ISSN1000-9825 Journal of Software 软件学报 2000,11(1):40~48

# 分布式多媒体数据库系统*

巩志国[1] 周龙骧[2] 董淑珍[3]

[1](中国科学院软件研究所对象技术中心 北京 100080)

[2](中国科学院数学研究所 北京 100080)

[3](河北师范大学计算机科学系 石家庄 050016)

E-mail: lxzhou@math20.math.ac.cn

**摘要** 着重分析分布式多媒体数据库系统的特有问题,包括实时性、时空同步以及服务质量的描述与实现等.给出了解决这些问题的策略和方法.研究了各种不同的实现途径——远程调用、中间件和Agent构造的方式;指出了这些实现方法的优点及其不足.

**关键词** 分布式多媒体数据库系统,实时性,同步,服务质量,中间件,Agent.

**中图法分类号** TP311

如果数据库分布在多个计算机的辅助存储器上,把驻留计算机及辅助存储器的地方称为一个结点(node)或一个站点(site),各结点间用链路连接构成计算机网络,这些物理上分布在计算机网络结点上的多个数据库,如果在逻辑上能把它们作为一个统一的数据库来看待时,我们就称此数据库为分布式数据库.换句话说,分布式数据库是一组结构化数据的集合,它们在逻辑上属于同一系统,而在物理上分布在计算机网络的不同结点上,其网络环境可以是局域网,也可以是广域网.分布式数据库除了具有集中式数据库所具有的特点之外,还有其固有的特点,如,自治与共享、可控制的冗余、事务的分布执行等等.这些基本问题到80年代中期已经研究清楚,并且许多原形系统,如SDD-1,POREL,ADA-DDM,SIRIUS-DELTA,MULTIBASE,R* 以及Distributed INGRES和C-POREL也已经开发成功[1].多媒体数据的纳入首先使集中式的数据库系统发展成能够管理多媒体数据类型的集中式多媒体数据库管理系统,而传统的分布式数据库系统也就发展成分布式多媒体数据库管理系统.但是,这种发展不仅仅是数据类型的扩充,因为多媒体数据与传统数据的管理要求有着本质的不同.所以,我们必须认真考虑由于多媒体数据类型的纳入所带来的新问题.Adjeroh和Nwosu[2]以及Pazandak和Srivastava[3]分别研究了集中式多媒体数据库系统的若干问题,在此我们不拟重复,而着重考察分布式环境下所产生的新问题.本文第1节着重研究和讨论多媒体的实时性、同步特性以及服务质量特性对分布式多媒体数据库系统的设计所造成的影响.第2节分析分布式多媒体数据库系统的基本结构特点.第3节研究不同的实现技术.最后总结全文.

## 1 分布式多媒体数据库系统的新特色

传统的分布式数据库系统提倡有一定的数据冗余,以减少数据通信的代价,增加局部数据库的自治性.但是,多媒体数据的尺寸庞大,冗余存储这样的数据会造成严重的资源浪费,因此,这种冗余方式在一般情况下对于分布式多媒体数据库系统已不再大力加以提倡.传统分布式数据库系统的其他特点仍适用于多媒体的情况.除此之外,分布式多媒体数据库系统还增加了一些新的特点.

---

* 本文研究得到国家863高科技项目基金(No.863-306-02-07-1,863-306-03-07-5)、国家科技部重点项目基金和部委级基金(No.7A.3.1.2)资助.作者巩志国,1963年生,博士后,主要研究领域为分布式多媒体数据库系统,数字图书馆.周龙骧,1938年生,研究员,博士生导师,主要研究领域为分布式多媒体数据库系统,电子商务.董淑珍,女,1964年生,讲师,主要研究领域为形式化方法,数据库应用.

本文通讯联系人:周龙骧,北京100080,中国科学院数学研究所

本文1998-09-28收到原稿,1999-08-16收到修改稿

© 中国科学院软件研究所 http://www.jos.org.cn

### 1.1 实时性

多媒体数据的展示具有实时性,即数据按一定的速率(帧数/s,样值数/s)进行展示.传统的分布式数据库系统总是将查询的结果一并传送到用户所在结点,对于具有实时性的多媒体数据已不再适合,其原因有两方面.(1)用户的等待时间过长.具有时序属性的多媒体对象的尺寸巨大,将整个数据对象一次性地传送到用户所在结点会花费很长的时间.(2)用户所在结点的存储容量有限.用于存储连续媒体的结点计算机一般都要采用磁盘阵列,甚至三级存储方式以满足连续媒体大数据量的需求,但网络系统中不可能每个结点都具有这样的存储容量.因此,数据的提供结点和消耗结点之间是实时提供和实时消耗的关系.这种要求对传统的网络通信协议提出了新的挑战.

