

用于 VOD 系统的块重组策略及分组调度算法^{*}

黄建忠 谢长生 傅湘林 方志鹏

(华中科技大学计算机学院外存储国家重点实验室 武汉 430074)

Block Rearrangement Policy and Grouped Scheduling Algorithms for VOD System

HUANG Jian-Zhong XIE Chang-Sheng FU Xiang-Lin FANG Zhi-Peng

(HuaZhong University of Science & Technology, Wuhan 430074, China)

Abstract With the advances in storage subsystems, compression and network technology, video-on-demand (VOD) is increasingly becoming an important services, the investigation to improve the performance of VOD servers has become a major focus. In this paper, we propose two rearrangements of data block and a request scheduling algorithms. With this approach, the data blocks with comparable access frequency are kept closer to each other, so as to reduce the moving distance of disk head and the seeking time. The simulation result demonstrates that the strategy and algorithms can effectively reduce disk access time and response latency respectively.

Keywords Block rearrangement strategy, Request scheduling algorithms, Response latency

1 引言

随着存储子系统、数据压缩技术、网络带宽各方面技术的发展,视频点播(VOD)得到了极大的推动。VOD的本质是用户根据自己的需求去主动选择所感兴趣的信息,这种主动性和选择性使得它在广告宣传、信息查询、娱乐、教育等领域变得前景广阔。国内外一些知名大学和公司已在这方面进行了深入的研究^[1]。

建立一个实用的 VOD 环境,其设计及研究重点是存储系统和网络系统^[2]。衡量 VOD 服务器性能的主要指标是保持特定服务质量(QoS)的同时所能达到的并发流数目,VOD 服务器性能越高,获得的并发流数越多。研究主要集中于:数据条块化^[3]、数据块分配^[4]、高速缓存^[5]和数据复制及迁移等。本文采用两种动态的数据块重组策略及分组请求调度算法来提高 VOD 服务器性能。

由于存取时间较长和高速缓存机制相对低效,磁盘被认为是 VOD 服务器的主要性能瓶颈。将数据条块化并分布在多个磁盘上,虽然能提高 I/O 带宽,但单个磁盘的服务时间仍旧是提高服务器整体性能的限制因素。

租赁模式(Rental Pattern)指出:对影片的访问是高度局部化的,即绝大多数的访问是集中在少量的影片上^[6]。利用这种局部性,我们提出两种动态的数据块重组策略来缩短磁盘访问时间。所谓“动态”,是指系统用该策略监视各视频的点播情况,并依据点播情况在适当的时候动态地对每个磁盘上的数据块进行重组。

针对各种策略,我们利用仿真实验进行了模拟,通过比较得到最佳的方案。仿真实验是在 IBM18es 磁盘阵列上进行的, DiskSim Simulator^[7]被用于仿真磁盘阵列的磁盘单元。为保证仿真的高精确度,特定寻道距离的寻道时间都使用实际数据。

2 数据块重组策略

在 VOD 系统中,用户请求是随机产生的。在一段相当长

的时间内,对一个足够大的用户群进行分析,可见它符合一定的分布规律。目前最常用的模型是 Zipf's 分布,Zipf's 分布指对 M 部影片按点播频率从高到低顺序排列,点播第 n 部的概率是 C/n ,其中 $C=1/(1+1/2+1/3+\dots+1/M)$ 。按照这种分布,点播第 5 部影片的频率是第 1 部影片频率的 $1/5$ 。

我们提出的动态块重组策略就利用了视频对象的存取局部化特性,将点播频率相近的视频块存放在一起,当系统负载较轻时触发重组操作。开始时,我们给每部影片指定一个受欢迎度,随着系统的运行,系统记录下视频的点播情况,从而修改视频的受欢迎度。

给出的两种重组策略是:中央布局(Centered-Layout,以下简称 CL)和边缘布局(Edged-Layout,以下简称 EL)。CL 将视频数据按照其访问频率存放在磁盘上,访问频率越高,存放位置越接近盘片中央;而 EL 则相反,按访问频率的高低从盘片边缘开始存放。

