

虚拟光盘库数据流服务器的设计与实现

易法令¹, 万继光²

(1. 长江大学 计算机科学学院, 湖北 荆州 434023;

2. 华中科技大学 计算机学院 外存储系统国家专业实验室, 湖北 武汉 430074)

摘要:文中通过分析虚拟光盘库的数据流服务器的功能,把数据流服务器划分为主控、管理、访问控制及数据 Cache 四大模块,具体阐述了每个模块的作用。在分析数据流服务过程基础上,重点研究了数据流服务系统设计与实现中的几个关键问题:用户访问数据流服务器的权限、流量、计费的方面的控制问题;在用户访问过程中超时处理技术;为提高数据流服务器的数据传输效率而采用的数据 Cache 技术。实验结果表明:虚拟光盘库具有较高的数据传输速率。

关键词:虚拟光盘库;Cache;连接控制;数据流服务器

中图分类号:TP333.4

文献标识码:A

文章编号:1673-629X(2006)07-0017-03

Design and Implementation on Data Stream Server of Virtual CDrom Library

YI Fa-ling¹, WAN Ji-guang²

(1. College of Computer Science, Yangtze University, Jingzhou 434023, China;

2. National Laboratory of Storage Systems, Computer School, Huazhong Univ. of Sci. and Techn., Wuhan 430074, China)

Abstract: Analyzes the functions of data stream server of virtual CDrom library, which are partitioned into main-control sub-module, management sub-modules, access-control sub-modules and cache sub-modules. On the process of data stream server, the following key technologies are analyzed and designed detailedly: the problems which users access data stream server on permission, flux and fare; the timeout processing technology; the data cache technology for improving data transfer rate. The test proves that virtual cdrom library has higher data transfer rate.

Key words: virtual CDrom library; cache; link control; data stream server

0 引言

“虚拟光盘库”是在光盘镜像服务器的基础上发展起来的附网存储设备(Network Attached Storage, NAS)^[1,2]。光盘镜像服务器现在已成为光盘在网络上共享的重要设备,它通过将数据光盘镜像在硬盘上,使用户能以硬盘的速度来共享光盘上的信息资源。但由于光盘镜像服务器在对各种类型光盘的适应性不足及对实时访问处理的不连续性问题^[3],推出了一种升级产品,即“虚拟光盘库”。它能按数据光盘自身的格式存储管理各种光盘资料;具有光盘自动镜像、数据冗余、加密光盘镜像、用户响应快、同时支持的用户数多及能够实现流量控制和光盘访问计费等特点。

虚拟光盘库体系结构如图1所示,从图中可以看出:虚拟光盘库包含两部分,即:客户端和服务端。客户端与服务器端通过自定义的网络协议进行数据通信,在客

端设有专用的访问服务器的管理客户端程序。由于虚拟光盘库是根据光盘数据的特性定义的网络传输协议,并还采用了数据 Cache 等多种技术,所以用户在访问镜像光盘时,看到的是如同直接访问光驱一样平滑的数据流。服务器端主要由光盘数据流服务器和管理服务器构成。文中主要研究服务器端的相关技术。

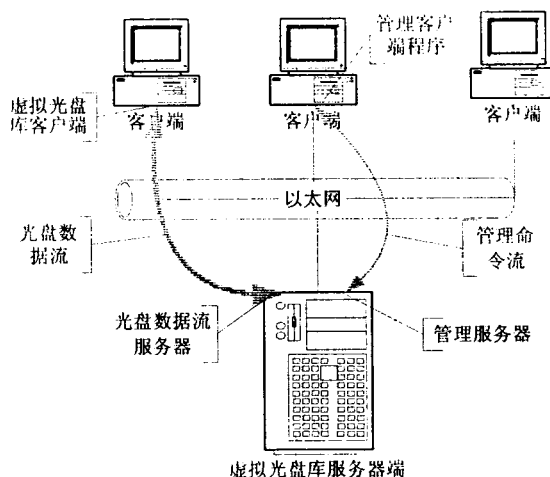


图1 虚拟光盘库的体系结构

收稿日期:2006-03-02

基金项目:湖北省教育厅重点科研基金项目(2002A04007)

作者简介:易法令(1969-),男,湖北荆州人,博士,副教授,研究方向为网络存储、并行处理、计算机网络。

1 光盘数据流服务器总体分析

根据服务器端的功能可以划分为两大部分:一是数据存储部分,主要包括磁盘的管理、光盘镜像及用户的管理等等,该功能与图 1 中的管理服务器相对应;一是进行数据流的服务,主要实现与客户端的连接及相关的管理控制功能,该功能与图 1 中的光盘数据流服务器相关。后者是文中研究的重点。

