一种基于周期合并策略的流调度算法*

向 哲, 钟玉琢, 冼伟铨

(清华大学 计算机科学与技术系,北京 100084)

E-mail: {xiangzhe,zhong,xian}@media.cs.tsinghua.edu.cn

http://www.tsinghua.edu.cn

摘要: 在视频点播系统中,流调度算法通过降低服务延迟和提高服务效率等措施,可显著提高系统服务能力.提出了一种新的流调度算法——PeriodPatch.该算法在Patching算法的基础上引入了周期调度的规则.PeriodPatch算法可以通过少数有序生成的组播节目流提供大量TVOD(true video_on_demand)级服务.PeriodPatch算法还保证系统在资源耗尽的情况下,可以提供高效、可预测的NVOD(near video_on_demand)级服务.仿真结果表明,PeriodPatch算法在TVOD服务下的系统资源消耗程度和在有限资源下的平均用户等待时间均优于其他算法.总之,PeriodPatch是一种高效、经济的流调度算法.

关键词: 流调度;周期;补丁流;VOD(video on demand)

中图法分类号: TP316 **文献标识码**: A

网络多媒体是计算机科学技术发展的重要发展方向,流调度策略及其算法是网络多媒体的组织管理核心.在VOD(video on demand)系统中,流调度算法主要解决合理调度视频服务器和网络资源、优化资源配置、实现服务能力最大化的问题.在大型VOD系统中,合适的流调度算法能够带来服务能力数量级的提高.因此,它具有非常明显的经济价值.

典型的TVOD(true video_on_demand)基于FIFO(first in first out)策略,典型的NVOD(near video on demand)则基于周期播放策略.上述策略均为固定策略.与之对应的主动策略包括Batch策略[1]、客户端缓冲策略[2]、adaptive piggybacking策略[3]等.基于主动策略,Yu提出了前向调度算法[4],Liao提出了SMP算法[5,6],Hua提出了Patching算法[7].这些算法充分利用系统资源,显著提高了系统性能,但它们仍有缺陷.本文结合Batch、客户端缓冲等主动策略,同时还吸收了固定策略中的周期调度思想,提出了PeriodPatch算法.基于该算法的大型VOD系统,性能和服务能力明显优于已知其他算法.同时,该算法还使系统随外部条件的变化而在TVOD和NVOD两种服务模式之间进行切换.PeriodPatch算法是一种高性能的自适应流调度算法.

本文第1节介绍流调度算法的系统模型.第2节描述PeriodPatch算法的流程,并介绍算法的特点.第3节对PeriodPatch算法进行性能分析.最后对算法作出总结和评价.

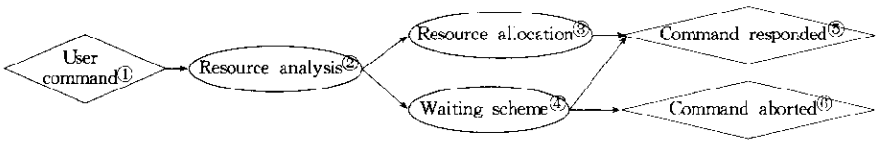
1 流调度算法的系统模型

在VOD系统中,用户向视频服务器发出命令,服务器接收命令后根据一定的规则作出应答,这一过程我们称之为一次系统服务操作SSO(system service operation),如图1所示.

* **收稿日期**: 2000-01-21; **修改日期**: 2000-04-06

基金项目: 国家自然科学基金资助项目(69973025);国家重点基础研究发展规划973资助项目(G1999032700)

作者简介: **向哲**(1975—),男,湖北荆州人,博士生,主要研究领域为网络多媒体,多媒体系统;**钟玉琢**(1938—),男,辽宁沈阳人,教授,博士生导师,主要研究领域为多媒体计算机,多媒体通信;**冼伟铨**(1954—),男,广西南宁人,副教授,主要研究领域为多媒体应用系统.



①用户命令,②资源分析,③资源分配,④等待策略,⑤响应命令,⑥放弃命令.

Fig. 1 System service operation

图1 系统服务操作(SSO)

根据功能上的差异,用户命令包括{Start,Stop,Pause,Resume,JumpForward,JumpBackward}等,这些命令构成了VCR(video cassette recorder)命令集.但VCR命令集不能反映媒体流的实际分配和销毁状况.因此,我们构造流操作命令集:

ApplyStream(MovieID,MovieTime),

TerminateStream(StreamID).