传统的通信协议(TCP/IP,Ethernet,X.25)已不能满足数据的实时传输的需求.连续媒体需要高速、等时传输,而对于传输时所产生的误码并不特别敏感;传统数据对传输的正确性要求较高,但对传输时的延时与抖动并不敏感.传统的通信协议提供的是无误的、点到点的通信,这种方式不能够有效地支持等时的(Isochronous)、多点对多点的通信.由此可见,传统的通信协议已不再能满足远程电视会议(teleconferencing)以及协同工作(cooperative work)等的应用需要.适合于分布式多媒体数据库系统的通信协议应该满足如下要求[4].

(1)能够根据用户对服务质量(quality of service)的描述,均衡协调传输速度、等时要求以及错误恢复各要素之间的关系;

(2)支持网络中两种连接类型的共存,即实时性的连接和非实时性的连接;

(3)支持多点对多点的通信;

(4)通信协议实现的负荷要小,以便有更多的时间用于对数据的传输.

针对这种需求,现在已经开发出了一些通信协议,如XTP,TP5,Heideberg Transport System,Tenet,ST-II,RSVP等[5].另外,将当前Internet的单点对单点、尽力型(best effort)通信协议修改成适合于多媒体传输的通信协议也是一种趋势.Internet通信的核心为IP协议,所以许多研究着眼于对原IP协议层的改进.另一种研究趋势是将ATM网络通信协议定义成两种类型:(1)实时保证的、延时限制的、等速传输协议;(2)尽力型的、速度可变的协议[6].

### 1.2 同步特性

对于传统的分布式数据库系统,其查询结果的输出,一般只考虑各片段的联结运算.对于多媒体数据而言,不但要考虑内容上的合成,而且还要考虑各多媒体对象在时序以及空间上的合成.我们需要在同步的描述和同步的实现这两个层次上考虑多媒体对象的同步集成.

#### 1.2.1 同步的描述

采用什么样的方法去描述多媒体对象之间的同步集成关系,是一件值得考虑的问题.这方面的研究成果比较多,其中典型的有时间Petri网[6~8]、时空算子方式[9]、文档方式[10,11]、面向对象模型方式[12]、链接方式[13,14]等.

由于时间Petri网对并行、并发及串行的表示非常有力,从而被人们普遍采用.这种方法的特点是直观,对于时间同步具有很好的表达能力.但这种方法一般要求事先确定出媒体对象的展示时间.这一点过于苛刻,因为由于用户的交互及系统负荷的变化,都可能动态影响展示时间.用户容许有一定的抖动(jitter)和延时(latency).

算子方式是用符号表示,由Allen[15]所定义的13种时序关系,进而多媒体对象之间的时序同步可以用这些算符与对象的有效表达式来表示.这种模型较为抽象,实现起来较为困难,而且很难体现多媒体集成的动态特性.

文档模型集多媒体文档的逻辑结构、空间布局结构和展示结构于一体,突破了传统文档模型仅包含逻辑结构和布局结构的局限,通过展示结构刻画多媒体数据的展示集成.但这种模型对用户的交互支持不足.

面向对象模型是利用对象的属性来描述多媒体对象之间的时空关系,链接方式则是利用链的概念来刻画多媒体数据之间的时序关系.它们都没有提供对用户交互的支持.由于同步描述模型所存在的上述各种问题,所以我们设计了Agent同步集成模型[16],以保证多媒体同步描述的力度和灵活性.Agent模型是一种动态模型,它不但支持多媒体对象之间的内容集成、时序集成以及空间集成,而且还支持用户的交互.有关Agent同步集成模型

© 中国科学院软件研究所 http://www.jos.org.cn

的详细论述见文献[16].

1.2.2　空间同步的实现

在分布式环境中实现多媒体对象之间的空间集成时,必须考虑网络通信的代价.典型的空间合成包括覆盖(overlay)、交搭(overlap)、毗邻(abut)、裁剪(crop)、放缩(scale)、字型格式变换(text format)等.这些空间合成运算是在用户所在的结点进行,还是在邻近数据所在的结点进行,对网络的通信负荷会有很大的影响.例如,两个图像 $A$ 和 $B$,它们均存储在同一个结点上,用户可能需要先分别对 $A$ 和 $B$ 进行 Crop 操作,然后再对得到的结果进行二元运算 Overlay(如图 1 所示),即用户所需要的最后结果为 $C=Overlay(Crop(A),Crop(B))$.显然,此时如果将合成后的结果传送到用户所在结点要比分别将 $A$ 和 $B$ 传送到用户结点后再进行合成所造成的通信代价低得多.另外,即使 $A$ 和 $B$ 不在同一结点,我们如果能够选择邻近 $A$ 和 $B$ 的结点首先进行空间合成,往往也会增加整体的效率.因此,在分布式网络环境下,系统应权衡通信及合成点的选择对整体性能的影响.

