

多级存储结构视频点播系统中的接纳控制算法研究^{*}

Study of Access Control Algorithm in Hierarchical Storage VOD Systems

荣波 孙景鳌 蔡安妮

(北京邮电大学 北京100876)

Abstract As the core of VOD system, video servers need a large storage capacity. Therefore, a hierarchical storage structure is inevitable. As a result, it's an urgent problem to find a suitable access control algorithm for VOD system based on hierarchical storage structure. In my thesis, an unfair access control algorithm based cost factor is discussed. The computer simulation of the performance of this algorithm is shown, and satisfied results are achieved.

Keywords VOD systems, Access control

0 引言

视频服务器作为视频点播(VOD)系统的核心,需要存储和管理大量经过压缩的视频节目。一方面,为了向用户提供实时、并发的视频流,需要使用高速的磁盘阵列来存放节目,但价格非常昂贵。另一方面,不同的节目受观众的喜爱程度不同,将点播频率极小的节目存放于价格昂贵的磁盘阵列中是不经济的。因此,有必要使用分级存储方案^[1,2],即将不常被点播的节目存放到磁带库或光盘库等相对廉价的大容量存储设备中去,作为非在线节目,通常这部分存储称为二级库。

对于多级存储结构视频点播系统来说,为了使其稳定、高效工作,一个好的接纳控制算法是非常必要的^[3]。以前对视频点播系统的研究多集中在解决单服务器磁盘IO受限问题^[4,5],而对于多级存储问题涉及较少。本文从资源共享的角度出发,为多级存储结构视频点播系统建立了模型,并提出了一种适于它的接纳控制算法。

1 视频点播系统中的多级存储结构

图1是一个分布式VOD系统的结构图。其中,二级库设备往往由独立的、称为文档服务器(Archive Server,简称AS)的计算机来管理。由于磁带库和光盘库的速度相对较慢,并且存取时延大,为了保证播放的质量,非在线节目需要先调入到指定的磁盘中去才能播放,这个过程称为数据迁移。存放迁移节目的这部分磁盘称为迁移缓冲区。相对于二级库,视频服务器(Video Server,简称VS)管理的磁盘阵列中的节目称为实时库(一级库)。它包括经常被点播的在线节目和迁移缓冲区中的节目。

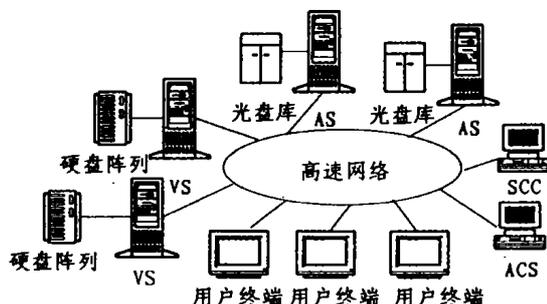


图1 一个典型的分布式VOD系统结构

在图1中,系统控制计算机(System Control Computer,简称SCC)负责整个系统的管理和配置等工作;AS负责二级存储设备(磁带库或光盘库)的管理,并实现与VS间的视频文件的迁移;VS负责磁盘阵列的管理、视频文件的I/O及视频流的网络传输等,以响应用户的请求,输出保证质量的视频流;接纳控制服务器(Access Control Server,简称ACS)负责整个视频点播系统的接纳控制。

用户点播终端的操作主要有两种:节目查询和播放。这两种请求都将发送到ACS进行处理。对于节目查询请求,ACS执行查询并将结果返回给点播终端。对于点播请求,如果节目处于某个VS中,ACS则指示终端与相应VS建立连接;如果节目不在VS中,则指挥相应的AS执行数据的迁移调度,将节目调入某个VS中去,供终端点播。

2 多级存储结构视频点播系统中接纳控制的基本过程

当一个点播请求到达分布式VOD系统时,接纳控制服务器首先查看该请求点播的节目是否在视频服务器的硬盘上。如果在,则根据当时视频服务器和网络资源的占用情况来决定是否接纳该请求。如果节目在二级库中,接纳控制服务器将进一步检查是否有足够的资源用于数据迁移。如果有,则再检查一下视频服务器和网络资源占用情况来决定是否接纳该请求。上述过程可以由图2表示。

