

面向对象数据库系统的体系结构*

车敦仁 周立柱 王令赤

(清华大学计算机科学与技术系, 北京 100084)

摘要 Client/Server 计算对目前软件系统的体系结构产生了很重要的影响, 特别是在数据库领域, 它几乎已被所有的面向对象数据库(OODB)无一例外地所选择. 本文综述 OODB 的 Client/Server 体系结构, 重点就页服务器结构及其 3 个变种, 及它们的性能改进措施进行深入地讨论.

关键词 Client/Server, 软件体系结构, 面向对象数据库, 页服务器.

关于体系结构(Architecture), 人们过去更多强调的是计算机系统和计算机硬件系统的体系结构. 随着软件系统规模和复杂度的日渐升级, 人们越来越认识到好的软件体系结构对于好的软件系统的重要性. 就体系结构的本意而言, 它是指建立系统时的构造范型(structuring paradigms)、构造风格(structuring style)和构造模式(structuring pattern)^[1]. 软件系统的体系结构也就是构造软件系统时所采用的构造范型、风格和模式. 软件体系结构对于软件系统的构造过程具有重要的指导作用, 同时它可以使软件设计师暂时抛开软件系统的功能细节来谈论或探讨软件系统的总体构架.

Client/Server 作为一种重要的体系结构范型, 它在软件方面的含义主要是指一个软件或应用系统的实现不是作为犹如磐石般统一的一整块, 而是被设计成包含很多成份的复合系统, 这些软件成份甚至可被分布于网络中不同的机器节点上, 并且依据软件成份的相对角色(roles)的不同区分为“客户”(Client)和“服务器”(Server), 客户软件能够请求服务器软件的服务^[2]. 到底什么是、什么不是软件的 Client/Server 体系结构曾引起过激烈的争论, 文献[2]给出的若干条准则代表了多数研究人员的共识. 使用 Client/Server 体系结构的一个主要动因是希望利用当前的工作站技术能够提供的丰富而廉价的资源. 把系统功能从服务器转移到客户端, 能够带给用户一系列好处^[2,3].

下面本文综述 OODB 的 Client/Server 体系结构, 重点就页服务器结构及其 3 个变种, 以及它们的性能改进措施进行深入地讨论.

1 OODB 的 Client/Server 体系结构

无论是集中式 DBMS 还是分布式 DBMS 作为一种软件, 其体系结构可以划分为两种基

* 本文 1995-01-09 收到, 1995-03-20 定稿

作者车敦仁, 1964 年生, 博士后, 主要研究领域为数据库, 知识库, 软件工程. 周立柱, 1947 年生, 教授, 主要研究领域为数据库. 王令赤, 1966 年生, 讲师, 主要研究领域为数据库, 软件工程.

本文通讯联系人: 车敦仁, 北京 100084, 清华大学计算机系软件组

本类型,即 Host/Slave 结构和 Client/Server 结构. 早期的多数关系数据库都是在只有字符终端和在主机上运行批程序的年代,围绕 Host/Slave(主/从)型计算模型而设计和建造的. 为方便起见,也把早期 DBMS 的体系结构称为 Host/Slave 模型. 从本质上说,这种体系结构让用户看到的是单一的进程,该进程把应用和数据库服务器集为一体^[4].

Client/Server 是近年来发展起来且在数据库界很流行的一种新型的 DBMS 体系结构,其真正要点是把数据应用视作是与数据库分离且不同的进程,它们可以在地理上分布,也可以同驻一台机器上. 由此看来 DBMS 的 Client/Server 体系结构恰好意味着 Host/Slave 体系结构的对应物^[4].

在具有 Client/Server 结构的 DBMS 中,数据的表现(Data Presentation)和数据的存取(Data Access)是分别作为客户与服务器的职责.