命令ApplyStream(AS)申请流数据的服务,其参数MovieID为影片标识;MovieTime为流的开始影片时刻.系统成功响应AS命令,将分配一条媒体流,用户记录该流的标识StreamID.命令TerminateStream(TS)退出媒体流,其参数StreamID为申请退出的流标识,系统根据情况决定是否销毁该媒体流.

VCR命令集和流操作命令集可以互相表示,Start和Resume对应于AS命令;Stop和Pause对应于TS命令;JumpForward和JumpBackward对应于TS和AS的命令组合.从本文后续章节可以看出,流操作命令集将极大地简化流调度算法的逻辑表示.

流调度算法负责SSO过程中资源分析、资源分配和等待策略的处理.资源分析查询当前的系统资源是否满足新的申请命令;资源分配为申请命令调度资源;如果系统资源不够充分,申请服务的用户将进入等待状态,等待状态的处理构成等待策略集.典型的等待策略包括有限强制等待、无限强制等待、用户自主等待、用户和系统交互决策等.

典型的TVOD系统采用FIFO流调度策略,用户发出AS命令,系统查询资源状况.如果资源充分,则为该用户分配一条流资源;否则用户退出系统.FIFO型系统易于实现和管理,但会严重浪费系统资源,且服务能力有限,对于大型VOD系统来说,经济成本过高.因此,设计流调度算法要设法解决资源的有效利用问题.

2 PeriodPatch算法

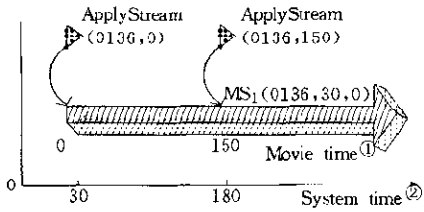
2.1 PeriodPatch算法策略

我们设计了PeriodPatch流调度算法,该算法基于以下策略:周期播放策略、Patch策略、客户端缓冲策略等.这些策略的有机组合使PeriodPatch算法在大型VOD系统中的性能优于其他流调度算法.

PeriodPatch算法定义了两种媒体流:主流(main stream,简称MS)和补丁流(patching stream,简称PS).这两种媒体流的功能是不同的,但在资源消耗上没有任何区别.这两种流的表示方法为:主流MS{MID,ST,MT}.MID表示影片ID;ST表示生成流的系统时间;MT表示流在ST时刻的影片时间.补丁流PS{MID,ST,MT,ET}.MID,ST,MT的定义与主流相同,ET表示流结束时的影片时间.

主流可供多用户复用,需要网络支持Multicasting等技术,本文不再赘述.一般而言,当第1个

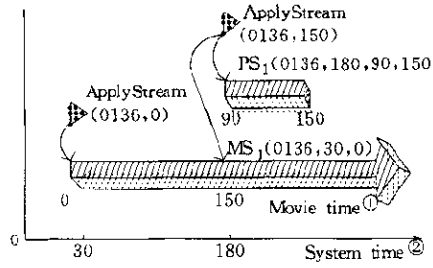
系统用户发出 AS 命令时,系统为该用户分配一条主流.此后,其他用户申请服务,系统总试图寻找能够合并的主流,如图 2 所示.用户 A 在系统时间 30 秒时发出命令 AS(0136,0),系统生成 MS_1 {0136,30,0}.在系统时间 180 秒时,用户 B 发出命令 ApplyStream(0136,150).此时,MS_1 的影片时间恰好为 150 秒.因此,系统不为用户 B 分配新资源,而是通知其加入流 MS_1,这样就实现了两个 AS 命令之间的流合并.



①影片时间,②系统时间.

Fig. 2 Batching stream

图2 流的合并



①影片时间,②系统时间.

Fig. 3 Patching stream

图3 补丁流

显然,上述流合并的发生概率非常小.因为新命令要求原来主流精确播放到某个确定位置,这种情况在实际系统中很少发生.我们采用了用户端缓冲和补丁流策略,以扩大主流接纳新用户的范围.用户端有一个数据缓冲区,可以缓冲从网络接收的一段媒体流数据.如图 3 所示,系统中有一条主流 MS_1 {0136,30,0},在系统时间 180 秒时,系统接受命令 AS(0136,90).此时,MS_1 影片时间为 0+(180−30)=150 秒.如果系统强制新用户加入主流 MS_1,那么该用户将丢失影片时间[90,150]之间的数据片断.我们采取如下措施:

(1) 新用户并入主流 MS_1,但不立即播放数据,而将数据暂存在用户端缓冲区内.

