分布式多媒体存储系统中的全局缓存管理

朱晴波,乔 浩,陈道蓄
(南京大学计算机科学与技术系,江苏南京 210093)

摘 要: 多媒体存储系统必须同时支持连续媒体和非连续媒体的访问.由于连续媒体的实时要求,系统必须为访问连续媒体保留大量的磁盘带宽,并且持续很长的时间,这使其他类型文件的访问性能严重下降.本文根据连续媒体的访问特性,提出了一个分布式多媒体存储系统的协同缓存策略 GLNU,充分利用系统中其他结点上可用的内存资源,提高缓存的利用率,以减少连续媒体的磁盘 I/O,从而提高其他媒体的访问性能.仿真试验表明 GLNU 在各种不同的参数下,均优于现有的缓存策略,是一种适合分布式多媒体存储系统的缓存策略.

关键词: 分布式; 多媒体存储系统; 协同缓存
中图分类号: TP316.6 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12-1832-04

Global Memory Management in Distributed Multimedia Storage Systems

ZHU Qing-bo, QIAO Hao, CHEN Dao-xu
(*Department of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu* 210093, *China*)

Abstract: Multimedia storage systems need to handle continuous media data and conventional types of data. Continuous media data access requires high I/O rates and tends to last for a relatively long period of time. However this can drasticaly degrade the performance of accesses to other data types. A new cooperative caching policy called GLNU suitable for the access to continuous media is proposed in the distributed multimedia storage systems. Simulation results indicate our new policy outperforms other cache management policies and greatly improves the cache hit rate and thus reduces the disk I/O for continuous media data.

Key words: distributed system; multimedia storage system; cooperative caching

1 引言

多媒体存储系统需要同时支持连续媒体(如视频、音频)和非连续媒体(如文本、图象).连续媒体(缩写为 CM)的访问需要消耗大量的磁盘带宽,并且持续很长的时间.长时间为连续媒体访问保留大量的磁盘带宽,严重影响了访问其他文件类型的性能.如何减少连续媒体的磁盘 I/O,就成为多媒体存储系统的一个关键问题.缓存技术是一种简单有效的解决方法.

单个结点的内存是有限的.与集中式的存储系统相比,分布式存储系统在可扩展性和性能价格比上有很大的优势[1].为了最大限度地提高多媒体存储系统的性能,可通过网络来充分利用系统中其他结点的空闲内存资源.为此,就有必要引入协同缓存技术.ATM 等高速网络的迅猛发展,使通过网络共享数据的代价远小于从磁盘读取数据[2].

2 相关工作和分析

在 *xFS* 协同缓存系统中[3],如果本地结点的缓存没有命中,就询问远地结点有没有所需的数据块.本地结点根据 LRU 策略被置换出去的数据块,如果是当前唯一的副本,就随机选择一个远地结点缓存该数据块,否则简单丢弃掉.这种算法称为 N-Chance[2].在 GMS 分布式共享内存系统[4]中,系统维护各结点数据块的全局时间戳信息,以近似实现一个全局 LRU 的置换算法.GMS 的研究表明使用全局信息比随机选择能更大的提高性能.Sarkar 在文献[5]中采用了与 GMS 不同的完全分布式的方法来管理全局信息,同时尽可能的避免采取分布式方式带来的时间信息的不准确度.以上三种方法都采用传统的缓存策略 LRU,以利用应用程序在时间和空间上的访问局部性进行数据置换,在多媒体存储系统中,由于连续媒体的时序特点,这种策略并不适用[6-8].

许多论文研究了适合连续媒体的置换算法,最具有代表性的两个策略就是 Interval Caching[6,7]和 Distance Caching[8],这些算法主要利用连续媒体顺序访问的特性,提高缓存的命中率,从而减少对磁盘的访问.但是,它们没有考虑如何在分布式多媒体存储系统中,利用系统中其他结点的内存空间,以进一步提高缓存的命中率.