1.1 数据流服务器模块划分

由于虚拟光盘库的服务器是针对特定应用的专用服务器,所以它的数据流服务与一般的网络服务器有所不同,其数据流服务应该具有以下功能:

(1)启动整个服务器端,并通过自定义的网络协议接受客户端的访问,实现与客户端的连接。

(2)对用户的访问连接进行控制,并记录用户的访问情况,实现计费等相关管理功能。

(3)为了提高用户连接访问的速度,在服务器端一般要设置一个大的数据 Cache,这样客户端就可以直接从数据 Cache 中读取数据,进而提高了访问效率。

根据上述功能要求,在设计数据流服务器时一共分为 4 个子模块:光盘流主控子模块、访问控制子模块、管理子模块及 Cache 模块,各模块之间的关系如图 2 所示。

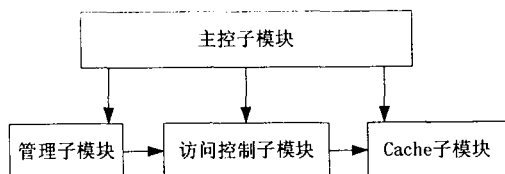


图 2 光盘数据流服务器的功能模块划分图

其中:主控子模块是对其它模块进行控制并启动自定义服务器的工作线程;访问控制子模块主要实现用户的连接控制、访问记录与计费服务等相关功能;管理子模块实现用户的连接及相关的控制管理功能;Cache 子模块实现服务器全局 Cache 功能。这些模块相互配合实现光盘库的数据流服务功能。

1.2 数据流服务过程分析

数据流服务器启动后,通过客户端的访问开始整个服务过程:首先是客户端通过以太网访问服务器端,并建立连接。在建立连接的过程中,服务器对用户的合法性及相关的连接参数进行控制。连接建立以后,服务器端要对数据传输过程进行管理,比如流量控制、用户数量的控制、连接时间的控制等等;如果连接超时,还必须进行相关处理。根据上述过程,数据流服务器采用的是一种多线程的处理方式。光盘流服务器模块一共有 3 个固定线程及一类用户的动态工作线程,用户工作线程是在每个用户连接时动态创建。各线程的功能如下:

(1)光盘流服务器主线程:响应用户连接,创建连接处理工作线程。

(2)连接超时检测线程:处理超时连接。

(3)动态管理线程:处理服务器管理的任务。

(4)连接处理工作线程:动态线程,用户的每个连接对应一个线程。用户的数据请求就是通过动态创建的线程来实现的。

2 数据流服务的实现

在数据流服务启动后,客户端可以通过适当的认证方式访问服务器端的光盘资源。在数据访问的过程中,重点需要考虑以下问题:连接的建立与控制问题;超时处理问题;数据 Cache 的相关结构与算法。

2.1 连接控制与超时处理

在整个系统对用户的访问有着各方面的控制,所有用户的信息及存储光盘的配置信息等都以文件的形式保存。服务器启动后,在系统中就建立了一个全局的用户信息表,当前控制所需的所有信息都在用户信息表中。当用户登录成功后,将用户在配置文件中的相关信息调入到对应的用户信息表,从而实时控制该用户的所有连接。用户信息表包括两级结构,一是用户名及历史信息;一是与用户名对应的当前连接信息。其结构如图 3 所示。在图 3 中有 3 种数据结构:用户名及历史信息结构(USER-INFO-TABLE)、用户当前的连接信息(USER-CURRENT-LINK-INFO)及在一定时间段内连接请求信息(LINK-REQ)。

从图 3 中可以看出:一个用户与一个用户信息表的节点相对应,一个用户可以建立多个连接,所有的连接以链表的方式保存在用户信息表中。在用户信息表中包含大量的控制信息,这些信息部分保存在配置文件 user.conf 中,部分是在用户访问的过程中产生的。当用户访问服务器,通过用户名和相关的密码验证后,开始检查配置文件中用户的相关数据。整个过程如下:

(1)查用户的帐户余额、IP 地址等是否符合要求。如果是,则转(2);否则,退出。

(2)检查用户是否已在用户信息表中。如果已存在,则转(3);否则,把用户信息加入到信息表中。

(3)检查用户的连接数是否已满。如果未满,则为用户建立一个连接;否则,退出。

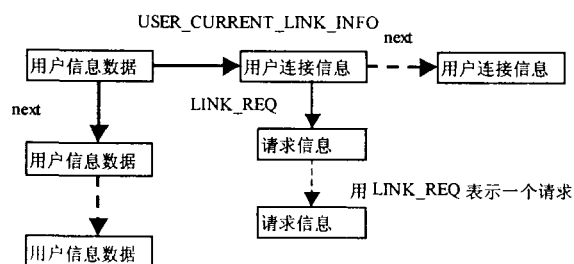
建立用户连接以后,用户就可以发送数据请求。对用户的数据请求,还需要进行权限检查,即:检查用户是否具有访问对应光盘数据的权限。如果能够访问,则开始向数据 Cache 发送数据请求,在数据读取的过程中,根据需要可以对单个连接进行流量控制。流量控制主要通过当前连接信息结构 USER-CURRENT-LINK-INFO 来实现。在该结构中 pstRegionReq 是一个最近请求链表的头指针,当最近请求数达到一定数量时,就计算对应的流量,如果流量超过规定的限额,则延时处理数据请求,进而达到流量控制的目的。

当一个连接在规定的时间内无数据请求时,则进行超时处理。在设计中,超时处理通过一个专用的线程进行管理。定时扫描用户信息表中的每一个连接,如果在规定的

时间间隔内,有未发出数据请求的连接,即“超时”,则中断该连接,并保存相关的数据。

用 USER_INFO_TABLE 表
示连接上的用户信息

用 USER_CURRENT_LINK_INFO
表示一个连接



(a) 用户信息表结构图

```
typedef struct tagUserInfoTable
{
    char * pcUserName;
    int iLevel; //用户的级别
    int iMaxConnectNum; //最大连接数
    int iMaxSingleConnect; //单个连接的最大流量
    int iTotatTime; //总的连接时间
    int iTotatFlux; //总的流量
    int iCurrentTotatTime; //当前总的连接时间
    int iCurrentTotatFlux; //当前总的流量
    int iCurrentConnectNum; //当前的连接数
    int iSpendMoney; //已用金额
    int iSpareMoney; //剩余金额
    USER_CURRENT_LINK_INFO * pstUser-
    Link;
    struct tagUserInfoTable * next;
} USER_INFO_TABLE;

typedef struct tagUserCurrentLinkInfo
{
    int iConnectID; //连接的 ID
    LINK_REQ * pstRegionReq; //
    当前给定时间段的请求信息
    struct tagUserCurrentLinkInfo *
    next;
} USER_CURRENT_LINK_INFO;

typedef struct tagLinkReq
{
    time_t stCurrentTime; //当前的
    连接时间
    int iCurrentFlux; //当前的流量
    struct tagLinkReq * next;
} LINK_REQ;
```

(b) 与用户信息表相关的数据库结构

图 3 用户信息表的整体结构图及对应的数据库结构

2.2 数据 Cache

由于读取数据光盘,一般的都是连续的大块数据,所以在实际设计中,采用了客户端、服务器端两级 Cache 结构^[4]。在服务器端 Cache 的数据组织采用控制结构与数据块分离的方法,控制结构由 CACHE_CDID 表示,整个 Cache 采用平衡二叉树来组织,整体结构如图 4 所示。服务器端的 Cache 一般都很大会,所以它与磁盘之间采用的是全相联映射方式,读取的数据可以根据需要放在 Cache 中任何位置,没有映射规则限制。

与数据 Cache 相关的基本操作和算法主要有:查找、替换和预取^[5]。

①查找:根据 Cache 系统的结构,采用的是二级查找方式,即:首先按线性的方式查找对应的数据光盘,然后针对每个光盘采用平衡二叉树搜索算法。相比其他查询算法,平衡二叉搜索树具有较高的查询效率。

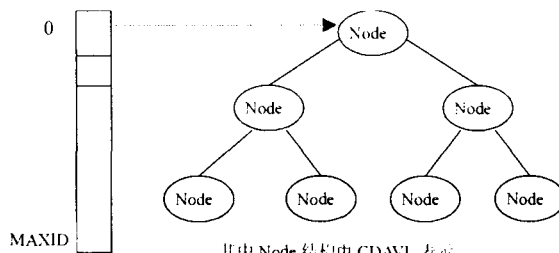
②替换:采取 LRU 替换算法,首先根据 LRU List 决定需替换的块;然后在对应块的位置用新数据替换原来的老数据;再根据逻辑块地址 (LBA) 值重新调整平衡二叉树,使其符合平衡二叉树的定义;最后,更新 LRU List 信息。

