

一种分布式多媒体信息点播系统模型*

刘积仁 张应辉 王兴伟 江 早

(东北大学软件中心 沈阳 110006)

摘要 分布式多媒体信息点播系统 DMIODS(distributed multimedia information-on-demand systems)作为对 VoD(video-on-demand)系统的一种扩展,已经成为多媒体信息处理技术中的一个新的应用领域.本文从分析 DMIODS 的几个特点出发,提出了一种 DMIODS 的抽象功能层次模型,依据这个抽象模型,给出了一个原型系统,并对原型系统作了实现.

关键词 分布式多媒体信息点播,功能模型,原型系统,媒体对象服务器,负载平衡.

中图法分类号 TP393

网络通信技术、计算机技术尤其是多媒体技术的快速发展和硬件成本的大幅度降低以及集成化程度的提高,使得视频点播 VoD(video-on-demand)成为可能.^[1,2]VoD 为人们对视频信息的获得提供了新的方法和途径,已被应用于点播电影、交互式视频游戏等娱乐性行业.但是在信息高速发展的时代,人们对信息的需求不仅限于对视频信息的获取.例如,对视频信息检索,可能并不需要信息的全分辨率再现,这就需要信息的可伸缩性编码,但 VoD 对此并没有很好的支持.再如,对于信息流的重组,现有的 VoD 系统也显得力不从心.所以,媒体信息表示和用户对信息再现方式要求的多样性、信息服务范围的广泛性,决定了视频点播不能够完全满足人们对信息点播的需要.因此,为了满足人们对各种各样信息的需要,基于 VoD 的基本概念、方法和系统设计中的一些技术,我们开展了对分布式多媒体信息点播系统 DMIODS(distributed multimedia information-on-demand systems)的研究. DMIODS 是一个基于分布环境的、能够针对用户的需要、为用户提供各种各样的多媒体信息的系统.它具有如下几个特点:

- (1) 信息交互方式的非对称性:从信息源到用户的正向通信量大,而用户对信息源的反馈信息量小;
- (2) 具有互操作性和可集成性:这是从用户角度对分布式系统的要求;
- (3) 支持连续媒体的传输:DMIODS 所支持的数据形式大多是带时间关系的连续媒体信息;
- (4) 支持多媒体信息流的同步和重组:保证静态信息流的空间同步和动态信息流的时间同步;
- (5) 支持具有可伸缩性的信息编码方式:编码后的视频流可以原来的分辨率/速度再现信息,但是这种视频流中信息子集的不同组合可以不同的分辨率/速率再现视频信息;
- (6) 支持动态 QoS 管理:因为信息是以可伸缩方式编码的,所以系统对资源进行重新权衡时,可以根据需求动态地调整用户的 QoS 参数;
- (7) 信息服务器的组成方式应该具有可伸缩的体系结构:根据资源的使用情况,采用适当的负载平衡策略,把信息存取和网络传输的负载均衡到相应的机群中.同时,系统具有信息存储和网络带宽的可扩展性;
- (8) 系统具有透明性:信息的重组在一定条件下是可见的,但是对用户是透明的;
- (9) 系统具有可移植性:具有可运行特征的媒体信息在不同的运行环境下都可以再现.

一个 DMIODS 系统应该由如下几种物理设备组成:①具有可伸缩性体系结构的信息服务器组和大容量的存储介质;②具有能够保证媒体同步传输和实时性需求的具有可扩展性的网络体系结构;③具有一定多媒体信息处理能力的用户终端设备.

作为对 VoD 技术的扩展,DMIODS 同样存在着许多需要解决的关键问题^[3],如支持 DMIODS 中多媒体信息的网

* 本文研究得到“九五”国家攻关项目基金资助.作者刘积仁,1955年生,博士,教授,博士生导师,主要研究领域为分布式多媒体信息处理技术,CSCW,组件技术.张应辉,1972年生,博士生,主要研究领域为分布式多媒体信息处理技术,组件技术.王兴伟,1968年生,讲师,在职博士生,主要研究领域为分布式多媒体信息处理技术,CSCW.江早,1965年生,博士后,讲师,主要研究领域为多媒体图象处理.