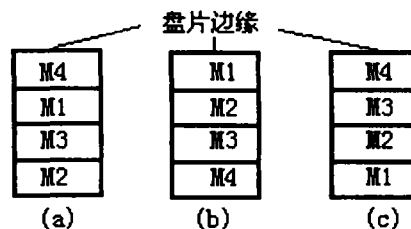


图1 组织方式

(a)随机;(b)边缘;(c)中央。

为便于比较,我们考虑了第三种选择,即随机布局(Random-Layout,以下简称 RL),RL 指视频数据被随机存放在磁盘上。图1给出数据块在一个磁盘上的不同放置。磁盘上存放 4 部影片的数据块, $P_{M1} > P_{M2} > P_{M3} > P_{M4}$, P_{Mi} 指影片 M_i 的受欢迎度。

选择 CL 重组策略时,根据视频对象的存取局部性,用户点播的视频文件大多数靠近磁盘的中央区域,因此磁头移动

^{*} 本文受国家自然科学基金项目(NO. 60173043)资助。黄建忠 博士生,研究方向:多媒体信息存储。谢长生 教授,博士生导师,研究方向为:多媒体信息存储、计算机新型存储系统结构、现代光盘存储技术、计算机设备评测技术与仪器。

到那些区域(柱面)的距离就短得多,从而缩短了寻道时间和寻道距离。后面的仿真结果显示,两种重组策略能明显提高磁盘的各项性能指标。

视频对象条块化的大小对上述策略会产生影响,粗粒度条块化能提高磁盘带宽,但是加长了磁盘访问时间。不过,条块化大小对各重组策略产生的影响相似。

3 分组调度算法

早期的调度算法集中于缩短寻道时间和寻道距离,如最短寻道时间优先算法(SSTF)和扫描算法(SCAN)。而 C-SCAN 算法和 LOOK 算法都是改进后的 SCAN 算法,C-SCAN 指磁头到达盘片边缘/中央时,跳到中央/边缘再进行移动;LOOK 算法指当请求的数据不在扫描方向上时,立即改变扫描方向。流媒体请求调度算法中,纯粹的 SCAN 算法很显然不合适,因为它会导致请求超时;最短期限优先算法(EDF)这方面是个不错的选择,但由于它没有缩短寻道时间,其性能较差。

目前,绝大多数算法都采用了循环调度的思想:每个视频对象被分成小块,每块的回放时间是固定的(如 1 秒钟),调度器循环地响应每个请求(即读取数据块)。在一个循环周期内,存储服务器必须读取所有请求的下一数据块,否则某些用户的视频回放就会产生抖动。采用循环机制的扫描算法虽能避免请求超时,然而必须以高启动延迟为代价。

我们上文提出的块重组策略中,数据块是按照其点播频率顺序存放的,而上述算法针对的数据块是随机存放的。为此,我们提出了一种分组调度(Grouped Scheduling,以下简称 GS):首先将视频对象按照其访问频率进行分组,组之间采用循环调度(round-robin),对组内的数据块采用 SCAN 调度。这样的话,当总共只有一个组时,分组调度就变为扫描调度。但每个组里只包含一个视频对象时,分组调度就变为循环调度。GS 算法的基本思想是对访问频率不同的视频对象区别对待。从图 3 可看出,GS 算法能降低请求响应延迟。

循环周期必须大于、等于每个视频数据块的播放时间,否则视频请求会得不到服务而产生饥饿现象。允许控制算法(Admission Control Algorithms)决定得到服务的请求数目,从而保证在循环周期内请求都能得到服务。在扫描算法中,循环周期约等于启动延迟,如图 2 所示,其中 R_i 指第 i 个请求,各请求之间的相互顺序就是数据在磁盘上的存放顺序。

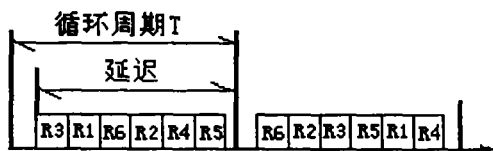


图 2 SCAN 算法中的响应延迟

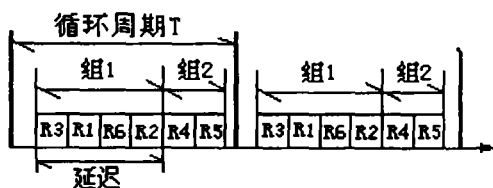


图 3 GS 算法中的响应延迟

如图 3 所示,在 GS 算法中,组内采用了扫描算法。当一组对象全部读取完毕,即可响应用户的播放请求,从而降低了响应延迟。根据 80/20 法则,请求响应延迟至少能降低 20%。