3 GLNU 协同缓存策略

一个结点内存中的数据分为两部分,一部分是本地用户

收稿日期:2001-09-20;修回日期:2002-08-08
基金项目:863 高科技计划(No.2001AA113050)

访问的数据块,称为本地块;远地结点用户暂存在该结点的数据,称为全局块.一个结点内存中本地块和全局块的相对多少是根据我们的算法动态变化的.简单的说,一个活动结点的内存逐渐充满本地块,而一个空闲结点的内存逐渐充满全局块.

我们的策略称为 GLNU 算法(Global Longest-Not-to-be-Used Alogrithm).在系统的每个结点上都采用相同的策略,在给出策略描述之前,首先给出主拷贝的定义[8]:当一个数据块从磁盘中读到内存,这个数据块就被称为主拷贝,这可以通过在数据块头设置一个标志位表示.每个主拷贝数据块的位置由一个全局目录来记录和维护.其次给出宿主结点的定义:假设文件以通常的方式分散存储在各个结点的磁盘上,每个结点都有一张文件和结点的映射表,磁盘上存放文件 f 的结点称为文件 f 的宿主结点.

3.1 置换算法

下面给出适合连续媒体的置换算法.最优的替换算法应该产生最高的命中率,从而使磁盘 I/O 最少,显然如果一个置换算法能将一个将来最长时间内不会被访问的数据块替换出去,它就是最优的.最优算法需要知道未来的访问信息,这在传统的应用程序中是不可能实现的.LRU 利用传统应用程序空间和时间上的局部性,认为在过去最长时间内没有被访问的数据块,就是将来最长时间内不会被访问的数据块,应该被替换出去.但是根据文献[8],假设所有用户访问同一个影片,LRU 产生的不命中率高到 88%到 99%,因此 LRU 算法不能满足连续媒体服务的要求.考虑连续媒体的访问特性,在不考虑 VCR 功能的情况下,如果现有的客户维持播放速率不变,一个数据块在将来什么时候被访问是可以近似计算出来的,在这里假设影片以 MPEG-1 进行压缩编码,播放速率为 1.5Mbps.对于内存中不会被现有客户访问的数据块,简单的对下一个客户的到达时间进行预测.假设客户的达到满足参数为 λ 的泊松分布,下一个客户的到达时间为当前到达客户的到达时间加上 $1/\lambda$.之所以做这个预测是因为要尽可能的将 CM 文件开头的数据块保留在缓存中,以提高命中率和客户的响应时间.根据这个近似计算的结果将数据块排序,被认为在将来最长时间内不会被访问的数据块(Longest-Not- to-be-Used Block)被替换出去.显然,如果不考虑 VCR 功能,且对未来客户到达时间的预测是准确的话,这个算法就是最优的.

3.2 GLNU 策略

(1)局部缓存层策略

当在结点 n 的客户需要读取数据块 B 时,如果结点 n 的本地块中有 B,则局部命中.如果在结点 n 的全局块中有 B,则局部命中,但需要将其变为本地块.如果本地内存中没有 B,即局部不命中,则向全局目录查询该块的主拷贝 B_{mc} 的位置,这个过程称为查询过程(Block Lookup).

(2)协同缓存层策略

如果 B_{mc} 在系统其他结点的内存中且是全局块,则将其直接拿到结点 n 中变为本地块,这时候结点 n 必须置换出去一个数据块.如果结点 n 有全局块,就任选一个将其送到 B_{mc} 所在结点;否则选择本地块中的主拷贝 LNU 块送到 B_{mc} 所在的结点,成为全局块.如果 B_{mc} 在系统其他结点的内存中且是局部块,这时候分两种情况考虑:情况一:整个系统中都是主拷贝数据,则直接将其拿到结点 n 中,如果结点 n 有全局块,就任选一个将其送到 B_{mc} 所在结点;否则选择本地块中的主拷贝 LNU 块送到 B_{mc} 所在的结点,成为全局块.情况二:整个系统中存在有非主拷贝数据,则产生一个新的拷贝,将其送到结点 n 成为本地块,这个新拷贝称为非主拷贝.此时结点 n 必须置换出去一个数据块,我们的置换策略分三个层次进行:第一,如果结点 n 的本地块中有非主拷贝数据,则从中选择 LNU 块简单的丢弃.第二,当结点 n 的本地块都是主拷贝数据,内存中有全局块时,则选择整个系统的非主拷贝 LNU 块丢弃,从结点 n 任选一个全局块送到其所在的结点.第三,当结点 n 的本地块都是主拷贝数据,内存中没有全局块时,则选择整个系统的非主拷贝 LNU 块丢弃,从结点 n 选择本地块中的主拷贝 LNU 块送到其所在的结点,成为全局块.