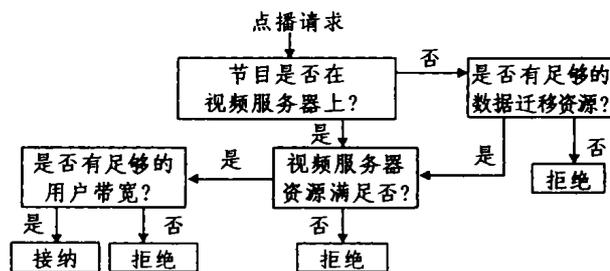


图2 视频点播系统中接纳控制的基本过程

接纳控制算法将决定在系统当前状态下,新到达的对某种类别节目的请求是否被接受。简单而又不失一般性,我们可以将节目分为两类。一类存储在视频服务器的硬盘上,另一类

^{*}国家自然科学基金资助项目(69972009)。荣波 博士,主要研究方向是多媒体信息网及其网络管理。孙景鳌 教授,博导,从事电视、多媒体通信和成像技术方面的教学和研究工作。蔡安妮 教授,博导,从事多媒体通信、图像处理和识别方面的教学和研究工作。

存储在二级库中。第一类节目通常是较为流行的,第二类节目则是一些不常见的点播率较低的节目。假设用向量 (n_1, n_2) 来表示用户请求数,其中 n_1 代表视频服务器硬盘中节目请求的数目, n_2 代表二级库中节目请求的数目,VOD系统的状态则可以由一个容量受限的状态空间表示,即 $n_1 + n_2 \leq C$, C 表示视频服务器能够同时处理的最大点播数。系统的状态空间还可以进一步受其它条件的限制。例如,设计者希望接纳控制算法更倾向于接纳第一类节目,从而可以人为添加约束条件。

一般来说,从资源共享的角度出发,可以将接纳控制算法分为3类:各类别完全无关的算法、各类别完全共享的算法和加约束条件的共享算法。

(1)各类别完全无关的算法:不同类别(例如,一级库和二级库节目)的点播请求各自占用不同的资源,它们之间完全互相独立。

(2)各类别完全共享的算法:不区分点播请求的类别,只要系统资源存在,就接纳该请求。

(3)加约束条件的共享算法:各种类别的点播请求共享系统资源,但是根据不同的类别,点播请求的接纳与否服从一定约束条件的限制。

在实际的视频点播系统中,从有效地利用资源来看,加约束条件的共享算法效果最好。

3 基于代价因子的不公平接纳控制算法

3.1 基于代价因子的不公平接纳控制算法的初步分析

本文提出的基于代价因子的不公平接纳控制算法属于加约束条件的共享算法。由于我们的系统采用多级存储结构,因此对于两类不同存储位置的节目,接纳控制算法需要区别对待。基于代价因子的不公平接纳控制算法的基本思想是分配给不同类别的节目请求以不同的代价因子,在接纳控制时根据类别区别对待。例如在两级存储结构的系统中,对于第一类节目(存储在视频服务器硬盘上的节目)分配给它代价因子1,对于第二类节目(存储在二级库中的节目)分配给它代价因子 b 。这样一来,VOD系统的状态空间将受到两种条件的限制:一是系统容量 C 的限制;二是不同的代价因子对不同类别的点播的限制;即:

$$n_1 + bn_2 \leq C \quad \text{且} \quad b > 1 \quad (1)$$

由上式可以看出,对于第一类节目只要资源充足,接纳控制算法就会接纳该请求。而对于第二类节目,由于 $b > 1$,因此即使资源充足接纳控制算法也不一定会接纳该请求。由此可以得出结论,这种接纳控制算法具有倾向性。在实际系统中这种倾向性是必要的,因为考虑到资源占用的多少和收费等问题,系统并不想将资源公平地分配给各类节目。

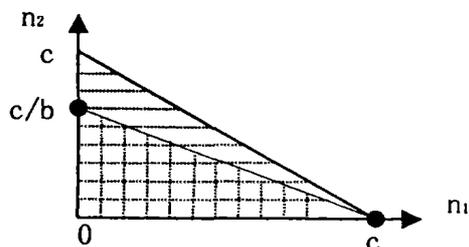


图3 基于代价因子的不公平接纳控制算法的系统状态空间图

基于代价因子的不公平接纳控制算法的系统状态空间如图3双重阴影部分所示。只要在状态空间图中找出阻塞状态

点,就可以计算出该接纳控制算法的阻塞率。阻塞状态点定义为:当系统处于该状态点时,再有新的点播请求到达,接纳控制算法将拒绝该请求。

3.2 基于代价因子的不公平接纳控制算法的分析模型

首先,我们将系统简化为单一视频服务器通过高速网络连接一组文档服务器。对于一个能够将用户点播负载良好地在多个视频服务器之间进行平衡的VOD系统,这样的简化是合理的。另外,为了分析方便,我们假设迁移缓冲区足够大,且由文档服务器至视频服务器的网络带宽足够宽,因而在迁移过程中不会出现任何阻塞。同时不考虑迁移缓冲区的管理算法问题,认为缓冲区中的二级库节目点播完后即被删除。如前所述节目共分为两类,假设两类点播请求的到达均为泊松过程,且系统服务时间服从负指数分布。其中,第二类节目的服务时间与第一类节目不同,第一类节目的服务时间为视频服务器的服务时间(节目长度),而第二类节目的服务时间为视频服务器的服务时间加上迁移时间。

