

视频服务器中多网络 I/O 的调度与接纳控制*

谢建国, 陈建二, 陈松乔

(中南大学 信息科学与工程学院, 湖南 长沙 410083)

E-mail: xjg@xtpu.org.cn; jgxie@163.net

http://www.csu.edu.cn

摘要: 在视频服务器中, 由于网络 I/O 带宽比磁盘 I/O 带宽相对丰富而未得到研究者的重视, 但当服务达到一定的规模、网络输出成为瓶颈时, 在多网络 I/O 情况下, 网络输出的量化计算则不可避免。着重探讨了视频服务器在多网络 I/O 并行输出的情况下, 如何进行最大输出调度的问题, 给出了 3 个调度算法及相应的接纳计算公式, 特别是一个带缓冲的优化调度模型。实验结果显示, 调度模型及规划性能是优越而有效的。该项工作对视频服务器的调度设计、资源的有效配置具有指导作用。

关键词: VBR 视频; 缓冲; 调度; LS 算法; 接纳控制

中图法分类号: TP316 文献标识码: A

连续媒体(continuous media, 简称 CM)或流媒体是指数字视频、声音、动画这类对延时敏感的数据媒体。储存和管理这类媒体的服务器叫作连续媒体服务器(即通常意义上的视频服务器), 其主要功能是媒体文件的存储管理、磁盘 I/O、缓冲及网口输出调度, 为用户提供 VoD(video on demand)服务。

自 20 世纪 90 年代初期, 流媒体技术一直得到研究者的重视。早期一些代表性文献的工作主要集中在视频存储、缓冲计算及磁盘调度策略上, 虽然涉及的问题相对简单, 问题的规模小, 但却做了很有益的开创性工作。文献[1]对这些工作进行了综述。近期的文献从不同的方面研究了流媒体技术, 主要包括磁盘调度与接纳控制^[2-6]、视频 Multicast^[7]、变位率视频流的带宽平滑技术^[8]等。

本文的研究领域属于视频服务器的缓冲调度问题, 具体讨论流媒体数据从内存经网卡向网络输出, 问题的背景是在多网卡或多网络 I/O 并行运行及轮转法服务调度的机制下, 考虑流媒体的实时、变位率特性, 引入多任务调度方法, 讨论如何进行流媒体输出调度、新流接纳及优化问题。以往的工作主要考虑磁盘 I/O 调度, 而网络输出带宽与磁盘输入带宽相比, 由于其在视频系统中资源相对丰富而被研究者所忽视, 我们首先在这方面作了详细的探讨, 本文提出的思想和方法不但对视频服务器的设计、资源的有效配置具有一定的指导作用, 而且对相关领域的研究也具有一定的借鉴意义。

1 工作模型

CM 服务器的基本功能模块分为 CM 文件的组织、磁盘检索和缓冲调度。对于具有磁盘 I/O 多路并行输入和网络多端口并行输出的计算机而言, 其缓冲调度的工作模型如图 1 所示。图中的数据块经磁盘 I/O 多路并行输入, 置入缓冲区后由网口 I/O 多路并行输出, 或直接由专用数据通道传经网口并行输出到路由或交换设备。对于工作模型, 我们给出如下几个概念:

* 收稿日期: 2001-02-06; 修改日期: 2001-07-16

基金项目: 国家自然科学基金资助项目(69928201)

作者简介: 谢建国(1964 -), 男, 湖南祁阳人, 博士, 副教授, 主要研究领域为网络多媒体, 网络优化; 陈建二(1954 -), 男, 广西桂林人, 博士, 教授, 博士生导师, 主要研究领域为计算机网络理论, 网络优化, 近似计算; 陈松乔(1940 -), 男, 湖南长沙人, 教授, 博士生导师, 主要研究领域为软件工程。

概念 1(调度周期 T). 令轮转法的调度周期为 T , 在一个调度周期内为每一路流调度一个数据块. 用 j 表示调度周期的顺序, $j \in \{1, 2, \dots\}$.