本文通讯联系人:刘积仁,沈阳 110006,东北大学软件中心 308 信箱

本文 1997-01-16 收到原稿,1997-05-08 收到修改稿

络传输,适合于 DMIOs 的多媒体数据编码技术,即怎样实现可伸缩的数据编码方式,多媒体数据库技术以及媒体信息如何在存储介质上的合理存放等.多媒体信息处理技术和 VoD 设计中的许多方法和经验为这些问题的解决奠定了很好的基础,许多研究机构在这方面也进行了系统的研究和探讨.例如,英国 Lancaster 大学的多媒体研究小组在视频服务器的结构和负载均衡机制等方面作了深入的研究;美国 Texas 大学的分布式多媒体实验室在大规模存储介质和 QoS 等方面进行了一定的研究.另外,IBM,SGI,ORACLE 等世界著名公司也有了相应的产品,如 ORACLE 的数据库系统已经开始支持对视频信息的访问等等.但是,对 DMIOs 而言,这些研究和开发工作有一定的局限性,不能很好地满足 DMIOs 的上述特点.另外,DMIOs 是涉及不同应用部门、与多种类型的信息服务领域有关的一种综合系统,必须有一个共同遵守的参考模型去指导 DMIOs 的设计与实现.本文第 1 节提出了一种 DMIOs 层次功能模型,第 2 节给出了这个模型的一个原型系统,并对原型系统进行了实现.

1 DMIOs 层次功能模型

DMIOs 是一种非对称的信息传输系统.设计中,我们采用了 Client/Server 模式,对 Client 和 Server 方中各层结构和功能有了一个统一的定义,把两边抽象成一个对等的功能层次模型,更好地体现了开放系统中各层之间的对应关系以及低一层向高一层提供的一致的、透明的服务.这里的一致性是指请求服务和得到服务的一致性.在层的定义中,使用了基于对象的思想,顺应自然地每层的结构和功能进行了抽象.DMIOs 的功能模型如图 1 所示.

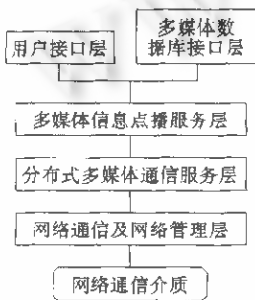


图1 DMIOs的功能模型

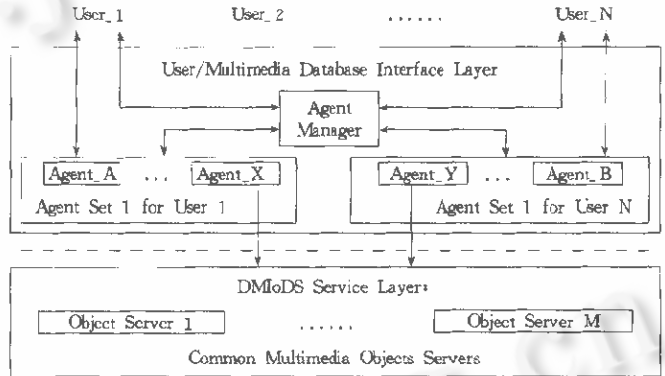


图2 Agent访问方法

1.1 用户接口和多媒体数据库接口层

用户接口层位于 Client 方,它向用户提供一种透明的服务,用户和服务请求都是通过它所分派的代理集合 (Agent Set)来完成,如图 2 所示.一旦系统接收到新的合法的用户任务,即产生相应的代理对用户进行服务,完成任务后,则取消相应的代理集合.Agent 可以把用户的不同服务请求解释成系统能够识别的基本元组信号,并把这些基本元组信号传递给信息点播服务层,基本元组信号的描述如下:

```

<TupleSignal>→<SignalName> <Set of Parameter>
<Set of Parameter>→
  <InforNameID> // 0 表示初始信息请求
  <InforNameDescription> // 信息的简短描述
  <ServiceTYPE> // 服务类型
  <Set of QoS Parameter> // QoS 配置情况
  <UserInfor> // 用户的身份证明信息
  