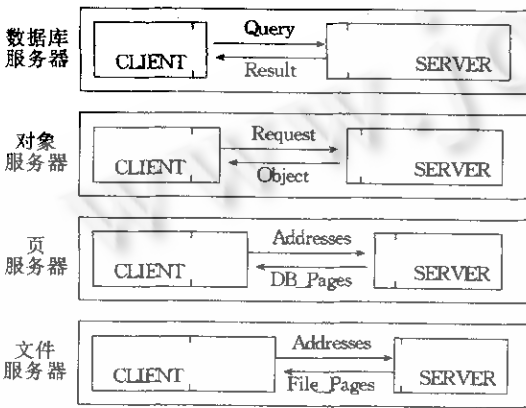


图1 OODB的四种Client/Server体系结构

OODB 把 Client/Server 体系结构又向前发展了一大步. OODB 的 Client/Server 体系结构可以划分为下列 4 种基本类型^[2,5-7],即数据库服务器(DB-Server),对象服务器(Object-Server),页服务器(Page-Server)和文件服务器(File-Server^[6]),见图 1. 它们的主要区别是对系统的客户成份和服务器成份所赋予的职责的多少不同,以及客户与服务器在交互时数据传输的粒度不同.

下面分别对这 4 种 OODB 体系结构及其优缺点进行论述.

1.1 数据库服务器

在数据库服务器体系结构中,几乎所有的数据库处理均在服务器方完成,只有极少的处理在客户方完成. 客户进程只是负责把应用的数据请求发送到服务器,并从服务器接收结果,然后再传送给应用^[2].

在 DBMS 的 Client/Server 体系结构中,数据库服务器是现代 RDBMS 广泛采用的一种体系结构,主要因为 RDBMS 诞生于大型机和小型机那个时代^[2]. 数据库服务器不适合 OODB 应用处理的特点,因而很少被具有 Client/Server 体系结构的 OODB 所采用^[2]. 因为 OODB 要支持的应用主要不是商务处理,而是诸如 CAD/CAM, OIS 和 AI 等所谓的非标准数据库应用. 这些非标准应用的特点是要对复杂对象进行较复杂的处理,于是常常要求长事务支持. 因而若采用数据库服务器体系结构,势必会导致服务器机器负载过重,从而无法满足非标准应用的性能需求. 鉴于目前工作站常常具有大量的剩余处理能力和虚存空间可资利用,因而,非标准应用非常希望把更多的工作放在工作站上来完成,一来可为非标准应用提供更快响应速度,二来可减轻服务器上的过载负荷. 对象服务器,特别是页服务器便是在这种情景驱动下的产物.

数据库服务器体系结构的优缺点均是显而易见的:其优点是需要进行网络数据传输的量少,其不足是服务器成为系统瓶颈,不能满足应用的快速响应要求.

1.2 对象服务器

对象服务器型的 Client/Server 体系结构把 DBMS 的处理功能在客户与服务器之间进行了基本上均衡的分布^[2]。服务器进程与客户进程进行交互的是逻辑数据单元,即单个的对象^[7]。服务器进程的典型职责是^[2,6]:保证数据完整性、进行查询的优化、执行存储过程、完成存储管理。因而需要管理锁,实施对磁盘的修改、维护和搜索索引结构,还要负责物理日志和恢复。而由客户进程来协调各事务,完成部分模式管理工作、并把持久对象表现到(present to)程序设计语言环境中^[2]。

在对象服务器体系结构中,工作站和服务器各自都维护有自己的对象缓冲区,当客户需要访问某一个对象时,先在自己的缓冲区中进行搜索,若未找到,再向服务器发出对象请求;若对象亦不在服务器的缓冲区中,服务器便从磁盘中检索该对象并回送给发出请求的客户;若对象(逻辑上)虽在服务器的缓冲区中,但已被其它客户加了相冲突的锁(conflicting lock),服务器还需要在本次对象请求与持有冲突锁的客户之间进行协调,只有当协调成功后才能把该对象发送给申请者并授予对该对象的相应的读/写权限。回调式封锁(callback locking^[7])是很适合 Client/Server 体系结构的一种并发控制技术,它是在出现锁冲突时,由服务器向每个持有冲突锁的客户发出对象回调请求(call back),各客户根据本地事务的实际使用情况做出相应的回答。本文在第 2 节还将对回调技术进行更详细的介绍。