概念 2(数据块). 在这里是最小的、不可分割的调度单位, 是指一条路流在一个调度周期内所需的数据量, 如文献[2]中的 CTL 块、CDL 块. 用 l 标识它的长度.

概念 3(流). 来自同一数据源数据块的序列. 流在服务器端是数据块周期性地被调度, 如视频中的帧率. 用 i 标识某一路流, 一条路流中最长块用 l_i^{\max} 标识, l_{ij} 表示 i 流第 j 周期需要的数据块长度.

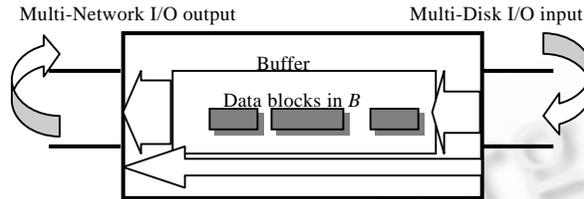


Fig.1 The work model for buffering and scheduling
图 1 缓冲调度工作模型

若服务器在第 j 个调度周期正在为 n_j 路流提供实时调度服务, 假定磁盘输入给出的块是及时的, 按照要求, 在每一个调度周期 j 内必须完成 n_j 个数据块的输出调度任务. 由于 3 个原因: 块是不可分割和压缩的; 在不同的调度周期内来自同一流的块其长度是变化的; 网口输出带宽资源有限. 为确保每一调度周期的完成, 应仔细规划. 下面就单网络 I/O 和多网络 I/O 两种情况来讨论数据块的调度和新流接纳问题.

2 单网络 I/O

在第 j 个调度周期, 设 CM 服务器的调度管理区正在为 n_j 路视频流进行调度, 对于同一流在不同的调度周期内, 其被服务的相对顺序可以不同, 如第 i 路流在前一周被首次调度, 在下一周期可以被最后调度(这要求客户缓冲区有足够两个周期内消耗的预缓冲数据). 令单个网络 I/O 的数据传输率是 R , 对于只有一个网络 I/O 出口的 CM 服务器来说, 当在第 j_s 个调度周期有新流 s 加入时, 理想服务模式下的接纳条件是式(1)(考虑 CPU 的处理速度很快, 不影响数据传输、控制信息和数据信息分道传输), 其中 M 表示最后一个流被服务完所需的周期数. 确保服务模式下的接纳条件是用 l_i^{\max} 代替所有的 l_{ij} 进行计算, 但不需要对 j 进行循环.

$$\sum_{i=1}^{n_j} l_{ij} + l_{sj} \leq R \times T, \quad j \in \{j_s, j_s + 1, \dots, j_s + M\}. \quad (1)$$

式(1)的计算量是确保服务模式下的 M 倍, 但比它能支持更多的流数. 一部电影按 2 小时计算, 调度周期取 1s, 对于能支持上千路流的服务器, 其接纳计算量在百万次, 在实时环境下是可以接受的.

3 多个网络 I/O

若 CM 服务器具有 p 个并行的网络 I/O 通道, 每一个网络 I/O 通道具有独立处理数据块的能力, 每一个数据块可被调度到任意一个网络 I/O 通道输出(如 CPU 负责块调度, 网络 I/O 通道处理数据组帧、打包等). 令 p 个网络 I/O 具有相同的数据传输率 R , 则一个网络 I/O 通道在 T 内的输出能力为 $L = R \times T$. 假定多磁盘 I/O 能及时地给出数据块, 在第 j 个调度周期内只要直观地考虑 n_j 个长度分别为 l_{ij} 的数据块, 能全部装入长度为 L 的 p 个箱中, 则说明 n_j 个数据块能在 T 时间内顺利地来自网络 I/O 通道输出. 这种情况等同于多个处理器中“一工作一处理器”的问题模型, 每一个通道可看做是一个处理器, 每一个长度为 l_{ij} 的数据块可看做是一个执行时间为 l_{ij} 的工作, 其目的是决定是否所给工作能在时间 L 内完成. 这个问题在文献[9]中称为 Makespan 问题, 它是一个 NP 问题.