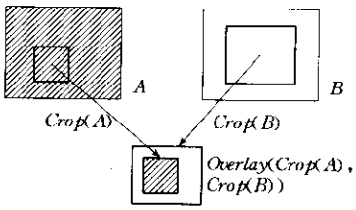



Fig. 1　The space combinational operation
图1　空间合成运算

1.2.3　时序同步的实现

多媒体时序同步既是分布式多媒体数据库系统的关键,又是其实现的难点.称其为关键,是因为时序同步是多媒体数据之间的本质关系,其功能的强弱直接影响到系统对多媒体需求的支持程度;称其为难点,是因为在分布环境下实现这一技术非常困难.这也是当前多媒体技术研究中的一个热点[17~19].

多媒体同步分为内同步(intramedia synchronization)和外同步(intermedia synchronization).内同步是指连续媒体内部帧之间的时序同步,外同步则指的是不同媒体之间的时序同步.在分布式环境下,影响多媒体同步实现的主要因素包括:

(1) 存储设备读取延迟,

(2) 网络延迟,

(3) 传输过程中的抖动,

(4) Client 端以及 Server 端内存的大小,

(5) 多媒体对象的尺寸等.

用户对多媒体展示的需求通过脚本描述(scenario)来刻画,计算机系统将这种描述转换为相应的展示调度方案(presentation schedule)以及相应的存取调度方案(retrieval schedule)(如图 2 所示).存取调度方案描述了每一参与展示媒体的读取速率以及各媒体存取操作的时序关系.因为每一媒体的大小、所在存储设备的性能以及网络的带宽等因素可能会有不同,所以这些媒体数据从存储服务器到客户端的传输时间可能会有偏差.为了满足展示时的时序要求,系统需要产生相应的存取方案.由此可见,媒体的存取调度顺序与展示调度顺序可能会有所不同.

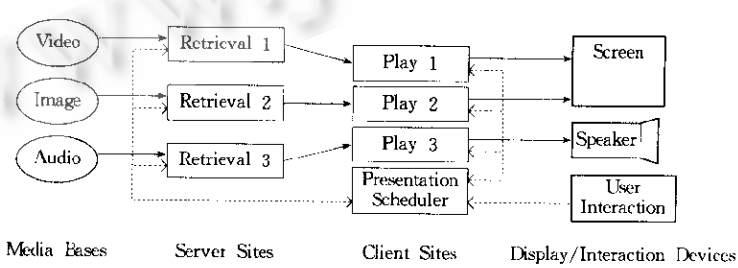



Fig. 2　The synchronized control in distributed environment
图2　分布式环境下的同步控制

在分布式环境下,端到端的延迟非常随机,每一媒体对象都需要较长的响应时间.为了平滑服务器的计算速度、网络传输速度和媒体播放速度之间的波动,在服务器端以及客户端需要有足够的内存缓冲区.文献[17]详细

© 中国科学院软件研究所　http://www.jos.org.cn

分析了不同媒体类型对内存的占用情况.

用户的交互操作(interaction)是多媒体系统的重要功能需求之一,而用户交互又会对多媒体的展示调度方案和存取调度方案产生很大影响.在分布式环境下,基于交互(interaction-based)的同步包括以下两个方面的问题.

(1) 再同步的确定.也就是说,当交互操作产生时如何调整展示调度顺序(外同步)媒体的播放速度(内同步).

(2) 位置的确定.即在客户端与服务器端缓冲区中,媒体块之间的对应关系如何再定位.

用户的交互操作具有随机性和多样性.随机性是指用户的交互可能发生在任何时间点上;多样性是指存在多种交互操作,包括常规操作、随机操作、变速操作和回放操作.系统应该能够根据各种交互操作产生相应的再同步和再定位策略.

(1) 常规操作.包括播放(play)、停放(stop)、暂停(pause)以及恢复播放(resume).这类操作不改变展示调度的顺序和展示的速度.再同步以及展示调度与存取调度之间的协调比较简单.

(2) 随机操作.包括向前跳转(jump-forward)、向后跳转(jump-backward)和链接跳转(hyperjump).这类操作虽然不改变媒体的播放速度,但媒体之间的调度顺序往往会有所改变.再者,客户端与服务器端的缓冲数据块之间需要重新协调定位.

(3) 变速操作.包括加速(speed-up)和减速(slow-down).这类操作需要改变媒体的播放速度,但调度顺序不变.

(4) 回放操作.包括正常回放(reverse)、快速回放(fast-reverse)和慢速回放(slow-reverse).再同步后的调度顺序可能恰好与原调度顺序相反.正常回放不改变播放速度,而快速回放和慢速回放都要改变播放速度.

### 1.3 服务质量

即使网络的频带再宽、计算机的内存再大、内存调度算法再优秀,系统的资源也是有限的,不可能满足无限多用户的需求.另一方面,各个用户针对自己的具体应用,并不一定需要系统必须严格按存储数据的缺省分辨率、播放速率和展示尺寸提供.为了能够得到系统的服务,用户一般容许有一定的质量降低,这与用户对传统数据的需求有很大差别.例如,字符数值串所反映的含义会因为串中某一码的丢失或错误而不能被理解.但对于视频、声频、图像等多媒体数据,即使其分辨率降低、部分码丢失,用户一般也能理解其展示的含义.所以,为了尽量多地增加用户数量,充分满足不同用户的需求,系统应提供服务质量(QoS)的描述接口.