对象服务器体系结构最本质的好处是可以在服务器与客户之间只传送消息和结果对象,少传送操作数对象。特别是可先在服务器上对大范围的对象进行筛选后,只把选中的对象传送到客户机上。假如筛选的命中率为 1%,就可节省 99%的通讯负荷。对象服务器的另一个好处是对方法执行地点的选择具有很高的灵活性。由于服务器和客户所看到的均是数据的逻辑单元(即 object),因而对象方法在任何一处都能执行,这样不仅便于选择计算成本较低的一方执行方法,也有利于平衡工作站和服务器之间的负载。因而对象服务器体系结构可以获得很好的响应时间、较大的系统吞吐率和较低的计算成本^[2]。

对象服务器也有下列一些严重缺点:(1)在最坏的情形,工作站上的每次对象引用都将导致一次远程过程调用。(2)服务器的设计极为复杂。(3)当一个方法应用于一组对象,而这组对象同时分布在工作站 cache、服务器 cache 和磁盘上时,情况是很糟的。(4)另外一些小问题:①不利于页面级的封锁(Locking);②对象将被多次拷贝;③当应用仅需一个大对象的一部分时,传输的却是整个大对象。(5)最后一个问题是与当前计算机技术发展的趋势(工作站高度发展)不合拍。

1.3 页服务器

在页服务器体系结构中,工作站与服务器之间进行数据交换的单元是页,该结构因此而得名。在此种结构中,服务器没有对象的概念,只能处理页面,也无法执行任何施用到对象上的方法。大部分的数据库处理是在客户一端,只有极少的工作在服务器方完成。服务器的主要职责是:(1)对磁盘页面进行存储和检索(也包括对磁盘资源的分配);(2)管理内存页面缓冲区;(3)提供并发控制和恢复服务^[6]。

页服务器体系结构中的处理方法与对象服务器体系结构的处理方法是相似的,所不同的是前者的数据传输、并发控制和复制管理均以页面为粒度,后者均以对象为粒度。页服务器体系结构的主要优点如下:(1)OODBMS 的绝大部分复杂管理功能均在工作站上进行,从而使服务器上的负载减小到最低限度。(2)由于在页服务体系结构中,单个服务器可以支

持更多的工作站,因而可免去了实现多服务器时的复杂性。(3)避免了上述对象服务器体系结构中遇到的所有问题。页服务器体系结构的主要缺点如下:(1)方法只得在工作站上执行。(2)对象级的封锁(Locking)较难实现。(3)非两段锁的 B 树封锁亦较难实现。(4)恢复要同时涉及到客户和服务器两方。(5)系统的性能对簇聚机制的有效性依赖较大。

1.4 文件服务器

文件服务器体系结构是对页服务器的进一步简化。工作站使用远程文件服务,如 NFS^[6],直接读写数据库页面,服务器的基本功能同页服务器体系结构中的服务器一样,主要进行并发控制和恢复管理。该结构基本具有页面体系结构中的全部优缺点。另外两个额外的缺点是:(1)NFS 的 Write 操作较慢(因无写缓冲);(2)工作站直接存取数据库页面,使 LOCK 请求与页面请求不能同步(要分离进行,造成多余的开销)。