4 性能评价方法

我们将讨论负载特性和仿真环境。

4.1 负载特性

我们假设到达服务器的请求遵循泊松分布(Poisson Distribution),其平均到达率为 λ ;到达间隔时间遵循指数分布(Exponential Distribution),其均值 $T=1/\lambda$ 。用户对视频对象的访问遵循 Zipf's 分布,而视频全部采用 MPEG-II 压缩方式,数据率是 3Mbps,视频长度是 90 分钟。

4.2 仿真环境

DiskSim Simulator 用于仿真磁盘阵列中的磁盘单元,是目前所有仿真器中仿真精度最高的。仿真时,分别生成了基于三种策略(CL,EL,RL)的负载。磁盘是 IBM18es 这个型号的,表 1 给出了磁盘的一些重要参数。视频对象被条块化(striping),条块的大小为 1.5Mb,即播放时间为 0.5 秒。

表 1 磁盘 IBM18es 的主要参数

磁盘容量(GB)	9.17
转速(RPM)	7,200
扇区总数(即块数)	17,916,240
柱面数	11,474
盘面数	5
缓冲段数目	7
每柱面平均扇区数	1561
缓存段大小(块)	511

5 仿真结果

为评判各种算法和策略的优劣,实验过程中引入了扫描算法。下面我们将在磁盘数目和到达间隔时间变化的情况下获得平均寻道时间、平均寻道距离、平均磁盘访问时间,其间采用了各种策略和算法。

5.1 磁盘数目的影响

我们研究在三种不同重组策略和两种不同调度算法的情况下磁盘数目对平均寻道时间、平均寻道距离、平均磁盘访问时间的的影响。磁盘的数目从 4 变到 20。仿真结果见图 4~6。