3.1 Graham's LS算法的数学描述

在 n 个数据块无序的情况下, 采用文献[10]Graham 的 LS(list scheduling)非空闲处理算法, 新来的块总是被放入最空闲的通道中. 令数据块 r 为 LS 算法最后完成的数据块(即数据块 r 的完成时间是 H), 并假设数据块 r 的长度为 l_r . Graham's LS 算法的调度结果如图 2 所示, 由图 2 看出, 时间 $H - l_r$ 以内为非空闲, 在 l_r 以后被调度的

块其开始时间都比 l_r 的要晚,用数学公式表示有

$$H - l_r \leq \sum_{x=1}^{r-1} l_x / p,$$

得到

$$H \leq \sum_{x=1}^n l_x / p + l_r - \sum_{x=r}^n l_x / p.$$

令

$$\bar{H}(1, n) = \sum_{x=1}^n l_x / p, \quad \bar{H}(r, n) = \sum_{x=r}^n l_x / p,$$

有

$$H \leq \bar{H}(1, n) + l_r - \bar{H}(r, n).$$

其中 $\bar{H}(1, n)$ 表示通道的平均工作时间,只要 $H \leq L$,所有的块都能在 L 时间内被处理.式(2)说明,如果 H 的上限小于 L ,则所有的块在给定的时间内都能被处理.

$$\bar{H}(1, n) - \bar{H}(r, n) + l_r \leq L. \tag{2}$$

从图 2 可以得出事实 1:如果一个数据块是 l_r ,则说明在它后面被处理的所有块其长度都比 l_r 小,而且有 $l_r \geq (\sum_{x=1}^n l_x) / (p-1)$, x 表示调度的顺序数, $x \in \{1, 2, \dots, r, \dots, n\}$.

3.2 随机调度及新流接纳

假定服务器正在为 n_j 路流提供调度任务,用 $l_j^{\max} = \max\{l_{ij} | i \in [1, n_j + 1]\}$ 表示第 j 个周期中最大的块,当在第 j_s 个调度周期有新流 s 加入时,在 LS 策略下,其最坏的调度结果是 l_r 取 $l_{j_s}^{\max}$,且在调度周期内最后被调度,此时 $\bar{H}_j(r, n_j + 1) = l_j^{\max} / p$.理想服务模式下新流接纳的条件由式(2)得到式(3), M 含义同上.

$$\bar{H}_j(1, n_j + 1) + (p-1)l_j^{\max} / p \leq L, \quad j \in \{j_s, j_s + 1, \dots, j_s + M\}. \tag{3}$$

$\bar{H}_j(1, n_j + 1)$ 中的下标表示第 j 个调度周期中所有数据块之和对通道数求平均.若式(3)成立,就可以接纳新流,而与它在一个调度周期内被调度的次序无关.式(3)的时间复杂度为 $O(M \cdot n)$,同式(1).

4 优化的调度及新流接纳

条件式(2)有两个可变参数 n 和 l_r .为提高 n 的值,优化的原则是 l_r 越小越好,尽可能让较小的块成为 l_r .一个视频服务器可采用两种缓冲策略,一种是采用双缓冲结构,另一种是采用单缓冲结构.以下就不同的缓冲策略分别讨论并行调度和新流的接纳.

4.1 双缓冲机制下的调度及新流接纳算法

对于具有双缓冲区的 CM 服务器:一个缓冲区存放输出数据,另一个缓冲区存放输入数据.这样,在调度之前 n 个数据块就已预置入缓冲区,有条件实现先排序、后调度.采用文献[11]的 LPT(longest processing time first)算法调度,可以获得好而相对稳定的调度结果,而随机调度的结果是不稳定的.但 LPT 算法的时间复杂度是式(3)的 n 倍.根据上面提供的数字,LPT 算法的计算量达 G 的数量级.若有许多个新流同时提出申请,在实时环境下,要在一个调度周期内完成计算,就将耗掉过多的 CPU 资源.