(2) 系统生成补丁流 PS_1 {0136,180,90,150},用户加入补丁流,并开始播放补丁流数据.这个过程持续到系统时间 240 秒,一共 60 秒.

(3) 系统时间到达 240 秒,补丁流结束.此时,用户缓冲区内为影片时间[150,240]之间的数据.用户开始播放缓冲区中的数据.

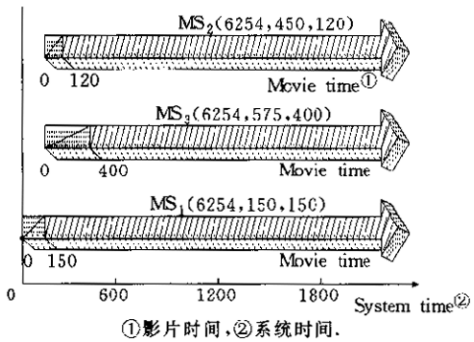
(4) 系统时间 240 秒后,用户继续从主流 MS_1 中接收数据并存储于缓冲区,同时仍从缓冲区中读取数据播放,缓冲区内数据保持 60 秒大小.

上述措施保证用户在获得实时点播服务的同时,独占流资源(补丁流)的时间仅为 60 秒,当补丁流传输完毕后,该用户使用流 MS_1 以节省系统资源.

若仅采用上述策略,还存在一个问题:主流分布无序.如图 4 所示,按照一定时序系统可能生成图中情况,3 条主流均播放影片 6254,但它们的影片时间非常接近.影片流密度过大,同样是对系统资源的浪费.在 PeriodPatch 算法中我们规定,主流必须符合周期性规范:相同影片 ID 主流影片时间的间隔是一个周期的整数倍.在算法实现上,我们对主流 MS{MID,ST,MT}规定:ST = MT(mod Period),这里,Period 是我们定义的周期常数.一般地,我们选择用户缓冲区大小作为这个周期常数.

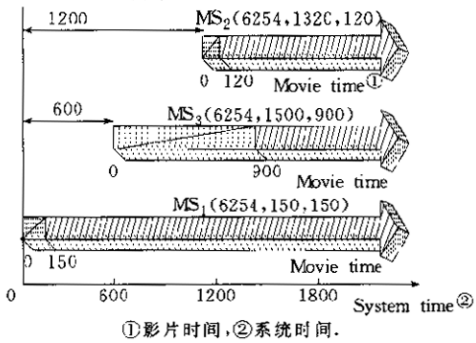
如图 5 所示,系统周期常数(用户缓冲区大小)为 600 秒,系统中原有主流 MS_1 {6254,150,150}.系统时间 1320 秒,系统接到用户命令 AS(6254,100),此时系统不生成主流 MS{6254,1320,100}服务用户,而是生成一条符合周期性规范的主流 MS_2 {6254,1320,120}.用户命令同 MS_2 的时差由补丁流 PS_2 {6254,1320,100,120}填充.周期性规范保证每个影片生成不同主流的影片时间间

隔固定,主流的密度分布均匀,且每部影片主流数目有上限,有利于节约系统资源.当系统的资源耗尽后,由于相同影片的媒体流分布均匀,因此,系统近似退化为周期播放的NVOD系统.



①影片时间,②系统时间.

Fig. 4 Random MainStream
图4 分布不均匀的主流

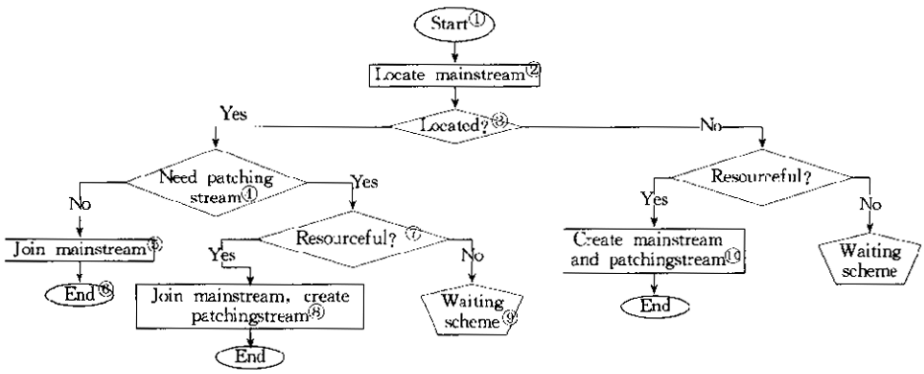


①影片时间,②系统时间.

