge to the transport system, such as network and disk I/O. The optimal smoothing of VBR at present has become an important research area. To stored video, pre-buffering is the most efficient and feasible method. In terms of the basement of VBR stream smoothing schedule for gaining satiable network's QoS, in the paper we introduce the methods of VBR stream re-smoothing schedule and buffer computing which is based on the characteristics of disk storage system access to data in elementary physic disk-block. The paper first presents a disk I/O schedule algorithm that considers one elementary physic disk-block as buffer size, than gives another disk I/O schedule algorithm whose time complexity is $O(n)$ and buffer computing in the special situation where the access unit is not more than a value.

Key words: VBR; buffer; smoothing; algorithms