4.2 单缓冲机制下的调度及新流接纳算法

在单缓冲机制下,磁盘输入的数据和等待网络输出的数据共享一个缓冲区,这种情况下数据块的到达是实时的、随机的.我们假定磁盘能及时给出数据块,且在共享缓冲区存在一块尺寸至少为 B 的数据滞留区(这可以通过一定的延时,错开磁盘 I/O 调度周期和网络 I/O 调度周期来实现),用来存放本调度周期无须立即发送的数据块,我们称其为缓冲箱 B (称其为缓冲箱是为了与缓冲区加以区别).上面提到,优化的方法就是让较小的块成为 l_r .在式(3)中,每次计算总是取本调度周期中最长的数据块作为 l_r ,且最后被调度.若在一个调度周期中将一些较短的块先放入缓冲箱 B 中,到本调度周期未被调度,不让长数据块成为 l_r ,这样可以减少 Makespan 的值,增加输出流量.

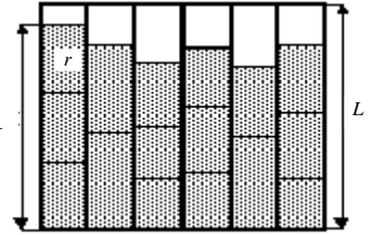


Fig. 2 The result of Graham's LS
图 2 Graham's LS 算法结果

假定缓冲箱 B 能容纳 m 个数据块,其中 m 是可变的,应满足条件 $\sum_{l=1}^{m-1} l x \leq B \leq \sum_{l=1}^m l x$. 对于缓冲箱 B , 设定一个阈值 l_B^{\max} , 规定只有不大于阈值 l_B^{\max} 的数据块才能进入 B 中. 同样, 令 l_j^{\max} 表示第 j 个周期中最大的块. 由事实 1 知, 在 j 周期当 B 中的数据总和 $B_j \geq (p-1)l_j^{\max}$ 时, 那么 l_r 一定是缓冲箱内的某一块; 若 B 中的块采用随机调度时, 在最坏的情况下取 $l_r = l_B^{\max}$ 且是最后被处理, 那么理想服务的新流接纳公式由式(4)表示. 当 B 中的块采用 LPT 算法调度时, 其实际结果比式(4)要好.

$$\overline{H}_j(1, n_j + 1) + (p-1)l_{B_j}^{\max} / p \leq L, j \in \{j_s, j_s + 1, \dots, j_s + M\}. \quad (4)$$

现在来讨论 B 和 l_B^{\max} 的取值问题. 各调度周期的 B 可能不相同, 为确保起见, B 应按式(5)计算, 取它的下限值, B 值越大时, 排序调度结果越好, 但对式(4)的性能没有提高.

$$B \geq (p-1) \max\{l_i^{\max} \mid i \in \text{all streams}\}. \quad (5)$$

l_B^{\max} 的取值由式(6)确定, 其中比例因子 λ_j 为小于 1 的正实数, 如 $\lambda_j = 0.5$ 等, λ_j 的具体取值可以由经验估计 (如小数据块所占的比例) 和计算得到, 见算法 A.

$$l_{B_j}^{\max} = \lambda_j \times l_j^{\max}, \lambda_j \in (0, 1). \quad (6)$$

式(4)的前提是 B 被填满, 但当没有足够的小块填充 B 时, 如何计算接纳条件? 用 σ 表示填充系数, B_j^{data} 表示第 j 个周期中数据块尺寸不大于 $l_{B_j}^{\max}$ 的所有块之和, 取 $B_j^{\min} = (p-1)l_j^{\max}$, 有 $B_j^{\text{data}} = \sigma B_j^{\min}$. 在 $B_j^{\text{data}} < B_j^{\min}$ 的情况下 (B 不能被填满), 那么成为 l_r 的可能是缓冲箱外的某一块, 或箱内的一块. 当 l_r 是缓冲箱外的某一块, 在最坏的情况下取 $l_r = l_j^{\max}$, 则箱中的 m 块必在 l_j^{\max} 之后被调度, 那么调度结果的上限值为式(7), 注意, 式(7)中要求 $B_j^{\text{data}} < B_j^{\min}$.