Fig. 5 MainStream under PERIOD rule
图5 周期规则下的主流

2.2 PeriodPatch 算法描述

PeriodPatch 算法的流程如图 6 所示.



①开始,②定位主流,③成功定位,④需要补丁流,⑤加入主流,⑥结束,⑦资源允足,⑧加入主流,创建补丁流,⑨等待策略,⑩创建主流和补丁流.

Fig. 6 Flowchart of PeriodPatch algorithm
图6 PeriodPatch算法流程

主流定位是指系统在接收AS命令之后,搜寻是否有可合并主流.假设系统用户缓冲区的大小为 T_{buf},命令 AS(MovieID,MovieTime)在系统时间 T 到达系统,此时可并入的主流 MS{MID,ST,MT}必须满足:

$$\begin{cases} \text{MS.MID}=\text{AS.MovieID}, & (1)\\ \text{MS.ST}=I, & (2)\\ \text{AS.MovieTime}\leqslant\text{MS.MT}<\text{AS.MovieTime}+T_{buf}, & (3)\\ \text{MS.ST}=\text{MS.MT}(\bmod T_{buf}). & (4) \end{cases}$$

由式(2)~(4)可得,

$$\text{MS.MT}=T-\lfloor(T-\text{AS.MovieTime})/T_{buf}\rfloor\times T_{buf}. \quad (5)$$

显然,符合上述条件(1)、(2)、(5)的主流有且仅有一个.如果系统没有定位到符合条件的主流,新生成的主流也必须满足上述条件.

如果定位到或新生成的主流相对用户命令有时差,系统将创建补丁流.补丁流 PS{MID,ST,MT,ET}必须满足:

$$\begin{cases} PS.MID = AS.MovieID, \\ PS.ST - T, \\ PS.MT = AS.MovieTime, \\ PS.ET = MS.ET. \end{cases}$$

等待策略处理系统资源耗尽后的等待事件.我们为系统设定临界时间(deadline time).在临界时间内,用户命令保留在系统.如果有空余资源,则系统为用户服务;若超过临界时间,则用户被迫退出系统.

2.3 PeriodPatch 算法特点

PeriodPatch 算法的突出特点是:在主动策略的基础上引入周期播放思想.当系统资源足够充分时,系统为 TVOD 系统,PeriodPatch 算法与其他算法性能相同.而当系统资源完全耗尽后,系统中的热点节目更趋向于周期化分布,即趋向于典型 NVOD 系统.热点节目是系统的服务主部,因此,系统服务也接近于 NVOD 系统.所以,PeriodPatch 算法有 TVOD 和 NVOD 的自适应性.

PeriodPatch 算法的周期性规范保证系统资源的共享度优于其他算法,因此,PeriodPatch 算法的边界服务能力(系统首次出现资源瓶颈现象时的服务能力)大于其他算法.比较图 4 和图 5 可以发现,PeriodPatch 算法生成周期间隔的主流可覆盖更大的服务区间.

3 PeriodPatch 算法的性能分析

我们通过流调度算法验证平台评测 PeriodPatch 算法的性能.我们用用户行为仿真器生成相同的用户行为样本,供不同的算法作比较.改变外部参数生成不同的用户行为样本,可以评测 PeriodPatch 算法的适应性.

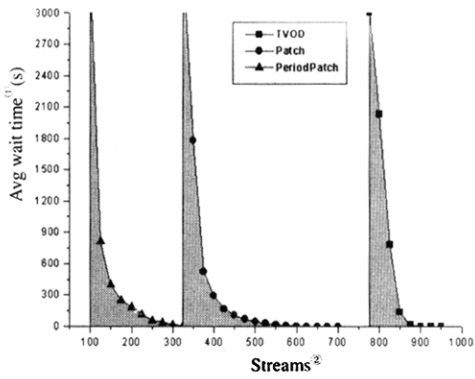
3.1 PeriodPatch,Patch 和典型 TVOD 算法的性能比较

我们使用的用户行为样本如下:系统有 50 部标准 MPEG-1 影片,用户对这些影片的点播服从 $\theta=0.271$ 的 Zipf 分布;用户访问系统的平均强度为 0.2 人/秒,用户到达时间服从 Possion 分布;每个用户自由发出 VCR 命令操作,VCR 命令操作的发生强度为 6 次/小时,发生时间服从 Possion 分布.我们模拟了 20 个小时的用户行为,生成的用户行为样本共有 16 162 个.其中平均有 709 个用户同时在系统中,平均有 675 个用户同时收看节目,最多有 925 个用户同时收看节目.