在上述假设条件下,整个系统状态的变化可以看作是一个状态空间为 Ω 的生灭过程,而且我们所研究的基于代价因子的不公平接纳控制算法符合凸条件。即除了阻塞状态点之外,在其它状态点上无论是新请求到达还是原有请求离开都不会阻塞。由以上分析我们可以定义系统模型参数如下:

$$\Omega = \{(n_1, n_2) : n_1 + bn_2 \leq C, b > 1\};$$

Ω_1 —第一类节目的阻塞状态点集合;

Ω_2 —第二类节目的阻塞状态点集合;

λ_1 —第一类节目的点播请求到达率;

λ_2 —第二类节目的点播请求到达率;

$1/\mu_1$ —第一类节目的平均服务时间;

$1/\mu_2$ —第二类节目的平均服务时间;

$\rho_1 = \lambda_1/\mu_1; \rho_2 = \lambda_2/\mu_2$;

C —服务器所能处理的最大点播请求数;

b —分配给第二类节目的代价因子;

P_{b1} —第一类节目的点播阻塞率;

P_{b2} —第二类节目的点播阻塞率;

$P(n_1, n_2)$ —状态点 (n_1, n_2) 的概率,且有如下关系:

$$P_{b1} = \sum_{(n_1, n_2) \in \Omega_1} P(n_1, n_2) \quad (2)$$

$$P_{b2} = \sum_{(n_1, n_2) \in \Omega_2} P(n_1, n_2) \quad (3)$$

3.3 基于代价因子的不公平接纳控制算法的计算机仿真分析

根据上述数学模型,我们采用Kaufman等人提出的方法^[6]计算公式(2)、(3),对基于代价因子的不公平接纳控制算法进行了计算机仿真分析。第1组仿真结果如图4所示,它表明了当 ρ_1, ρ_2 保持不变而代价因子 b 变化时,点播阻塞率 P_{b1}, P_{b2} 的变化趋势。其中参数设定如下: $C = 300, \rho_1 = 320, \rho_2 = 90, c/b; 0-300$ 。

由图可知,当 c/b 的值较小时,第一类节目的点播阻塞率较小,第二类节目由于受接纳控制算法中代价因子的限制,其点播阻塞率较大,这时视频点播系统的资源大部分被第一类节目所占用。随着 c/b 值变大,第二类节目受代价因子的限制越来越小,其点播阻塞率也逐渐变小。而由于第二类节目在资源上的竞争,第一类节目的点播阻塞率逐渐变大。此时从整个系统资源占用的角度来看,第二类节目的比重逐渐增加。当 c/b 的值逐渐接近于 c 时,仿真结果显示接纳控制算法的性能逐渐接近于完全共享策略,即第一类节目的点播阻塞率和第二类节目的点播阻塞率逐渐趋向于同一值。

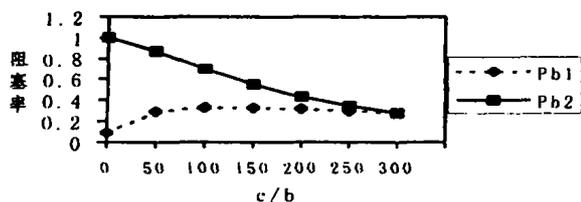


图4 代价因子变化时阻塞率的变化情况

第2组仿真结果表明了当 ρ_1 和代价因子 b 保持不变而 ρ_2 变化时,点播阻塞率 Pb_1, Pb_2 的变化趋势,如图5所示。其中参数设定如下: $C=300, \rho_1=270, \rho_2: 0-350, c/b=80$ 。由图可以看出,当 c/b 保持恒定时,随着第二类点播请求到达率的增加,第二类点播请求的阻塞率迅速增加,而第一类点播请求的阻塞率增长幅度要远小于第二类点播请求。由上可知,在这种情况下基于代价因子的不公平接纳控制算法对第一类点播请求起到了很好的保护作用。

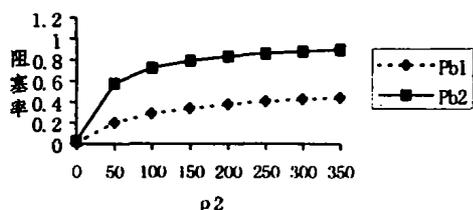


图5 ρ_2 变化时阻塞率的变化情况

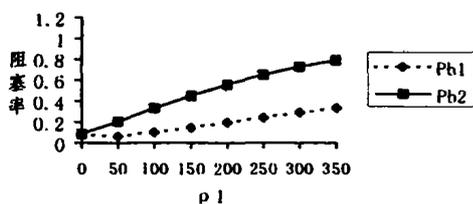


图6 ρ_1 变化时阻塞率的变化情况

第3组仿真结果表明了当 ρ_2 和代价因子 b 保持不变而 ρ_1 变化时,点播阻塞率 Pb_1, Pb_2 的变化趋势,如图6所示。其中参数设定如下: $C=300, \rho_1: 0-350, \rho_2=100, c/b=80$ 。