#### 1.3.1 服务质量的基本概念

传统的查询接口语言(如SQL)只需要用户给出要什么(what),而由系统负责如何(how)去查询所需数据.为了支持多媒体数据,系统应允许用户描述"需要什么"、"何时展示"、"在屏幕的什么位置展示"、"按什么质量传输数据".Gibbs利用同多媒体数据封装在一起的属性来刻画多媒体数据的服务质量[20],但是这种方式给出的只是多媒体数据的表示质量,而不是展示质量.这是因为在多媒体数据实际展示时不一定会按原表示的质量传送给用户.由此可见,对QoS的描述必须考虑到用户展示设备的状况和用户对展示的特殊需求.Staehli和Walpole等人[21]通过引入"内容"(content)、"视"(view)、"质量描述"(quality specifications)等概念来刻画QoS."内容"描述了多媒体对象的逻辑输出值,实际上就是指数据表示中的缺省显示尺寸、分辨率、播放速率等."视"描述了在某一时间区间内,从"内容"到物理显示区域的映射."质量描述"反映的是从"内容"到"视"的实际展示与理想展示之间匹配程度的阈值,如果超过阈值就不能满足用户的QoS需求.我们则采用Agent模型作为多媒体对象展示集成的描述方法.在Agent中,用户可描述实际的展示尺寸、分辨率、播放速率等参数,只需在我们的Agent模型中增加服务质量的阈值参数就可以实现用户对QoS描述的完整需求.

#### 1.3.2 服务质量的实现

多媒体对象的展示特性可由一个多维向量$\boldsymbol{P}=(P_1,P_2,\ldots,P_n)$来刻画,其中每个分量代表某一个基本的展示特性(如展示尺寸、分辨率、展示速率等).用户对服务质量的描述可抽象地表示为$QoS=(Q_1,Q_2,\ldots,Q_n)$,其中$Q_i=(Q_{min_i},Q_{ideal_i},Q_{max_i})$.这里,$Q_i$描述了用户对基本展示特性$P_i$的质量要求,其中$Q_{ideal_i}$是用户对$P_i$要求的

© 中国科学院软件研究所 http://www.jos.org.cn

理想参数,$Q_{min_j}$和$Q_{max_j}$分别表示$P_i$的下限和上限.我们用$R=(R_1,R_2,...,R_m)$表示系统为用户的展示请求而分配的系统资源,其中$R_i$描述了某一基本的资源分配情况(如CPU占用时间、内存大小、网络带宽等).在执行时,系统应将向量QoS映射为资源分配向量$R$.最为常用和最为简单的映射方法是在展示请求开始时,通过用户和系统之间的协商(negotiation)来确定系统对展示请求的资源分配.一旦资源分配成功,在整个展示过程中,系统将保持为用户所分配的这些资源.在局域网络环境下,这种方式比较简单,较易实现.但在大型网络甚至Internet环境下,系统很难一直保持所分配给用户的系统资源,其主要原因有以下几个方面.

(1) 许多网络——如当前的Internet网——属于尽力型(best-effort type),这使得资源的保持非常困难,甚至根本不能达到.

(2) 分布式多媒体数据库系统可能包含若干个数据库服务器,它们所采用的通信协议与处理系统很不相同.如果用户结点与数据资源结点同属于一个局域网段时,资源的保持就较易做到.而如果用户在远程甚至通过Modem访问数据资源时,静态的资源保持就很难做到.

(3) 在大型网络环境下,经常出现临时性的网络通信拥塞,使数据传输的延迟以及传输出错率急剧上升.为了保证用户的QoS需求,系统和用户必须重新协商,以再次决定资源的分配,这显然增加了系统的复杂性.

(4) 学术界以及商业界极力倡导利用移动设备作为网络终端.这些终端设备通常是利用无线通信方式与主干网连接,在无线通信技术(红外线通信、区域性电磁波通信、卫星通信等)所提供的通信服务中,其数据的传输延迟和出错率较高,这也给静态资源的保持带来很大的困难.

鉴于以上静态QoS资源管理的不足,Jan Gecsei[22]讨论了动态QoS资源管理的方法(adaptation),其基本的工作原理为:在分布式系统中设置一个控制器(controller),该控制器感知多媒体对象传输和展示状态的变化,一旦这些参数超出了用户的QoS需求,控制器则触发一系列动作(action),以重新优化资源的分配(如图3所示).根据这一基本原理,我们作如下具体设计(如图4所示):(1) 在用户结点和数据资源结点分别设置一个多媒体展示状态监视器(QoS monitor agent)和资源优化器(resource optimizer agent);(2) 状态监视器一旦检测到当前的展示状态已超出用户的QoS所容许的范围,即刻将QoS参数的变化以短消息的方式传递到资源优化器;(3) 资源优化器重新产生对用户资源的分配参数,然后,资源管理子系统按照这些新的资源需求参数向用户提供多媒体数据,以满足用户的QoS需求.