1.5 小结

在 OODB 的 4 种 Client/Server 体系结构中,从数据库服务器到文件服务器,依次把愈来愈多的数据库处理从服务器转移到了工作站,这与目前工作站的处理能力及存储容量的迅猛提高这种发展势头是一致的。在数据库服务器和对象服务器这两种体系结构中,工作站与服务器间进行数据交换的单位均为对象,而在页服务器和文件服务器体系结构中数据交换单位均为页面。在 OODB 体系结构谱系中,数据库服务器与文件服务器分别处在两极,被目前的 OODB 产品或原型系统所采用的较少。RDB(进行了 OO 扩充)采用数据库服务器的较多,这与 RDB 的历史有关;Objectivity 采用的是文件服务器。早期的 OODB 多数采用了对象服务器体系结构,如 Itasca(及 ORION)、Versant 和 O2 的第 1 版(是一个原型系统)均属于对象服务器体系结构^[7]。较新的 OODB 产品系统更多地采用了页服务器体系结构,如 EXODUS、GemStone、O2(V2 以后)、ObjectStore、SHORE 和 MIDS/BUAA^[8,9]等均采用了页服务器体系结构^[7],ONTOS 则允许程序员在对象服务器与页服务器之间选择。页服务器之所以占取了主导地位,是因为此种结构与当前网络计算环境的发展趋势比较相符,以及前面已经述及的此种结构的种种好处。

不仅于此,页服务器体系结构在受到较新的 OODB 青睐的同时,还演变出了一系列不同的变体^[7],本文在第 3 节介绍 3 种重要的变体。

2 提高 OODBMS 性能的 3 个关键技术

在具体介绍 OODB 页服务器 3 种变体形式之前,我们先来介绍目前在 Client/Server 模型下提高 OODB 系统性能的 3 个关键技术。下一节要讨论的页服务器的 3 种变体形式均与这 3 个关键技术有关,或者甚至直接依赖于这些技术。

改进 OODBMS 系统性能的两个基本措施是对通信和服务器磁盘存取进行极小化,页服务器和对象服务器所共同采用的典型作法是允许客户机利用本地存储空间支持数据项的缓冲(cache),为了更加充分地利用客户机存储资源,甚至提出了跨事务缓冲(Intertxaction Caching^[7])的想法,跨事务缓冲需要专门的缓冲一致性维护协议,以确保客户看到的是一个协调的数据库视图。

在 Client/Server 环境下维护数据缓冲区的一致性,离不开并发控制和复制管理

(Replica mangement). 并发控制普遍采用封锁策略. 在 RDB/VMS 的自适应封锁算法 (adaptive locking algorithm) 中采用了“衰减锁”(lock de-escalation)^[10] 技术, 这与在主存数据库环境中独立提出的衰减锁思想几乎是完全相同的概念^[7]. 自适应封锁算法(或衰减锁技术)在多结点数据共享环境下有着非常好的性能优势^[7]. 而在复制管理中, 回调(Call-back^[7,11])技术是性能优势很明显的一种复制管理方法.

下面对跨事务缓冲、衰减锁和回调技术的思想精华再做更详细一点的介绍.

(1) 跨事务缓冲: 支持跨事务缓冲的目的是为了更充分地利用客户机内存, 提高对应用的数据请求的响应速度. 跨事务缓冲使得在客户机上维护的缓冲区内内容可跨越多个事务的执行. 在相继执行的事务流中, 处在后面的事务可直接利用在其前面执行的事务已获得的数据库子集, 而不必通过网络再向服务器申请所需的全部数据项. 这样不仅减轻了通信负载, 且能更快地响应客户的数据请求. 跨事务缓冲也相应地要求较复杂的缓冲一致性维护协议.

(2) 自适应封锁(或衰减锁): 衰减锁技术的基本思想^[7]是指锁粒度不是一成不变的(比如固定为页面级封锁或对象级封锁), 而是可以根据需要逐渐缩小. 其主要做法是在不存在并发控制冲突的前提下, 按最大的粒度(如文件)获得锁, 以便与(分布在别处的)锁管理器的交互最少; 不过为了避免过度的资源(数据)竞争, 如果随后有冲突发生的话, 则允许让已获得的锁再衰减为更细的粒度(如页面或对象).

(3) 回调技术: 并发控制(封锁)即是要强制多个并发事务的可串行化, 复制管理是对因跨事务缓冲而导致的相同数据项在不同的客户缓冲区中的多个副本之间的一致性的管理. 客户缓冲区的一致性管理不仅要求并发控制还要求对复制进行管理, 并发控制的手段是封锁, 复制管理的一种有效方法是回调. 此二者结合起来形成的所谓“回调式封锁”(callback locking)策略已被证明是维护客户缓冲区一致性的一种非常有效的策略.