```

信息名描述项中数据类型和地址信息对用户来说是透明的,用户所能看到的信息名只是一个关键字、图标或其它简短的描述信息.用户接口层的主要功能如下:

- (1) 接收并识别合法的用户命令,把用户命令解释成系统能够识别的基本元组信号;
- (2) 为用户分派 Agent 集合,组织有效的 Agent 集合去完成基本元组信号命令;
- (3) Agent 对象可以激活信息服务层中对应的 Object Server 去完成具体操作;
- (4) 优化 Agent 资源利用;

(5) 向用户提供信息资源访问的 API 接口。

多媒体数据库接口层位于 Server 方,作为信息的提供者,除了能完成传统信息的查询之外,还能够支持带时间属性媒体连续流的并行访问。多媒体数据库接口层和用户接口层一样,也是通过 Agent 集合向下层提供信息的,多媒体数据库接口层的主要功能是:

- (1) 接收信息请求命令,把命令解释成数据库能够识别的查询指令;
- (2) 组织有效的 Agent 集合去完成信息的存取;
- (3) 提供多进程/多线程的并发访问;
- (4) 采用优化策略,使单个存储介质上多媒体信息流的存放更加合理。

系统产生的代理集合都是由代理总管(Agent Manager)来管理和协调的,采用代理总管可以提高系统的自适应能力,符合分布式人工智能(DAI)的要求。通过代理总管,系统能有效地组织代理集合去完成用户的请求任务,一旦代理集合确定后,用户或数据库就只与代理集合中的代理进行联络。处理过程中,代理根据用户的请求或数据库的使用情况周期性地向总管进行汇报,如果需要,可向代理总管申请特权,增加单个 Agent 的适应能力;长时间空闲的代理,总管可以把它解除,或把它转交到别的代理集合中去;代理集合中所有的代理都不能胜任的工作可向总管申请新的类型的代理去完成。这样,系统就可以不断地、周期性地根据变化的任务、资源、外部信息以及系统的历史行为,改变系统中 Agent 的类型与数目,实现组织重构。麻省理工学院的 Hewitt C, Alpha G 等人提出的 Actor 模型^[4]很适用于用户管理单元中的 Agent 管理。

1.2 多媒体信息点播服务层

多媒体信息点播服务层是模型中媒体信息处理的核心部分,它隐藏了信息处理的所有细节,向不同功能类型的用户代理或数据库查询代理提供透明的服务。多媒体信息点播服务层的主要功能是:

- (1) 提供公用多媒体对象服务器(Common Multimedia Object Server)功能;
- (2) 接收用户代理的服务请求命令,激活相应的服务对象,并寻找对应的信息资源;
- (3) 为用户代理传递初始的信息服务内容;
- (4) 根据服务的类型和 QoS 要求,以适当的形式向用户再现信息;
- (5) 支持动态 QoS 管理,协商调节用户 QoS 要求;
- (6) 向多媒体数据库接口层传递信息查询命令,提供多媒体数据的过滤功能;
- (7) 平衡信息资源和网络带宽的负载,最大限度地提高整个系统的信息吞吐量;
- (8) 统计视频信息的点播率,在点播的峰值时期,自动完成信息的复制和迁移;
- (9) 支持可伸缩性的信息存储体系结构,当有存储介质加入或减少时,系统可以根据信息的使用情况,对资源进行平衡,最大限度地保证系统的整体吞吐量;
- (10) 管理和监视整个系统中公用多媒体对象服务器资源的登记和使用情况。