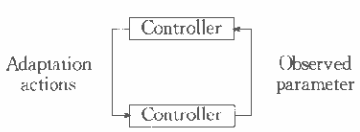
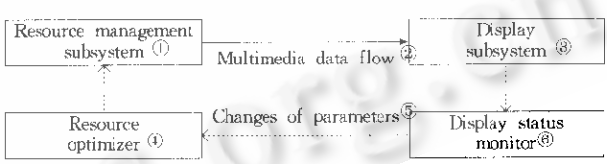



Fig. 3 The made of dynamic resource management
图3 动态资源管理方式




①资源管理子系统,②多媒体数据流,③展示子系统,
④资源优化器,⑤参数的变化,⑥展示状态监视器.

Fig. 4 The implementation structure of a dynamic QoS resource mangement
图4 一种动态QoS资源管理的实现结构

这种动态的QoS资源管理模式具有以下优点.

(1) 分布式系统较健壮.系统不会因为QoS请求不被满足而重新进行初始对话.

(2) 对用户的透明性好.在展示过程中,系统自动调整资源的分配而无需用户的干涉.

(3) 资源的利用率高.在运行过程中,系统动态优化每个用户所占用的资源,充分挖掘资源的可利用潜力.

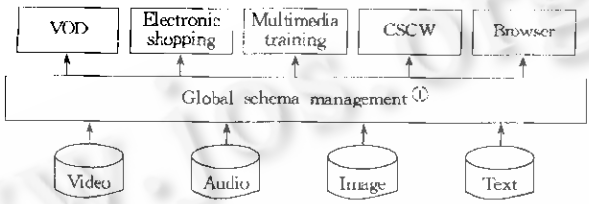
### 1.4 体系结构

分布式多媒体数据库系统的体系结构如图5所示.各种不同的局部多媒体数据库经过全局模式层得到统一.全局模式管理完成数据的一致性维护、进行必要的模式变换以及对全局事务的并发控制,使数据的结构、物理存储、分布性等对外透明,向上提供一个统一的多媒体操作接口.接口具有对多媒体同步和QoS描述的功能以及多媒体基于内容查询等功能,以支持各种类型的多媒体应用,如视频点播、电子购物、多媒体教学、群件系统、多媒体浏览与查询等.

© 中国科学院软件研究所 http://www.jos.org.cn

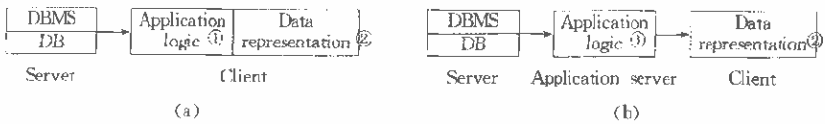
### 1.5 客户/服务器结构

上面我们已经给出了分布式多媒体数据库系统(MDDBMS)的主要特点.在理想的、具有完全透明性的MDDBMS中,计算机网络上的各个结点是对称的、平等的,不存在任何中心,某些结点的崩溃不会影响整个系统的运行.但是,这种完全的透明性实现起来极为困难,性能很不理想,尚不存在一个这种意义下的商业化系统.当前走红的则是客户/服务器结构的系统.从数据库管理系统的角度看,这种结构是一种集中式的DBMS.由传统集中式的数据库管理系统演变到客户/服务器结构的系统是一种平滑的适化过程[1],其结构如图6(a)所示的两层结构.这种结构上的划分充分利用了用户端的计算机(一般是微机)的处理能力,减轻了服务器的负荷.另外,为使具有相同应用逻辑的各不同用户共享应用事务的处理系统,进一步将应用逻辑管理功能与数据表示功能相分离是一种改进结构.这种三层结构首先使各不同应用共享DBMS,而具有相同应用逻辑的不同用户又可共享应用事务的管理,其结构如图6(b)所示.




①全局模式管理.

Fig. 5 The architecture of the Distributed Multimedia DBMS
图5 分布式多媒体数据库系统的体系结构




①应用逻辑,②数据表示.　　①应用逻辑,②数据表示.

Fig. 6 Client/Server structure
图6 Client/Server结构

由于多媒体数据的复杂性,多媒体数据库一般采用面向对象的机制,将多媒体数据、展示属性、操作方法封装在一起,以降低用户使用多媒体数据的复杂性.传统上,面向对象数据库的客户机与服务器之间的数据交换方式主要有以下几种[23].(1) 结果交换.客户发出请求(request),而服务器根据客户的请求只将数据的处理结果传递给客户端.其特点是降低了通信负荷,但同时也增加了服务器的处理负担.(2) 对象交换.服务器根据客户的请求,将整个对象传递给用户端,即使用户只需要对象的一小部分.这种方式与第1种情况相比,减轻了服务器的负担,但增加了通信负担以及客户端的处理负荷.(3) 页面交换.在这种方式下,服务器传送的是客户所需对象所在的页面,显然,这种方式使服务器的处理负荷最小,而通信及客户机的负荷最大.