(3)磁盘层策略

如果在其他结点的内存中也不存在,则全局不命中,必须查找文件映射表,到宿主结点的磁盘去读该数据块到结点 n 的内存中,并设为主拷贝.此时结点 n 必须置换出一个数据块,置换策略同样分三个层次进行:第一,如果结点 n 的本地块中有非主拷贝数据,从中选择 LNU 块简单的丢弃.第二,当结点 n 的本地块都是主拷贝数据,内存中有全局块时,选择整个系统的非主拷贝 LNU 块丢弃,从结点 n 任选一个全局块送到其所在的结点.如果整个系统中都是主拷贝数据块,选择整个系统的主拷贝 LNU 块丢弃掉,然后从结点 n 任选一个全局块送到其所在的结点,依然是全局块.第三,当结点 n 的本地块都是主拷贝数据,内存中没有全局块,选择整个系统的非主拷贝 LNU 块丢弃,从结点 n 选择本地块中的主拷贝 LNU 块送到其所在的结点,成为全局块.如果整个系统中都是主拷贝数据块,选择整个系统的主拷贝 LNU 块丢弃,从结点 n 选择本地块的主拷贝 LNU 块送到其所在的结点,成为全局块.

3.3 简单的讨论

从以上可以看出,全局块都是主拷贝.算法的思想是相当直观的,一个活动结点的内存逐渐充满私有块,并开始使用系统中的其他结点的内存,而一个空闲结点的内存将逐渐充满全局块.通过首先丢弃非主拷贝数据的方法,尽可能把主拷贝保存在内存中来提高全局命中率,因为主拷贝的丢弃意味着下一次再访问这块时必须从磁盘读取.当系统内存中只有主拷贝数据时,选择全局 LNU 块替换出去,以上两点是针对提高全局命中率提出的,这是我们的首要目标 - 减少磁盘访问.在将结点的主拷贝替换到其他结点时,首先选择全局块,因为这些是远地结点的数据,然后再选择本地 LNU 块,这些是为了提高局部命中率,以减少内部网络带宽的消耗.全局目录可以维护在一个管理结点上,也可以象文献[5]一样由各个结点轮流维护.

4 仿真试验和性能分析

将用模拟程序测试在不同参数情况下四种算法(LRU,LNU,GLRU,GLNU)的优劣,以及讨论各个参数对算法的影响.GLRU 和 GLNU 是全局协同缓存策略,LRU 和 LNU 是指无协同

的缓存策略,也就是当本地内存不命中时,直接向该文件的宿主结点请求读磁盘.每次仿真试验持续 150 分钟,在试验开始之前缓存的内容为空.评价算法优劣的最重要的指标就是缓存的命中率,连续媒体文件访问的缓存命中率越高,其他文件的访问的性能就越高.同时,我们还测试平均磁盘带宽和平均内部网络带宽的消耗. 有关测试中所用的参数见表 1.

表 1 基本测试参数

平均达到时间间隔(s)	T_{exp}
Zipf 参数	θ
CM 文件数	M
CM 文件长度(Min)	Len
结点个数	N
空闲结点百分比	P
每个结点内存(MB)	Mem
数据块大小(KB)	B

假设客户到达满足泊松分布,客户对 CM 文件的访问选择符合参数为 0.271 的 Zip$f$ 分布[6,7].CM 文件以 MPEG-1 标准压缩,播放速率为 1.5Mbps,以通常方式分散存放在各个结点上.我们在下面的测试中变化其中一些参数的值,并讨论它们的影响.