$$H_j \leq \overline{H}_j(1, n_j + 1) + l_j^{\max} - (B_j^{\text{data}} + l_j^{\max}) / p. \quad (7)$$

当 l_r 是缓冲箱内的某一块时, 式(4)仍然成立. 在 B 不满时, 为确保式(4)的结果, 要求式(7)的上限小于式(4)的左项, 有

$$\overline{H}_j(1, n_j + 1) + l_j^{\max} - (B_j^{\text{data}} + l_j^{\max}) / p \leq \overline{H}_j(1, n_j + 1) + (p-1)l_{B_j}^{\max} / p,$$

得到

$$\sigma B_j^{\min} \geq (1 - \lambda_j)(p-1)l_j^{\max},$$

得到

$$\sigma \geq (1 - \lambda_j). \quad (8)$$

由式(8)知填充系数 σ 和比例因子 λ_j 有着反比关系, λ_j 越小即进入缓冲箱的块尺寸小, 则缓冲箱越应填满. 因此, 当 λ_j 一定后, σ 满足式(8)时, 则新流的接纳可用式(9)进行计算, 接纳算法为 A, 便于算法书写方便, 引入符号 $\Delta L = L - [\overline{H}_j(1, n_j + 1) + \lambda_j(p-1)l_j^{\max} / p]$, 若 $\Delta L \geq 0$ 表示新流可以接纳.

$$\overline{H}_j(1, n_j + 1) + \lambda_j(p-1)l_j^{\max} / p \leq L, j \in \{j_s, j_s + 1, \dots, j_s + M\}. \quad (9)$$

算法 A.

输入: M 个调度周期及每个周期的所有块尺寸, 输出接纳结果.

1. $\forall \lambda_j = \lambda_0, \lambda_{\text{step}} = 0.05$; {初始化所有 λ 及 λ 变化的步长 λ_{step} , 它们的初值是可选的}
2. For $j=1$ to M
3. compute B_j^{\min} ;
4. While (loop==continue)
5. compute $B_j^{\text{data}}, \sigma$;
6. Case1: ($\sigma \geq (1 - \lambda_j)$ and $\Delta L \geq 0$) Admission=OK, loop=exit;
7. Case2: ($\sigma > (1 - \lambda_j)$ and $\Delta L < 0$) $\lambda_j = \lambda_j - \lambda_{\text{step}}$, loop=continue;
8. Case3: ($\sigma < (1 - \lambda_j)$ and $\Delta L > 0$) $\lambda_j = \lambda_j + \lambda_{\text{step}}$, loop=continue;
9. Case4: ($\sigma \geq (1 - \lambda_j)$ and $\Delta L < 0$) Clean (λ), Return (Admission=NO); {清除本次设置的 λ_j }
10. End while;
11. Record (λ_j); {记住 λ_j , 以后实际调度时要用到}
12. End for; Return (Admission=OK).

算法 A 中有 3 个循环: 一个是 For 循环, 循环次数不大于 M ; 另一个是 While 循环, 循环次数取决于步长 λ_{step} .

在 10 的数量级;主要一个是 ΔL 的计算,其计算量是 n .所以,算法 A 的时间复杂度是 $O(cM \cdot n)$,优于算法 LPT.

5 实验与性能比较

数据块特性:若轮转法调度周期的时间取 1 秒,则对于以 MPEG-1 恒质量压缩的视频流,其数据长度在 0~2Mb 之间;而对于以 MPEG-2 恒质量压缩的视频流,其数据长度在 2~8Mb 之间.模拟实验中的数据来自文献 [12],MPEG Trace 的名字为 atp(ATP Tennis Final 1994:Becker-Sampras).