对于多媒体数据库来讲,首先,第2种和第3种交换方式均不可取,因为:(1) 多媒体对象的数据量巨大,而客户机一般采用微机或工作站,其存储的容量有限,若将整个多媒体对象或对象所在的页面整个缓冲在客户端会有很大困难;(2) 通信的负荷过大,使用户等待时间过长;(3) 客户机对于多媒体对象之间的实时同步合成的处理能力有限.对于第1种交换方式,也必须先由服务器完成对象之间的时空同步处理,以流的方式,增量均速地传递给客户机,亦即服务器向客户机的数据提交为Push方式,而不是Pull方式.一般来讲,客户机与服务器之间的通信被逻辑地划分为上行频带(客户机到服务器)和下行频带(服务器到客户机)两种方式.

上行频带用于传递用户的请求命令,要求有很高的可靠性,而对实时性的要求不高.下行频带则用于多媒体数据的传送,对于实时性有严格的要求,而对于数据的正确性要求不高.

## 2 分布式多媒体数据库的实现途径

我们在前文从概念上分析了分布式多媒体数据库系统的基本特点和结构,实现时则可以利用不同的技术

© 中国科学院软件研究所 http://www.jos.org.cn

手段.

### 2.1 远程调用范型

远程调用(RPC)是进程调用最自然、最直接的扩充,是实现 Client/Server 结构时最原始的方法.一般在客户端含有适于各种开发语言的接口定义以及相应的编译器.远程调用语句经过编译器翻译成对服务器的调用码,由通信机制传送给服务器,再由服务器端将这些调用码翻译成局部的进程调用,以完成远程服务.远程调用的实现还包括一个实时(run time)库,用于实现网络通信.OSF DCE(open software foundation's distributed computing environment)成为这种调用方式的一种标准.利用这种方式实现多媒体数据库的 Client/Server 结构时,必须对传统的 OSF DCE 定义标准进行必要的扩充:(1) 增加对连续媒体(Video,Audio)操作的调用定义.传统的 RPC 调用方式一般只适合于短消息的通信控制,而连续媒体的操作则要求批量数据的均匀的、长时间无干扰的通信控制.(2) 增加对多媒体同步描述的功能.多媒体数据之间的时序同步是本质性的需求,接口中应包含对同步描述的定义.(3) QoS 定义功能.RPC 接口标准应该能够处理用户对服务质量的定义,以便尽可能多地增加用户数.

远程调用方式虽然使 Client 端与 Server 端能够具有较好的耦合性,运行的效率也比较高,但存在一些不足.(1) 开发的效率较低.企业可能有多个数据库服务器,用户必须了解它们所提供的不同接口、数据的分布情况,以协调对各个服务器的调用.(2) 软件的复用性差.企业的各个数据库服务器可能是在不同时期建立的,当应用中需要利用各服务器中的数据共同完成某项任务时,用户不得不在 RPC 接口层上重新编辑应用程序,这样就极大地浪费了人力和物力.为了提高分布式系统构造的灵活性和软件的可复用性,屏蔽掉各厂家数据库服务器接口的不统一,一种新的技术——中间件(middleware)——应运而生.

### 2.2 中间件技术

从广义上讲,中间件的作用是为了屏蔽不同操作系统接口的差异及分布性,为用户提供一个统一的应用开发接口[24].在本文中是指为了屏蔽各数据库接口的不同及数据的分布性而提供的一个统一的接口软件层,可以通过这个软件层透明地访问异构的多媒体数据库系统,中间件在系统中的位置如图 7 所示.虽然中间件尚无严格的定义,但它的主要特点是比较一致的,即跨越多个应用;运行于各个不同的数据库之上;具有分布特性;支持标准的接口和协议.

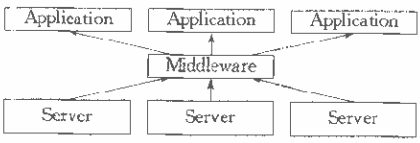



Fig. 7 The position of the middleware in system
图7 中间件在系统中的位置

ODBC 是当前被广泛采用的中间件技术,不同的数据库厂家(Oracle,Sybase,Informix,DB2 等)都提供了支持这种标准的驱动器,不同模式的数据库系统接口在 ODBC 中得到统一,用户只需根据 ODBC 所提供的标准接口,透明地访问各数据库服务器.当前,ODBC 所提供的接口是一种标准的 SQL 语言.由于多媒体数据库访问的特殊需求,需对 ODBC 进行扩充.

分布式面向对象范型是另一类中间件技术,这一范型标准充分利用面向对象模型的优点,屏蔽掉由于对象所在平台的不同、位置的不同以及对象迁移所造成的问题,为用户提供一个一致的、分布透明的面向对象接口.因为多媒体数据库一般倾向于采用面向对象的范性实现,所以这种接口方式对于多媒体数据库的分布式构造较为平滑.这一范型的工业标准较多,如 Microsoft 公司的 COM(component object model)、IBM 公司的 SOM(system object model)、国际标准组织 OMG(common management group)的 Corba 等.这类中间件标准同样需要增加多媒体数据访问特性(实时性、同步性、QoS)的支持,才能真正适合分布式多媒体数据库的构造.