回调式封锁算法包括两个方面: 回调读 CB_Read(CallBack-Read)和回调写 CB_Write. 回调读的过程如下: (假如采用页级锁粒度)当客户需要访问的页面不在其缓冲区时, 便向服务器发出对该页面的请求. 若其它客户均未持有对该页面的写锁, 服务器便向请求者返回该页的一份拷贝; 否则服务器就需等待所有这些锁都被释放后才能将所申请的页面发往申请该页的客户. 回调写过程的思想与回调读过程的思想大体相似.

3 OODB 页服务器体系结构的 3 种变体

在概念级, 一个数据库由一集对象组成. 在物理级, 一个数据库可被划分为一集有固定长度的页面. 页面是在磁盘和主存之间数据传输的最小粒度. 对象是一个逻辑单位, 其大小与物理页的尺寸无关. 较大的对象一个就能跨越多个页面, 较小的对象可以多个共居一页. 普通对象的尺寸通常都不超过一页, 只有特殊对象(如 BLOB)需要跨越多个页面. 为了讨论方便起见, 下面不妨假定只涉及普通对象.

OODBMS 在许多方面都存在数据粒度的选择问题, 正是这些不同的粒度选择构成了 OODBMS 不同的 Client/Server 体系结构. 下面介绍 OODB 的 3 种重要的系统功能所涉及到的数据粒度, 这 3 种粒度选择与前面的 3 种关键技术有密切关系, 且 3 者是彼此正交的.

(1) 客户/服务器间数据传输的粒度——粗线条地决定了客户/服务器体系结构的种

类。OODB 通常选择的传输粒度为对象或页面,从而形成了所谓的对象服务器和页服务器。

(2) 并发控制(封锁粒度)——为支持多客户(即多用户)对数据的访问,DBMS 需要保证事务可串行化。大粒度封锁(极端情形为把整个库锁起来——退化为单用户系统)的特点是低开销、高冲突。

(3) 复制管理(回调)——跨事务缓冲允许同一份数据在多个客户缓冲区中都持有数据的一个副本,缓冲一致性的维护要求对复制的管理与并发控制相配合。复制管理同样有一个粒度的选择问题。

由于其显著的性能优势,页服务体系结构不仅是被当前的 OODB 原型和产品采用最多的一种 Client/Server 体系结构,也是被研究最深入的一种 Client/Server 结构。在前面介绍的三维粒度选择中,把第一维(即传输粒度)固定选为页面,根据第二、三维的不同组合就可形成页服务体系结构的一系列变体。下面介绍 3 种有重要意义的变体。

为了叙述方便,下面把页服务器简写为 PS,而用 PS-xy 表示 PS 的变体,其中 x 代表并发控制的粒度,y 代表复制管理的粒度。x 与 y 均可取“O”或“A”,分别表示静态对象粒度和自适应(动态决定)粒度。

3.1 对象封锁/对象回调(PS-OO)

PS-OO 方案试图把页面传输具有的通讯优势与对象的封锁和回调所允许的高并发度结合起来。在 PS-OO 中,若对象已在缓冲区中,则客户可不受服务器的干预直接去读这些对象。若所要求的对象不在本地缓冲区中,则由客户决定对象所驻的页面,并向服务器请求这个页面。服务器可在必要的时候从磁盘读取该页面并要保证其它客户均未对这个对象施加写锁。在把这个页面发送到请求该页面的客户之前,服务器还要把该页中已经被其它客户加了写锁的对象打上“不可用”标记,然后才将该页发送到请求该页的客户机上。

当客户希望修改某个对象时,便向服务器请求该对象的写锁。服务器随后所做的处理就如同基本对象服务器的处理一样,即向持有该对象副本的各远程客户发出对象级的回调请求。只不过在 PS-OO 中,客户对回调请求采取的响应不是把要调回的对象全都清除出缓冲区,而只是打上“不可用”标记。