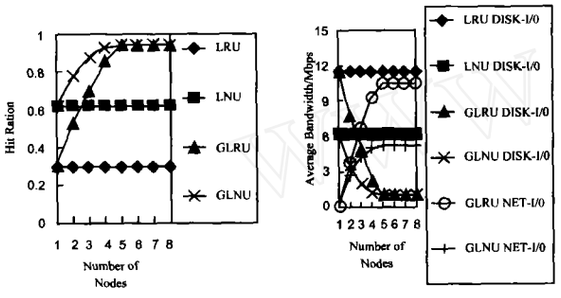


图 1 结点数的影响

(T_{exp} = 400s, Len = 100min, Mem = 256M, B = 64KB, M = 1)

图 1 说明增加系统结点的个数对各个算法的影响.假设用户要访问的媒体集为一个 CM 文件,由图中可以看到在用户要访问的媒体集大小确定时,随着结点数目的增加,GLRU 和 GLNU 能够利用的内存资源增加,缓存的命中率也随着增加,LRU 和 LNU 因为无法利用其他结点的内存,缓存命中率保持不变.GLNU 在用户要访问的媒体集大于各结点可利用的内存之和时,命中率明显优于其他三种算法.当结点数目增加至 5 个以上,访问媒体集小于可利用内存,也就是要访问的数据都在缓存里,此时 GLNU 和 GLRU 缓存命中率达到最高.相应的,随着缓存命中率的增加,GLNU 和 GLRU 所需要的磁盘带宽下降,直到达到最低点. 同时 GLNU 和 GLRU 所消耗的内部网络带宽增加,从图中可以看出,GLNU 消耗的平均磁盘带宽和内部网络带宽的总和约等于 LNU 所需的磁盘带宽,同样 GLRU 消耗的平均磁盘带宽和内部网络带宽的总和约等于 LRU 所需的磁盘带宽,这表明全局协同缓存的策略是用内部网络带宽来代替磁盘带宽,以减少磁盘的 I/O,GLNU 所消耗的磁盘带宽始终少于其他三种算法.

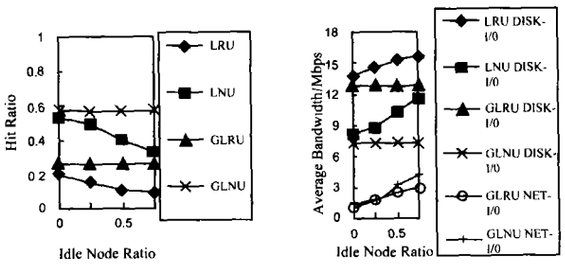


图 2 空闲结点百分比的影响

(T_{exp} = 400s, Len = 100min, Mem = 256M, B = 64KB, M = 4, N = 4)

图 2 说明了空闲结点的百分比对各个算法的影响.从图中我们可以看到在用户要访问的媒体集的大小一定,系统结点个数一定时,活动结点和空闲结点相对比例变化的影响.当系统中每个结点都是活动结点时,也就是空闲结点的百分比为零时,GLNU 的命中率只比 LNU 高 7.5%,这是因为 GLNU 使用了全局信息的缘故.由于 GLNU 和 GLRU 可利用的内存资源总和不变,它们的命中率也不变.然而 LNU 和 LRU 的命中率随着活动结点的减少而降低,因为它们无法利用空闲结点的内存资源.相应的,LRU 和 LNU 所消耗的磁盘带宽也随之增加,而 GLNU 和 GLRU 消耗的磁盘带宽保持不变,但网络带宽随之增加.

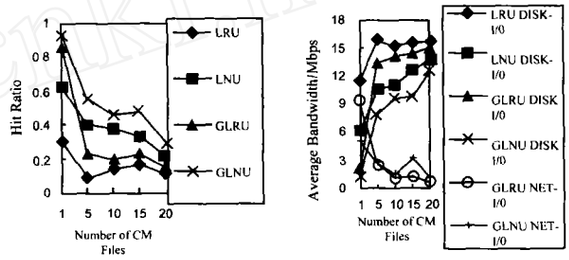


图 3 CM 文件数的影响

(T_{exp} = 400s, Len = 100min, Mem = 256M, B = 64KB, N = 4, P = 50%)