另外还有一种,虽然还没有标准化,但理论上却非常完美的技术,即利用 Agent 模型构造分布式系统[25].Agent 技术是分布式人工智能领域中发展起来的一种新型计算模型,具有智能化程度高、分布式系统构造灵活、软件的复用性强等优点.Agent 能够感知外界发生的消息,根据自己所具有的知识自动作出反应.利用这一技术实现分布式多媒体数据库系统时,应考虑:(1) 如何将已有的计算实体(多媒体对象、元组、一般的数据文件等)构造成 Agent;(2) 如何实现 Agent 之间的通信.

改造原计算实体的方法一般有 3 种,即 Transducer(变换)、Wrapper(包装)和 Rewrite(重写),如图 8 所示.

© 中国科学院软件研究所 http://www.jos.org.cn

对外界来讲,Transducer 具有 Agent 的功能,当它感知到外界的消息后,会根据自身内部的机制将外部消息分解、转换为对原计算实体的调用.这种方式对分布式多媒体数据库系统的实现较容易,因为它充分利用了多媒体对象所提供的操作接口,无需破坏对象的封装性.Wrapper 方式是直接访问原数据对象的数据结构,并有可能修改数据的结构以向外提供 Agent 的接口.这种方法从运行效率上讲,要比第 1 种方式高,但改造的工作量相对增加了.Rewrite 方式是重写原计算实体,完全用 Agent 的方式构造其计算结构,所以这种方法最彻底,运行效率最高,但改造代价要比前两种都要大.改造后的分布式多媒体数据库系统如图 9 所示.

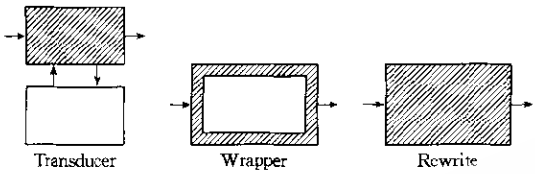



Fig. 8 The reconstruction mode of traditional computing component
图8 传统计算实体的改造方式

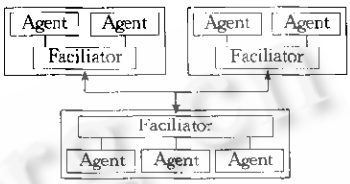



Fig. 9 The distributed agent construction
图9 分布式的Agent构造结构

Agent 之间的通信实际上是通过 Faciliator 完成的,Faciliator 概念来自对 Mediator 的推广与发展[25].Agent 之间的通信实际上首先是与同一结点上的 Faciliator 通信.Faciliator 负责将消息分解,然后发往不同结点上的 Faciliator,再由本地的 Faciliator 传送给相应的 Agent.

我们采用 Transducer 方式,将多媒体展示集成描述用 Agent 来刻画[16],虽然我们的 Agent 模型还谈不上什么国际标准,但这种方法对多媒体对象的操作描述非常有力,充分利用了多媒体对象的消息接口,维护了对象的封装性,多媒体对象的存储位置、定义语言对外透明,支持多媒体对象的时空同步定义及 QoS 描述.进一步地,这种模型利用多媒体对象的展示状态来控制多媒体对象的同步关系,以动态适应网络与系统负荷的变化.

总之,中间件技术是实现分布式环境的有效方法,它利用中间件屏蔽掉各数据库服务器接口的不同,向上提供统一的、分布透明的开发接口.ODBC 适合于关系数据库服务器的分布式构造,面向对象范型的中间件适合于面向对象数据库的集成,而 Agent 更适合于多媒体数据库的分布式建立.

## 3 结束语

网络环境下的多媒体应用系统已有许多,如视频点播、电子购物、远程电视会议等.但这些都是非常专门化的、局部性的应用,各应用系统之间一般是毫无关系的.随着信息领域内技术的发展,人们希望能够在较大范围,乃至全球范围内,真正并且逻辑上一致地共享各种资源.这就要求我们必须从分布式数据库系统的高度来看待这一问题,以便真正管理好类型繁多的多媒体数据资源.目前,对分布式多媒体数据库系统的研究尚不多见,其主要原因是这一问题的困难性.我们通过与传统的分布式数据库系统的比较,分析了分布式多媒体数据库系统的固有特点及其结构,研究了各种实现途径的优劣.由于篇幅所限,对于分布式多媒体数据库系统与集中式多媒体数据库系统之间的共有问题,我们在此没有进行讨论,可参见相应的文献.