3.2 对象封锁/自适应回调(PS-OA)

PS-OA 与 PS-OO 的差别仅在于在对回调的响应上采取了更为灵活的做法。在 PS-OO 中有可能会发生这样的情形:客户 A 读了页面 P,P 便保留在 A 的缓冲区中且不再被 A 使用。此时若有另一个客户 B 希望修改页面 P 中的多个对象,A 就不得不对服务器因 B 而发出的多次回调请求进行响应。PS-OA 正是为了避免 PS-OO 在这种情况下因多次通信造成的低效性。

在 PS-OA 中,当服务器收到了对给定页面上某个对象的写锁请求后,便向持有该页的副本的所有其它客户发出回调请求。收到回调请求的客户便要检查该页中是否有任何对象正被本地某个事务封锁着。若是,回调响应与 PS-OO 中的相同,即只允许把特定的对象回调回去。否则,若该页中已没有任何对象正被本地事务所使用(加锁),便把整个页面从缓冲区中清除,这时的回调响应又与基本 PS 中的回调响应类似。

3.3 自适应封锁/自适应回调(PS-AA)

PS-AA 在封锁和回调上都试图来采用自适应方式。在没有数据冲突的情况下,PS-

AA 与基本 PS 是一样的,而在发生冲突时只把对发生了冲突的数据的操作衰减为更细的粒度.当引起衰减的数据冲突消退后,PS-AA 还具有再次恢复大粒度操作的能力.

在 PS-AA 中,客户要在对象和页面两种粒度上记载本地读/写操作.而服务器仍同在 PS-OA 中一样,只在页的粒度上跟踪被各客户缓冲的数据副本.

当客户希望读一个不在其缓冲区中(或标记为“不可用”)的对象时,便向服务器发出读申请.服务器收到读对象申请后,即检查(页和/或对对象级的)冲突情况:(1)无冲突——请求可以满足.(2)对象级冲突——请求将被阻塞直至其它客户将写锁释放为止.(3)页级冲突——先将锁粒度衰减为对象级,然后再按前面的(1)或(2)去处理.

当客户对它希望修改的对象不具备写权限(即未获得对该对象或其所在页面的写锁)时,便对服务器发出写请求.服务器起初对写对象请求的处理就如在 PS-OA 中的一样——若无对象级的写冲突,服务器便向所有持有该页缓冲副本的客户发出回调请求.服务器收到的回调回执只可能是下列两种情况之一:(1)各客户当前都不使用该页面,便把对该页的写权限授予申请者;(2)否则,就只能把仅对该页中特定对象的写权限授予申请者.

4 总 结

本文已对 OODB 的体系结构,特别是 OODB 的页服务器体系结构进行了系统综述.现再把相关文献中所做的定性分析及定量模拟试验得出的一些重要结论做一次简要总结:在(OO)DBMS 的 4 种基本体系结构中,(1)数据库服务器主要是 RDB 实现面向对象扩充时所采用的一种自然选择(渊源于 RDB 的历史背景);(2)对象服务器因语义问题简单,服务器并发控制的实现相对(页服务器)较容易,因而在最早的一批 OODB 原型实现中选用对象服务器的较多.对象服务器的两个主要缺陷是数据传输的效率低,且不能利用超级工作站的剩余处理能力和存储容量;(3)页服务器由于不关心语义问题,网络数据传输效率高;又因为大量的数据库处理在客户方完成(充分利用了工作站的高性能),因而服务器的吞吐率较高.现代的 OODB 产品多数采用了页服务器体系结构.但当对象簇聚(Clustering)程度较低且客户缓冲区不能开得足够大时,页服务器的性能会表现得不如对象服务器(这正是对象服务器的长处).PS 对于顺序浏览查询的性能是较高的;(4)文件服务器利用 NFS 读数据库页面的性能是非常好的,但 NFS 的写操作之慢也是出了名的,较大的客户缓冲区对于提高文件服务器的修改查询的性能会有一定的帮助.