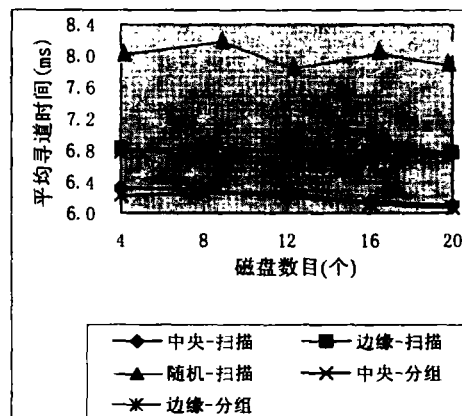


图 4 磁盘数目对平均寻道时间的影响

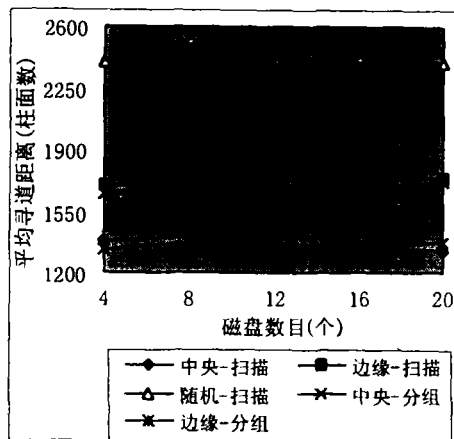


图5 磁盘数目对平均寻道距离的影响

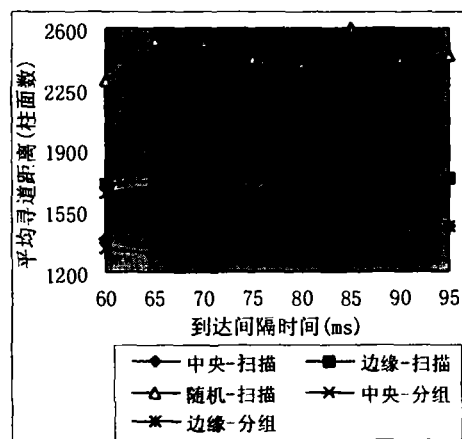


图8 到达间隔时间对平均寻道距离的影响

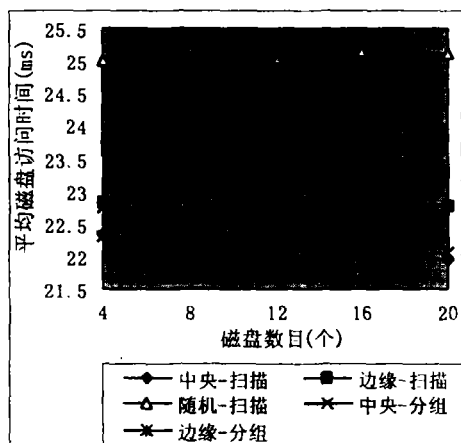


图6 磁盘数目对平均磁盘访问时间的影响

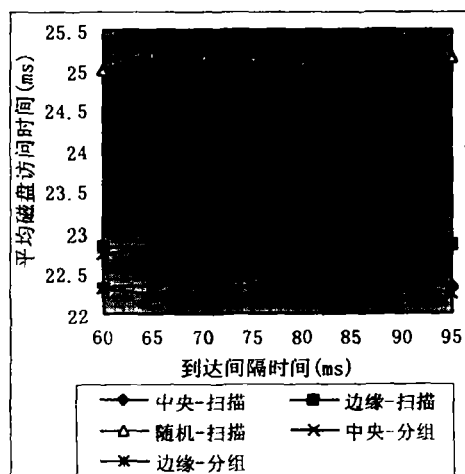


图9 到达间隔时间对磁盘访问时间的影响

可看出,CL 和 EL 相对于 RL 策略更能有效提高磁盘的各项指标,而分组调度算法与扫描算法的性能相差不大。

5.2 到达间隔时间的影响

图 7~9 是磁盘各指标在到达间隔时间影响下得到的仿真结果。磁盘的数目为 4。

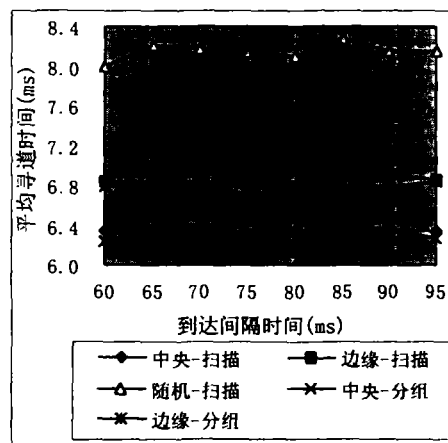


图7 到达间隔时间对平均寻道时间的影响

可看出,在到达间隔时间变化的情况下,CL 和 EL 比 RL 策略更能有效提高磁盘的各项指标。

结论 从仿真结果可知,本文中提出的重组策略(CL、EL)能有效地提高磁盘的性能,其中 CL 的效果更好。

当采用 SCAN 算法且磁盘数目是 4 时,CL 相对于 RL 能降低平均寻道距离近 60%;当磁盘数目是 20 时,平均寻道距离降低近 65%;而当磁盘数从 4 变为 20 时,平均磁盘访问时间分别下降 11.9% 和 13.7%。

从仿真结果可看出,GS 算法和 SCAN 算法对磁盘各指标的影响相差不大,然而从第 3 部分的分析中可知,GS 能明显降低请求响应延迟。

综上所述,CL 策略和 GS 算法能更好地提高各项指标。

参考文献

- 1 Vogel A, et al. Distributed multimedia applications and quality of service-A survey. IEEE Multimedia Journal, 1995, 3 (8): 87~95
- 2 Haskin R L. The Shark Continuous-Media File Server. In: Proc. of IEEE 1993 Spring COMPCON, San Fransisco, CA, Feb. 1993. 12~17
- 3 Chervenak A L, et al. Choosing the Best Storage Systems for Video Service. In: Proc. of ACM Conf. on Multimedia, 1995. 109~119
- 4 Flynn R, et al. Disk Striping and Block Replication Algorithms for Video File Servers. In: Proc. of Intl. Conf. on MCS, 1996. 590~597
- 5 Dan A, et al. Buffering and Caching in Large-Scale Video servers. In Digest of Papers. IEEE Intl. Computer Conf. 1995
- 6 Chervenak A L. Tertiary Storage: An Evaluation of New Applications. [PhD thesis]. U. C. Berkeley, Dec. 1994
- 7 Ganger G, et al. The DiskSim Simulation Environment. Version 2, Reference Manual, CMU, Dec. 1999
- 8 刘衡竹, 胡衡江, 陈福接. VOD 服务器设计的支撑技术. 计算机工程与科学, 1999, 21(3)