**参考文献**

1 Zhou Long-xiang, Chai Xing-wu. Hierarchical architecture of distributed multimedia database systems. Chinese Journal of Computers, 1996,19(7):481~491
(周龙骧,柴兴无.分布式多媒体数据库系统的分层体系结构.计算机学报,1996,19(7):481~491)

2 Adjeroh D A, Nwosu K C. Multimedia database management-requirements and issues. IEEE Multimedia, 1997,4(3):24~33

3 Pazandak P, Srivastava J. Evaluating object DBMSs for multimedia. IEEE Multimedia, 1997,4(3):34~49

4 Muhlhauser M, Geesei J. Services, frameworks, and paradigms for distributed multimedia applications. IEEE Multimedia, 1996,3(3):61

5 Steinmetz R, Nahrstedt K. Multimedia: Computing, Communications, Applications. Upper Saddle River, NJ: Prentice Hall, Inc., 1995

6 Little T D C, Ghafoor A. Synchronization and storage models for multimedia objects. IEEE Journal on Selected Area in

© 中国科学院软件研究所 http://www.jos.org.cn

Communications, 1990,8(3):413～427
7 Diaz M, Senac P. Time stream Petri nets: a model for multimedia streams synchronization. In: Chua Ta-seng, Tosiyasu L Kunii eds. The International Conference on Multimedia Modeling (MMM'93). Singapore: World Scientific Press, 1993. 257～273
8 Al-Salqan Y Y, Chang C K. Temporal relations and synchronization agents. IEEE Multimedia, 1996,3(2):30～39
9 Schloss G A, Wynblatt M J. Providing definition and temporal structure for multimedia data. ACM Multimedia Systems, 1995,3(5):264～277
10 Khalfallah H, Karmouch A. An architecture and a data model for integrated multimedia documents and presentational applications. ACM Multimedia Systems, 1995,3(5—6):238～250
11 Karmouch A, Emery J. A playback schedule model for multimedia document. IEEE Multimedia, 1996,3(1):50～61
12 Iino M, Day Y F, Ghafoor A. An object-oriented model for spatial-temporal synchronization of multimedia information. In: Proceedings of the 1994 International Conference on Multimedia Computing and System. Washington, DC: IEEE Press, 1994. 110～120
13 ISO. Hypermedia/time-based structure language: HyTime (ISO 10744). International Standard Organization, 1992
14 ISO. Multimedia and hypermedia information coding expert group. ISO/IEC JTC1/SC29/WG12, MHEG Working Draft "WD. 1.0", Version 1.0, 1993
15 Allen J F. Maintaining knowledge about temporal intervals. Communications of the ACM, 1983,26(11):832～843
16 Gong Zhi-guo, Zhou Long-xiang. An agent model for integration of multimedia object presentation, Journal of Software, 1999,10(1):35～42
(巩志国,周龙骧.多媒体对象的 Agent 展示集成模型,软件学报,1999,10(1):35～42)
17 Liao W, Li V O K. Synchronization of distributed multimedia systems with user interactions. Multimedia Systems, 1998, 6(3):196～206
18 Huang C M, Chen J S *et al*. MING-I: a distributed interactive multimedia document development mechanism. Multimedia Systems, 1998,6(5):316～333
19 Candan K S, Prabhakaran B, Subrahmanian V S *et al*. Retrieval schedules based on resource availability and flexible presentation specifications. Multimedia Systems, 1998,6(4):232～250
20 Gibbs S, Breiteneder C, Tsichritzis D. Data modeling of time-based media. In: Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 1994. 91～102
21 Staehli R, Walpole J, Maier D. A quality of service specification for multimedia presentations. Multimedia Systems, 1995,3(5—6):251～263
22 Jan Gecsei. Adaptation in distributed multimedia systems. IEEE Multimedia, 1997,4(2):58～66
23 Che Dun-ren, Zhou Li-zhu, Wang Ling-chi. The architecture of object-oriented database systems. Journal of Software, 1995,6(10):599～606
(车敦仁,周立柱,王令赤.面向对象数据库系统的体系结构.软件学报,1995,6(10):599～606)
24 Bernstein P A. Middleware: a model for distributed system services. Communications of the ACM, 1996,39(2):86～98
25 Genesereth M R, Ketchet S P. Software agent. Communications of the ACM, 1994,37(7):48～53

## Distributed Multimedia Database Systems

GONG Zhi-guo[1] ZHOU Long-xiang[2] DONG Shu-zhen[3]

[1](*Object Technology Center Institute of Software The Chinese Academy of Sciences Beijing* 100080)
[2](*Institute of Mathematics The Chinese Academy of Sciences Beijing* 100080)
[3](*Department of Computer Science Hebei Normal University Shijiazhuang* 050016)

**Abstract** In this paper, the typical problems in distributed multimedia database systems, such as real-time requirements, synchronization, QoS specifications, are analyzed. Policies and methods to solve these problems are provided. And various implementation paradigms for distributed systems, such as RPC (remote process call), middleware, Agent, are investigated.

**Key words** Distributed multimedia database system, real-time, synchronization, QoS (quality of service), middleware, agent.

© 中国科学院软件研究所 http://www.jos.org.cn