在这 4 种基本服务器类型中,文件服务器处在一个极端,它把绝大部分的工作交由客户方完成,因而从理论上讲,文件服务器体系结构应提供最大的吞吐率.但一般认为文件服务器的综合优势不及对象服务器和 PS,对象服务器又不及 PS.成组对象服务器(Grouped-Object Server)^[7]可认为是针对对象服务器数据传输性能欠佳的一种补偿(文献[6]提及的 bundling 机制其实也就是想提供成组对象服务),目前还未见到对成组对象服务器更深入的研究文献.

一般认为 PS 明显地优于对象服务器^[5,7],因而在现今的 OODB 产品中采用 PS 结构或类似 PS 结构者居多.而在 PS 的多个变体中,具有自适应能力的 PS-AA 被证明能够提供最大的性能,它比纯对象服务器、纯 PS 以及 PS 的其它一些变体均有明显的性能优势^[7].

文献[6]是较早论述 OODB 体系结构的一篇文章,其中详细论述并通过实验模拟,分析

比较了3种基本的OODB体系结构,即对象服务器、页服务器和文件服务器.文献[5]则侧重于从DBMS功能在客户方和服务器方之间的分配的不同,研究了3种不同的DBMS体系结构:CS,PU和EWS,并且没有把DBMS限定为OODBMS,所做的模拟实验是基于RDM的.文献[5]中的这3种结构可以大致映射为文献[2]中所讨论的3种结构:数据库服务器、对象服务器和页服务器.文献[2]是站在Client/Server计算模型的角度定性地论述了数据库服务器、对象服务器和页服务器这3种结构.比较系统且全面地论述对象服务器、页服务器特别是页服务器的3个变体(PS-OO,PS-OA和PS-AA)的当推文献[7],文献[7]还就不同的系统模型和负载(Workload)模型对对象服务器、页服务器及页服务器的3种变体的性能进行了模拟,模拟结果(尽管无法穷举各种可能)表明PS-AA是最好的.

参考文献

- 1 Anderson Bruce, shaw Mary *et al.* Software architecture: the next step for object technology (PANEL). In: OOPSLA'93, 1993. 356-359.
- 2 Loomis Mary E S. OODBMS' client/server architecture. JOOP, 1992,4(2):40-44.
- 3 Franklin M J, Carey M J *et al.* Local disk caching for client-server database systems. In: VLDB'93, 1993. 641-654.
- 4 Worden Daniel. SYBASE developer's guide. SAMS Publishing, 1994.
- 5 Roussopoulos N, Delis Alexis. Modern client-server DBMS architectures. SIGMOD RECORD, 1991,20(3):52-61.
- 6 DeWitt D, Maier D *et al.* A study of three alternative workstation-server architectures for object-oriented database system. In: VLDB'90, 1990. 107-121.
- 7 Carey M J, Franklin M J *et al.* Fine-grained sharing in a page server OODBMS. SIGMOD RECORD, 1994,23(2):359-370.
- 8 Che Dunren, Mai Zhongfan. The overall design of MIDS/BUAA: a multimedia intelligent database system. In: ICYCS'93, 1993. 5. 67-5. 70.
- 9 车敦仁, 麦中凡. MIDS/BUAA 存储管理子系统的设计. 软件学报, 1995,6(6):372-378.
- 10 Joshi A. Adaptive locking strategies in a multi-node data sharing system. In: VLDB'91, 1991. 181-192.
- 11 Lambetal C. The objectstore database system. CACM, 1991,34(10):50-63.

THE ARCHITECTURE OF OBJECT-ORIENTED DATABASE SYSTEMS

Che Dunren Zhou Lizhu Wang Lingchi

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Abstract Client/Server computing has very strong impact on the architectures of present software systems. Especially in the database area, nearly all object-oriented databases (OODBs) adopt this architecture. This paper surveys the client/server architectures of OODB systems, and in particular focuses on OODB's page-server architecture, including it's three important variants, and their efficiency improvement strategies as well.

Key words Client/server, software architecture, object-oriented database, page-server.