在功能上,多媒体信息点播服务模块实现了多媒体信息处理的透明性。不论是何种媒体信息,对用户代理而言,只有服务请求和得到服务两种功能,至于不同类型的信息怎样调用不同的程序去处理等功能已经隐藏在模块内部。适合多媒体信息本身的特点,处理大数据量媒体流的多媒体对象服务程序最好在本机运行。用户/数据库代理在激活相应的媒体对象服务器时,可根据对象的登记情况,首先在本地服务器中查找。

1.3 分布式多媒体通信服务层

传统网络通信的模式和方法是在特定的发展背景和条件下,根据当时的应用需求提出来的,经过多年的完善和改进,现在已经相当成熟。但是随着硬件水平的不断提高,应用范围的不断扩大,传统的网络通信方式已经在很多方面,尤其是支持多媒体信息通信方面表现出了它的不合理性。多媒体信息的数据类型,不同于传统的数据类型,它不仅包含了无时间属性的大容量静态数据类型(如大文本、图形、图像以及程序等),而且主要的数据类型是与时间有关的连续媒体数据类型(如音频、视频、三维 VR 数据类型等)。与这些数据类型相对应的是许多传统网络所很难具备的各种信息服务类型,如视频点播、电视会议、多点播送(Multicast)等。当然,还存在各种单媒体数据间的同步问题,如音频和视频信息的“对口型”问题。

为了有效地支持多媒体信息的通信,有两种有效的途径:一种是设计全新的适合于多媒体信息,特别是连续媒体信息传输的低层网络协议;另一种是基于目前的网络协议,在传输层之上增加多媒体通信服务层。依据实际的使用情况和硬件环境,我们采用了第 2 种方案。功能层次模型中的分布式多媒体通信服务模块是为适应第 2 种途径而设计的,其主要功能是实现媒体信息传送的透明性。它的主要功能有:

(1) 对于无时间属性的数据信息,可在为满足 QoS 而规定的时间内将信息向上或向下传送,保证信息的空间同步;

(2) 对于与时间相关联的媒体信息,根据 QoS 要求,尽量减小延迟值和延时抖动现象保证媒体信息的时间同步.

1.4 网络通信及网络管理层

此模块主要是依赖于现有网络通信机制,向上提供透明的网络传输.它的屏蔽所有具体网络的特点,保证数据的可靠传输,主要解决分布的透明性、信息迁移的透明性以及数据复制的透明性等问题.

DMIoDS 的功能模型是一个抽象的层次模型,每个层次可以由 1 个或几个对象组成,对象的属性和方法定义后,它的部分具体实现可以由对象服务器(Object Server)完成.图 2 中,分配给 User-1 的 Agent-X 对象中有一个方法定义,其实现部分可以是虚拟的,它的具体实现在信息点播服务层 Object Server 1 中.当它调用某个方法时,就能激活 Object Server 1 中对应的方法去实现用户代理的服务请求,向用户代理提供透明的服务.图 2 中分配给 User-N 的 Agent Y 和 Agent X 一样,可以激活 Object Server 1 为它服务.对用户而言,它只需要知道服务的名称,而不需要知道服务的具体细节,这些细节包括服务名称所对应的函数名及参数传递的语义特征等等,在这些名称和函数之间有一张动态的对应表,参与组件化的对象间通过协商可以修改对应表.另外,对于 Object Server 中的具体实现,Client 和 Server 方的重点是不一样的.在 Client 方,多媒体信息点播服务模块是最重要也是最复杂的部分,而在 Server 方,支持对多媒体数据库的并行访问则变成了主要矛盾.下面是我们基于 DMIoDS 的功能层次模型,在 Ethernet 和 FDDI 环境下实现的该模型的原型系统.

2 DMIoDS 模型的原型系统实现

2.1 系统组成

为了适应异构环境的分布系统特征,我们选用了 10Mbps 的交换式 Ethernet 网(用 IBM 8271 Switch 作 HUB)和 FDDI Token Ring 作为网络连接,如图 3 所示.