5.1 n 个数据块的模拟调度比较

根据 atp 的数据特性,构造代表 200 流的一组数据块进行模拟实验,用来比较 3 个接纳算法的性能.数据块尺寸为 46KB~218KB.对这一组相同的数据块在不同的顺序下,按照 3 个算法进行模拟调度,图 3、图 4 显示了其中 50 次调度结果.注意,每一次调度,式(3)和算法 A 有相同的数据块到达顺序.实验结果显示,算法 A 有相当稳定的结果,与数据块被调度的顺序关系不大.

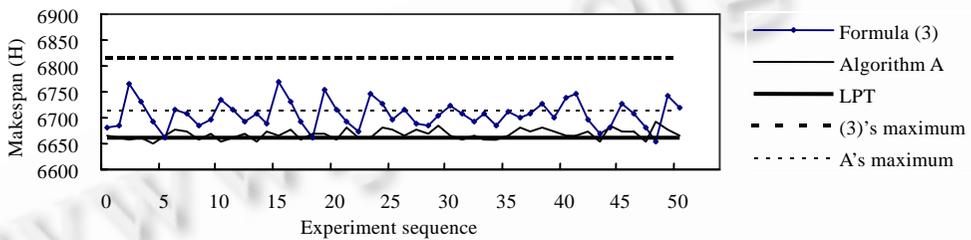


Fig.3 The scheduling results of three algorithms ($p=4, \forall \lambda=0.4$)

图 3 3 个算法的调度结果($p=4, \forall \lambda=0.4$)

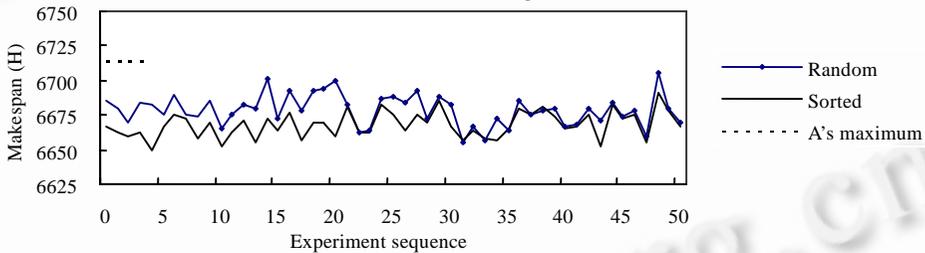


Fig.4 The scheduling results of data blocks in B by sort and random ($p=4, \forall \lambda=0.4$)

图 4 缓冲箱内数据块随机和排序调度结果($p=4, \forall \lambda=0.4$)

5.2 n 个数据流的模拟调度比较

atp 有 40 000 帧,略小于半小时的播放时间,GOP(group of picture)的尺寸是 12 帧,速率是 25fps.调度周期按一个 GOP 播放时间计算,GOP 块尺寸为 3.5KB~108KB,平均尺寸为 32.7KB,模拟过程是每一个周期加入一个新流(也就是 atp 中的第 1 块),直至达到算法的饱和条件,结果列于表 1 中.

表中式(3)和算法 A 的计算值是基于最坏情况下考虑的,所以模拟值好于计算值,但式(3)的模拟值不可靠,而算法 A 的模拟值相对要可靠得多,因为缓冲箱中数据块是按先大后小排序调度的.LPT 的算法实际上是一个模拟调度过程,算法的计算结果和模拟结果是一致的.

Table 1 The maximum of stream supported, $R=10\text{Mb/s}, \lambda=0.35\sim 0.5$

表 1 支持的最大流数 $R=10\text{Mb/s}, \lambda=0.35\sim 0.5$

p	2		3		4		6	
	Computing	Simulating	Computing	Simulating	Computing	Simulating	Computing	Simulating
Formula (3)	32	33	46	47	61	63	90	92
LPT	36	36	53	53	70	70	102	102
Algorithm A	35	36	51	52	68	69	99	101

结果, 计算上限值, 模拟值, 公式(3), 算法 A.