由图可以看出,当 ρ_1 较小时,第一类点播请求的阻塞率基本保持不变,而第二类点播请求的阻塞率迅速增大。当 ρ_1 大到一定程度时,系统容量的限制开始起作用,第一类点播请求和第二类点播请求的阻塞率都迅速增大。事实上在这种情况下,代价因子对第一类点播请求的接纳限制不大,系统容量才是主要的限制条件。

结束语 通过对上述仿真结果的分析,我们可以看出基于代价因子的不公平接纳控制算法通过对代价因子的调节实现了对不同类别节目的不公平接纳。上述算法对于提高基于多级存储的分布式 VOD 系统的整体服务指标和服务效益有很大帮助,因而具有较大的实用价值。

参考文献

- Burbeck D W, Rowe L A. Hierarchical storage management in a distributed void system. *IEEE Multimedia*, Fall 1996. 37~47
- Siu-Wah L, John C S L. Scheduling and data layout policies for a near-line multimedia storage architecture. *Multimedia System*, 1997, 5:310~323
- Vin H, et al. An observation-based admission control algorithm for multimedia servers. In: *Proc. of the 1st IEEE Intl. Conf. on Multimedia Computing and Systems*, May 1994. 234~243
- Reddy A L N, Wyllie J. Disk Scheduling in a Multimedia I/O System. *ACM Multimedia*, 1993. 225~233
- Diagle S J, Strosnider J K. Disk Scheduling for Multimedia Data Streams. *ACM Multimedia Computing and Networking*, 1996, 2188:212~223
- Kaufman J S. Blocking in a shared resource environment. *IEEE Transactions on Communications*, 1981, COM-29 (10): 1474~1481

(上接第142页)

用软的连接处理,可能会带来中断响应延迟过大的问题,但至少,在系统的结构图上,可以清晰地看到这一点。

结论 本文的最初动机是为了回答一个问题:为什么操作系统的驱动程序比一般的应用程序在结构上要复杂。在尝试各种分析方法后,最终,从软件体系结构的角度来对其进行剖析,才引发了对连接间实现结构的细致分析。在本文中提出如下观点:

1) 连接件在结构上,应分为两部分:连接结构和协议。其中,连接结构是静态的;协议是与应用相关的,动态的。并建议在描述连接件的整体结构时,将其两者分开描述。

2) 连接件的实现结构限定为6种。控制连接(4种)为:同步硬连接、同步软连接、异步硬连接、异步软连接。数据连接(2种)为:软数据连接和硬数据连接。其中,软连接和硬连接这种差别对整个系统的结构存在着产生重大影响的可能。但某些形式化的标记方法很难直观地区分它们。

3) 提出一种新的评价软件体系结构复杂性的方法。

参考文献

- Shaw M, Garlan D. *Software Architecture: Perspectives on an Emerging Discipline*. Englewood Cliffs, NJ: Prentice-Hall, Inc. 1996

- 周之英. 现代软件工程(中). 北京:科学出版社, 2000
- Allen R J. A Formal Approach to Software Architecture. *CMU-CS-97-144*, 1997
- Garlan D, Monroe R, Wile D. ACME: an architectural interconnection language. [Technical Report CMU-CS-95-219]. 1995
- Shaw M. Comparing Architectural Design Styles. *IEEE Software*, 1995, 12:11
- Gamma E, et al. *Design Patterns: Elements fo Resuable Object-Oriented Software*. Addison Wesley Longman, Inc. 1995
- 黎伟建. 领域特定的软件体系结构. *中山大学学报(自然科学版)*, 1998. 6
- 张家晨,等. 基于主动连接件的软件体系结构及其描述方法. *软件学报*, 2000. 11
- Hoare C A R. *Communicating Sequential Process*. Englewood Cliffs, NJ: Prentice-Hall, Inc. 1985
- Ducasse S, Richner T. Executable Connectors, towards Reusable design elements. In: Jazageri M, Schauer H, eds. *Proc. of the sixth European Software Engineering Conf. (ESEC'97) Berlin*: Springer-verlog, 1997
- Allen R, Garlan D. A formal basis for architectural connection. *ACM Trans. on Software Engineering and Methodology*, 1997, 6 (3)