③预取:采用根据历史记录进行光盘数据预取的策略,根据用户最近访问光盘的请求的扇区偏移量来确定请

求之间的连续性,进而确定预取系数,通过预取系数来计算预取的长度。

以光盘的 ID 号为单位组织数据,
以 CACHEID 结构表示每个节点

每个光盘有应平衡二叉
搜索树



其中 Node 结构由 CD AVL 表示

(a) 光盘数据在 Cache 中的组织结构

```
typedef struct tagCacheCDID
{
    struct avl - tree * pstStAvlRoot; //查询树的根指针
    int iFrameType; //光盘的帧类型
    CACHE_CDID
}

typedef struct tagCDAvl
{
    struct avl stAvl //平衡二叉树节点的结构
    long iLba //请求的 LBA 值
    char * pcCacheBuffer //节点对应数据
    块指针
    iCD - AVL
```

(b) 与 Cache 组织结构对应的数据库结构

图 4 光盘数据在 Cache 中的数据库结构

3 结果测试

在 100M 的局域网内,用 Iometer 对服务器的性能进行了测试,在服务器端和客户端都设置了 Cache。根据光盘数据读取的一般特点,主要测试了顺序读的性能,同时还将其读性能与 Windows2000 提供的网上邻居顺序读性能进行了对比。图 5 是性能对比图。从图中可以看出,虚拟光盘库在数据传输方面具有较好的性能。

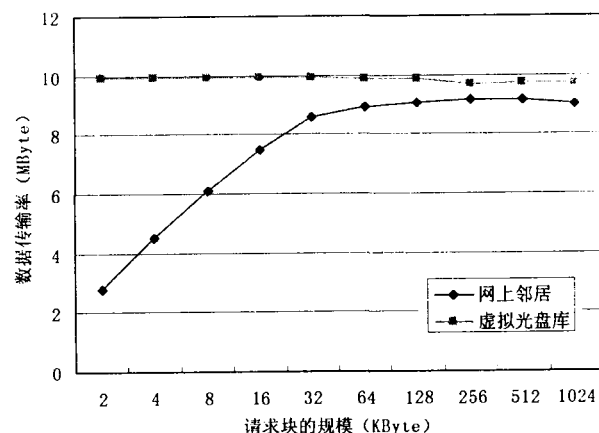


图 5 虚拟光盘库与网上邻居的顺序读性能对比图

4 结论

虚拟光盘库为共享各种类型的光盘资源提供了较好的平台,其服务器端采用了存储与服务分开处理技术,使服务器端能更好地进行模块化的设计;对用户访问连接的各种处理方式增强了虚拟光盘库作为存储服务产品的适应性。同时,在服务器/客户端建立两级 Cache 的技术大大提高了数据的传输速率。

(下转第 134 页)

数据的读入及结果保存。

3.4 信息隐藏处理算法软件构件的实现

对于图 4 所示的算法类,考虑到不同用户的需求,其软件构件采用动态连接库(DLL)的形式,具体的建立方法如下:

(1) 用 MFC DLL 模式建立工程项目:FArithmetic

(2) 在工程 FArithmetic 中加入如图 4 所示的类定义 CFArithmetic.h 和类实现 CFArithmetic.cpp;

(3) 建立一个对外接口文件 INCAlgorithm.cpp,实例化算法类的指针对象 Infp,并对每一个具体的实现建立如下所示的对外接口:

```
extern "C" void __declspec(dllexport) shinefp_lowfilter_smallblock(int carrier_width, int carrier_height BYTE ** pCarrier)
{ Infp -> shinefp_lowfilter_smallblock(carrier_width, carrier_height, pCarrier); }
```

(4) 对工程进行编译,如果正确,就会生成一个有用的文件:CFArithmetic.dll;

(5) 在算法研究交互系统 FingerProcess 中,将文件 CFArithmetic.dll 复制到本项目工程 FingerProcess 的目录下(也可以复制到其他目录下),通过标准的接口,就可以调用动态连接库中的算法实现。

由图 2 和图 3 的软件构件组成的类结构关系如图 4 所示。

4 结束语

笔者利用 VC++ 编译平台,用信息隐藏前后处理框架和信息隐藏处理算法软件构件建立了一个信息加密算法的研究平台。该平台可以对图像、视频、语音等载体信息进行信息加密的算法研究,具有以下优点:

1) 具有较好的独立性和可扩充性。主要是面向研究人员,对算法前后处理的软件构件,开发人员可以加入许多新的载体文件,以适应新的需求。

2) 图 3 所示的信息加密的算法类,由于不必要公开其

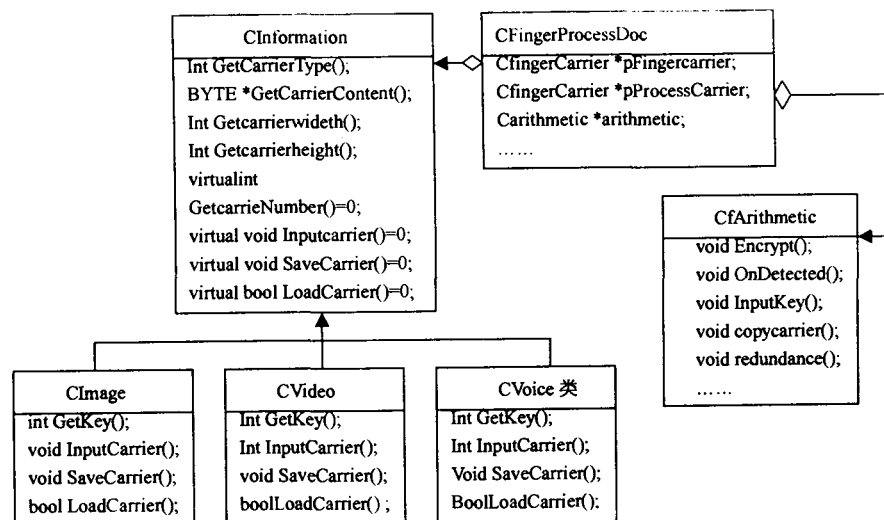


图 4 信息隐藏前处理类与算法类的关系

实现的细节,为用户提供了一个对外的公共接口,因此,具有二次开发的能力,用户可以调用算法类中的操作,用于其它信息隐藏处理的操作。

3) 信息隐藏算法的改变不会影响到已经建好的系统,同时,研究人员可在系统中加入自己的算法,这个软件构件为研究算法的人员提供了一个可对比的样板,对评价算法的优劣提供了一个参考,使研究具有可延续性。

参考文献:

- [1] 贾宏宇. 基于构件+构架复用技术的工控软件设计研究[J]. 计算机工程与应用, 2002(1): 30-33.
- [2] 李丽娟, 何克清. 基于 stragetry 模式的图像处理软件构件[J]. 计算机应用, 2005(5): 1099-1101.
- [3] 赵方圆, 魏志强, 林勇. 基于构件的可重构软件开发平台技术研究[J]. 计算机工程与设计, 2005(1): 33-37.
- [4] 王晓咏, 杨明福. 基于 .NET 平台的构件开发若干问题研究[J]. 计算机应用与软件, 2005(2): 27-31.
- [5] 钮心忻. 信息隐藏与数字水印的研究与发展[J]. 计算机教育, 2005(1): 22-24.
- [6] Martin R C. 敏捷软件开发[M]. 邓辉译. 北京: 清华大学出版社, 2005.
- [7] Richard E G, Johnson H R, Vissides J. 设计模式: 可复用面向对象软件的基础[M]. 李英军, 马晓星, 蔡敏, 刘建中译. 北京: 机械工业出版社, 2000.

(上接第 19 页)

参考文献:

- [1] 谢长生, 刘文藻, 万继光. 一种基于 Linux 的附网光盘镜像服务器的研究与实现[J]. 计算机工程与科学, 2004, 26(5): 1-3.
- [2] 秦保社. 网络存储技术的发展[J]. 微机发展, 2002, 12(1): 74-77.
- [3] 汤慕娜, 裴京. 虚拟镜像光盘网络服务器的设计与实现[J]. 计算机工程, 2003, 29(4): 140-142.
- [4] 易法令, 万继光, 谭志虎, 等. 光盘镜像服务器的 Cache 技术研究与实现[J]. 计算机工程与科学, 2004, 26(10): 91-94.
- [5] 易法令, 谢长生, 万继光. 一种新的基于 RAID 的 CACHE 技术研究与实现[J]. 小型微型计算机系统, 2004, 25(12): 2173-2178.