图 3 说明 CM 文件数对算法的影响.从图中我们可以看到系统结点个数和空闲结点百分比一定时,四种算法的命中率都随 CM 文件数,也就是访问媒体集的增大而减少.GLNU 的命中率始终优于其他三种算法.值得注意的是,GLRU 的命中率随着访问媒体集的增加而急剧减少,比无协同的 LNU 算法更差,表明在有连续媒体的环境下,GLRU 算法已经不再适用了.同时,当 CM 文件数增加到一定程度比如 20 个以上,访问媒体集与可用内存空间的比率约为 24:1,此时 GLNU 对命中率的提高效果就不是很明显了.相应的,算法所需要的磁盘带宽随着缓存命中率的降低而增加.GLNU 所需要的磁盘带宽是四种算法中最少的.

总的来说,在各种参数变化的情况下,GLNU 都优于其他三种策略,尤其当可用的空闲内存很大,访问媒体集相对较小时,对性能的提高很大.从试验数据分析可以得到结论,GLNU 是一种适合分布式多媒体存储系统的缓存策略.

5 结束语

在多媒体存储系统中,减少连续媒体对磁盘的访问,能够

极大的提高其他媒体的访问性能.因此设计一个好的缓存管理策略是至关重要的.我们提出了一个基于连续媒体访问特性的协同缓存策略 GLNU,通过充分利用分布式存储系统中的内存资源,提高了缓存的利用率,用网络带宽来代替磁盘带宽的消耗,减少了连续媒体的磁盘 I/O.通过仿真试验可以得出结论,在各种参数变化的情况下,GLNU 均优于其他缓存策略.我们的进一步工作是在考虑 VCR 功能的情况下,改进协同缓存策略 GLNU 并研究其性能,以及其他应用的运行对协同缓存系统的影响.

参考文献:

[1] S A Barnett, et al. A cost comparison of distributed and centralized approaches to video-on-demand [J]. IEEE J. Selected Areas in Communications, 1996, 14:1173 - 1183.

[2] M Dahlin, et al. Cooperative Caching: Using Remote Client Memory to Improve File System Performance [C]. In Proc. of the First Symp. on Operating Systems Design and Implementation, 1994. 267 - 280.

[3] T Anderson, et al. Serverless network file systems [J]. ACM Trans. on Computer Systems, 1996, 14(1):41 - 79.

[4] M J Feeley, et al. Implementing Global Memory Management in a Workstation Cluster [C]. In Proc. 15-th Symposium on Operating Systems Principles, 1995. 201 - 212.

[5] P Sarkar, et al. Efficient Cooperative Caching Using Hints [C]. In Proc. of the 2nd Symp. on Operating Systems Design and Implementation, 1996.

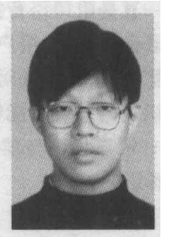
[6] A Dan, et al. Buffer management policy for an on-demand video server [R]. IBM Research Report RC 19347.

[7] A Dan, et al. Buffering and Caching in Large-Scale Video Servers [C]. In Proc. Compcon, 1995. 217 - 224.

[8] Ozden, et al. A buffer replacement algorithms for multimedia storage systems [A]. In Proc. of the Third IEEE Intl. Conf. on Multimedia Computing and Systems [C]. 1996. 172 - 180.

[9] F M Cuenca-Acuna, et al. Cooperative Caching Middleware for Cluster-Based Servers [C]. In Proc. of the 10th Intl. Symp. on High Performance Distributed Computing, 2001.

作者简介:



朱晴波　男,1976 年生于江苏无锡,1999 年毕业于南京大学,现为南京大学计算机系硕士研究生,主要研究方向为分布式计算.

乔　浩　男,1978 年生于江苏江宁,1999 年毕业于南京大学,现为南京大学计算机系硕士研究生,主要研究方向为分布式计算.

陈道蓄　男,1947 年生,南京大学计算机系系主任,教授,博士生导师,主要研究领域为分布与并行计算机系统.