(1) SUN SPARC 20 兼作视频服务器和路由器,其上运行 Solaris 5.5.1 操作系统;

(2) IBM RS6000 作为主要的视频信息服务器,其上运行 AIX 4.1 操作系统;

(3) SGI 兼作信息服务器和高质量的视频客户终端,其上运行 IRIX 6.3;

(4) 用 PC(Pentium133)机作为客户端和对象服务器,其上运行 Windows NT 4.0 或 Windows 95,支持 JAVA RMI.^[5]

2.2 系统功能描述

RS6000, SUN APARC 20 和 SGI 一起作为信息服务器,通过底层的 TCP/IP 协议,接收并识别合法的用户,向用户提供初始的信息服务目录,接受用户的服务请求命令,搜索到所需的信息,并在用户和信息资源之间建立一条逻辑上的通道.它还可以登记资源的使用情况,协调用户的 QoS 要求.另外,通过负载均衡策略,当访问某个服务器的用户增多时,可以把用户对信息的访问平稳地均衡到其它的服务器上.

在 WindowsNT/95 上通过 JAVA RMI 可以实现对远程对象的访问.

用 RS6000 作网络管理服务器,通过 SNMP,网络管理员可以监视整个网络资源的配置情况和使用情况、网络通信的流量控制情况、网络的安全性能等等.

2.3 负载均衡策略

我们采用遗传算法^[6]来实现负载均衡策略.给定一个原子服务时间 τ 和服务周期 ΔT ,每个服务周期内的负载情况可以用如下两个正整数向量表示.

$$\vec{E} = \{E_1, \dots, E_i, \dots, E_n\}, \quad \vec{S} = \{S_1, \dots, S_i, \dots, S_n\}$$

表明在一个服务周期内,第 i 个服务请求在第 S_i 个服务介质上得到服务,并且得到服务的时间为 E_i , n 为某个服务周期内信息请求数目.所以,第 j 个服务介质上的负载量为

$$P_j = \sum_{i=1}^n E_i \zeta_i, \quad \text{其中 } \zeta_i = \begin{cases} 1 & j=S_i \\ 0 & j \neq S_i \end{cases}$$

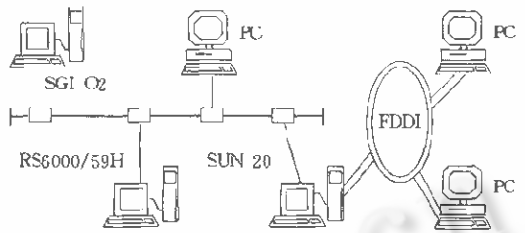


图3 DMIoDS的原型系统实现

为了保证信息流的 QoS, 必须满足:

$$\sum_{i=1}^n E_i \leq \frac{m \Delta T}{r} \tag{1}$$

设第 j 个服务介质平衡量为 $B_j = \frac{P_j}{P_j^0}$, 其中 P_j^0 表示第 j 个服务介质的最大负载量, 则 $E(\vec{B}) = \frac{1}{m} \sum_{j=1}^m B_j$, $D(\vec{B}) = \frac{1}{m} \sum_{j=1}^m (B_j - E(\vec{B}))^2$, m 为一个区域内的服务介质数目.

为了让尽量让每个服务请求在同一个服务介质上得到服务, 设惩罚函数为 $T = \frac{k}{n} \sum_{i=1}^n T_i$, 其中 k 为一个常量, $T_i = \begin{cases} 0 & S_i = S'_i \\ 1 & S_i \neq S'_i \end{cases}$, 表示第 i 个服务请求的惩罚函数, 其中 S'_i 表示第 i 个服务请求在上一个周期内得到服务的服务介质编号. 最终的估价函数为

$$Eval(\vec{E}, \vec{S}) = D(\vec{B}) + T \tag{2}$$

在满足式(1)的条件下, 系统可以接收新的服务请求. 每隔一个 ΔT , 服务请求可能变化, 通过对 \vec{E} 和 \vec{S} 每代之间的复制、交叉和变异, 使式(2)逐渐趋向近优, 使各服务介质的负载趋于平衡.