我们分别构造了典型 TVOD(基于 FIFO 流调度算法)、基于 Patch 流调度算法和基于 PeriodPatch 流调度算法的模拟系统,运行用户行为样本.我们衡量系统性能的参数为用户平均等待时间.

图 7 为无限等待策略下 3 种算法性能的比较.从图中可以看出,典型 TVOD 系统由于采用 FIFO算法.其边界资源消耗约为 900 条流,因为系统内最多有 925 个用户同时收看节目,因此必须有 925 条流资源才能使用户平均等待时间为 0.Patch 算法明显优于 FIFO 算法,大约 600 条流资源就能为最多 925 个用户提供无延迟的服务,资源复用提高了系统的性能.而 PeriodPatch 算法的性能最高,它的边界服务资源为 400 条流资源,仅仅为 Patch 算法的 2/3.这说明周期性规范强化了资源复用率.

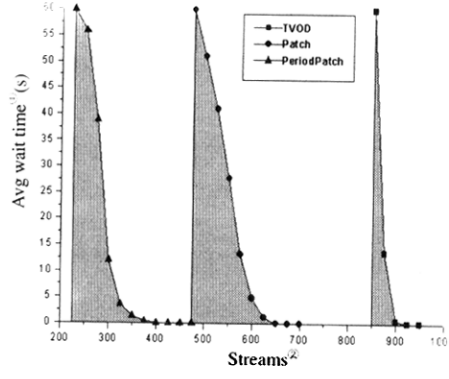
图 8 为有限等待策略下 3 种算法性能的比较,等待时间为 5 分钟.我们假设 30 秒是用户能够容忍的上限,在这个条件下,PeriodPatch 系统仅需约 275 条流资源就能提供有效服务,相应地,Patch 系统和典型 TVOD 系统则分别需要 550 和 875 条流资源.



①平均等待时间,②流数.

Fig. 7 Performance in unlimited waiting scheme

图 7 无限等待策略下的算法性能



①平均等待时间,②流数.

Fig. 8 Performance in limited waiting scheme

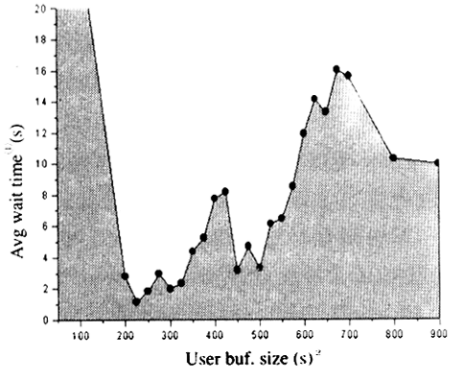
图 8 有限等待策略下的算法性能

3.2 PeriodPatch 算法适应性

算法适应性是衡量算法性能的重要指标. 如果系统的性能对外部条件非常敏感,仅适合某些特定条件下的应用,那么该算法的实际意义就不会很大.

图 9 反映了 PeriodPatch 算法对用户缓冲区大小变化的适应性. 我们的测试条件同上,仅将系统资源固定为 300 条流资源. 试验表明,当用户缓冲区小于 100 秒时,系统性能很差. 这是因为缓冲过小导致 PeriodPatch 算法主流定位的搜寻范围过窄,流合并的机会相应减小,PeriodPatch 算法趋向于 FIFO 算法. 试验显示,当用户缓冲区在区间[200,350]和区间[420,600]以内时系统性能较好,而设计过大的用户缓冲区不仅在经济上浪费,而且降低了系统的性能.