从图和表中可以看出,在算法 A 的算法下,当缓冲箱中的数据块排序调度时,有着和 LPT 相近的结果,而算法 A 的复杂性要好得多,所以,在双缓冲区机制下采用算法 A 进行接纳计算和调度,可以节约许多计算资源.另外,算法 A 和式(3)之间的流数差别取决于算法和块尺寸的分布,大块越大,差别也越大.网口带宽的利用率和 R 的大小及个数有直接关系,上述实验算法 A 的带宽利用率在 97%以上,而式(3)为 91%.

为了验证算法 A 的真实性,用 3 个 10Mb 的网卡,出口带宽总和为 30 Mb/s.考虑实验不能真正地并行(每发一个包产生两次 CPU 中断,要占用一定的时间资源),在内存和网口之间进行数据发送,测得只用到了 26M 带宽,扣除中断调用时间开销,约不到 1M 的资源由于块调度的特性而未被利用,而式(3)只用到 24M 带宽,与上述的模拟有类似的结果.

References:

- [1] Gemmell, D.J., Vin, H.M. Multimedia storage server: a tutorial. *IEEE Computer*, 1995,28(5):40~49.
- [2] Chang, E., Zakhor, A. Cost analyses for VBR video servers. *IEEE Multimedia*, 1996,3(4):56~71.
- [3] Kenchammana, D.R. I/O scheduling for digital continuous media. *ACM Multimedia System*, 1997,5(4):213~237.
- [4] Liu, Heng-zhu, Chen, Xu-can, Chen, Fu-jie. Study of servicing policies in video-on-demand server. *Journal of Software*, 1999,10(1):13~18 (in Chinese).
- [5] Wu, Fei, Ren, Hong, Chen, Fu-jie. Researches on theoretical model and buffer management of continuous media streams. *Chinese Journal of Computers*, 1999,22(1):39~44 (in Chinese).
- [6] Chiueh, T. The integration of real-time I/O and network support in the stony brook video server. *IEEE Network*, 1999,13(2):30~36.
- [7] Li, Xue, Ammar, M.H., Paul, S. Video multicast over the Internet. *IEEE Network*, 1999,13(2):46~60.
- [8] 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. *IEEE/ACM Transactions on Networking*, 1998,6(4):397~410.
- [9] Garey, M.R., Johnson, D.S. *Computer and Intractability: a Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.
- [10] Gramm, R.L. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 1966,45(9):1563~1581.
- [11] Gramm, R.L. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 1969,17(2):416~426.
- [12] <http://nero.informatik.uni-wuerzburg.de/MPEG/trace>. 1995.

附中文参考文献:

- [4] 刘衡竹,陈旭灿,陈福接.视频点播服务器中服务策略的研究.软件学报,1999,10(1):13~18.
- [5] 吴飞,任鸿,陈福接.连续媒体流的理论模型及缓冲对策研究.计算机学报,1999,22(1):39~44.

Scheduling and Admission Control in Video Servers with Multi-Network I/O*

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

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

E-mail: xjg@xtpu.org.cn; jgxie@163.net

<http://www.csu.edu.cn>

Abstract: Network I/O scheduling in video servers has been mostly neglected because in general the bandwidth of network I/O is richer than that of disk I/O. However, recent applications have indicated that when the scale of service increases to certain level, network I/O bandwidth will become a bottleneck and the problem of network I/O scheduling will be much more crucial. This paper focuses on scheduling video data blocks in video servers with multi-network I/O. Three scheduling algorithms and the corresponding admission control formulas are presented, especially a scheduling optimal scheme with buffers is proposed. Experimental results show that the scheduling scheme is effective and efficient. The results can be directly used in server network I/O scheduling and efficient configuration of server resources.

Key words: VBR video; buffer; scheduling; list scheduling algorithm; admission control

* Received February 6, 2001; accepted July 16, 2001

Supported by the National Natural Science Foundation of China under Grant No.69928201