在用户方, 使用环状缓冲机制, 为了保证回放的连续性, 缓冲池大小应为 $S_{buffer} = V_c \frac{2\Delta T}{r}$, V_c 为客户对媒体流的消费速度.

2.4 性能分析

为了分析 DMIoDS 的性能, 本文以 NFS(网络文件系统)为比较对象, 把服务器的硬盘通过 NFS 变成客户机的逻辑盘, 在客户端运行 MPEG 播放程序, 并和 DMIoDS 在点播 MPEG 文件时的性能作了比较. 通过对同时支持的用户数和播放的 QoS 作对比, 得到的结果见表 1.

表 1 对比结果

用户数	NFS	DMIoDS
1	播放正常	fps=30
2	播放正常	fps=30
3	不连续	fps=30
4	抖动严重	fps=30
5	无法播放	fps=30
6	—	fps=26
7	—	fps=22, 无抖动
8	—	fps=20, 较连续
9	—	不接纳新用户

表中 fps 表示每秒的帧数. 可以看出, DMIoDS 在允许接纳新的用户的前提下, 能够保证用户的 QoS 要求; 通过负载均衡机制, 当有较多的用户同时得到服务时, 系统能够保证不会出现较大的延时抖动. 另外, 在每个周期内, 服务器方使用的是批处理操作, 这样使得网络资源冲突的概率大大减少. 客户端有效的缓冲机制, 也缓解了延时抖动现象.

应当说明的一点是: 在局部范围内, 原型实现系统的部件之间没有严格物理上的界限. 不同的部件可以由不同的主机服务器来承担, 也可以是几个不同的部件放在同一个服务器上, 如 RS6000 可以当作一个网管服务器, 而且一些静态数据库和程序库也可以存放其上.

3 结束语

本文通过分析 DMIoDS 的特点, 提出了一种 DMIoDS 的层次功能模型, 并且根据该模型实现了一个原型系统. 该原型系统支持视频流的可伸缩性编码, 实现了媒体对象服务器、多媒体信息的同步和重组、负载均衡机制和客户缓冲机制等, 取得了预期的结果. DMIoDS 模型对指导系统的设计与实现起着重要的作用, 同时, 原型系统的设计与实现也证实了 DMIoDS 功能模型的合理性. 在未来的工作中, 随着硬件条件的不断成熟, 我们将在支持具有可伸缩性的视频服务器和网络体系结构上作进一步的实现.

参考文献

1 Rangan P V, Vin H M, Ramanathan S. Designing an on-demand multimedia service. IEEE Communications Magazine, 1992.

- 430(7):56~64
- 2 Chang YH, Coggins D *et al.* An open-systems approach to video on demand. *IEEE Communications Magazine*, 1994, 32(5):68~80
 - 3 刘积仁, 都军, 张霞. 分布式多媒体技术的研究及其展望. *电子学报*, 1995, 23(10):44~50
(Liu Ji-ren, Du Jun, Zhang Xia. Research and prospect on distributed multimedia techniques. *Journal of Electronics*, 1995, 23(10):44~50)
 - 4 Hewitt C. Offices are open systems. *ACM Transaction on Office Information Systems*, 1986, 4(3):271~287
 - 5 Sun Microsystems Inc. Java remote method invocation specification. Revision 1.0, Final, 1997
 - 6 Gen M, Cheng R. Genetic algorithms and engineering design. New York: John Wiley and Sons, 1996

A Model of Distributed Multimedia Information-on-demand System

LIU Ji-ren ZHANG Ying-hui WANG Xing-wei JIANG Zao

(Software Center Northeastern University Shenyang 110006)

Abstract As an extension of VoD (video-on-demand) system, DMIODS (distributed multimedia information-on-demand system) has become a new application field of multimedia information processing technology. By analyzing several key problems involved in designing the DMIODS, the authors present a kind of abstract hierarchical functional model of DMIODS in this paper. According to the functional model, they present a prototype system and implement it.

Key words Distributed multimedia information-on-demand, functional model, prototype system, media object server, load balancing.