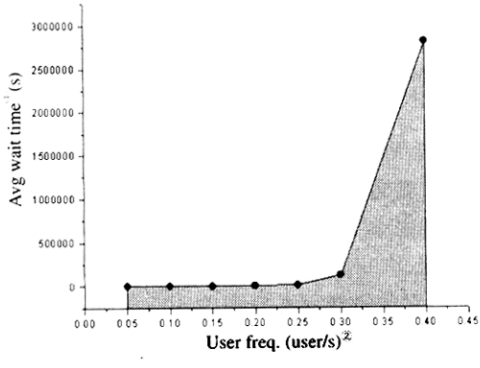
图 10 反映了 PeriodPatch 算法对用户访问强度的敏感程度,试验条件同前. 我们发现敏感曲线明显分为两段. 当用户强度小于 0.30 人/秒时,系统性能稳定;否则系统过载,系统性能急剧下降. 这说明,系统对用户强度有一个服务能力上限,即 0.30 人/秒. 在实际应用中,应采取许可控制协议等方法防止过载对系统整体性能的影响.



①平均等待时间,②用户缓冲区大小(秒).

Fig. 9 Effect of client buffer size

图 9 用户缓冲区对算法的影响



①平均等待时间,②用户访问频率(用户/秒).

Fig. 10 Effect of user visiting frequency

图 10 用户访问强度对算法的影响

4 结 论

在大型 VOD 系统中,PeriodPatch 是一种有效的流调度算法.算法在 Patch 策略的基础上引入周期性规范,强化了媒体流的共享程度;同时,算法根据资源消耗的程度,自动调节系统的服务模式,在 TVOD 和 NVOD 之间进行切换.仿真试验表明,PeriodPatch 算法的性能明显优于其他同类算法,且具有相当的适应性.

References:

[1] Dan, A., Sitaram, D., Shahabuddin, P. Scheduling policies for an on-demand video server with batching. In: Proceedings of the ACM Multimedia'94. New York: ACM Press, 1994. 15~23. ftp://ftp.sys.toronto.edu/upload/hui/acm94.ps.

[2] Almeroth, K.C., Ammar, M.H. The use of multicast delivery to provide a scalable and interactive video-on-demand service. IEEE Journal of Selected Areas in Communications, 1996,14(6):1110~1122.

[3] Colubchik, L., Lui, J.C.S., Muntz, P.R. Reducing I/O demand in video-on-demand storage servers. In: Proceedings of the ACM SigMetrics'95. New York: ACM Press, 1995. 25~36. http://www.cs.dal.ca/smedley/DM-MM-web/grad/Presentations/vod.pdf.

[4] Yu, P.S. Design and analysis of a look-ahead scheduling scheme to support pause-resume for video-on-demand application. ACM/Springer Multimedia Systems, 1995,3(4):137~150.

[5] Liao, Wan-jiun, Vitor, O.K.Li. The split and merge protocol for interactive video-on-demand. IEEE Multimedia Magazine, 1997,4(4):51~62.

[6] Vitor, O.K.Li, Liao, Wan-jiun. Peoformance model of interative video-on-demand systems. IEEE Journal of Selected Area in Communications, 1996,14(6):1099~1109.

[7] Hua, K.A., Cai, Ying, Sheu, Simon. Patching: a multicast technique for true video-on-demand services. In: Proceedings of the ACM Multimedia'98. New York: ACM Press, 1998. 35~43. http://www.dsg.cs.ucf.edu/papers/patching/patching.html.

A Stream Scheduling Algorithm Based on Period-Patching Strategy*

XIANG Zhe, ZHONG Yu-zhuo, XIAN Wei-quan

(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

E-mail: {xiangzhe,zhong,xian}@media.cs.tsinghua.edu.cn

http://www.tsinghua.edu.cn

Abstract: Stream scheduling algorithm used in video-on demand (VOD) system can largely increase system service ability by eliminating the service latency and improving the system efficiency. In this paper, a novel stream schedule called PeriodPatch is proposed. On the basis of Patching, a PERIOD rule is introduced, so that multicast streams are created orderly and fewer multicast streams are needed for the true video-on-demand (TVOD) service. Furthermore, PeriodPatch schedule ensures that the system can provide near video-on-demand (NVOD) service with a predictive and acceptable latency to the client if resource are exhausted out. The simulation results show that PeriodPatch is more efficient than others, with respect to both system resource required for TVOD service and average client waiting time (service latency) for a fixed available resource. In conclusion, the PeriodPatch is an efficient and economical stream schedule for VOD system.

Key words: stream-scheduling; period-criterion; patching stream; video-on-demand

* Received January 21, 2000; accepted April 6, 2000

Supported by the National Natural Science Foundation of China under Grant No. 69973025; the National Grand Foundamental Research 973 Program of China under Grant No. G1999032700