维普资讯 http://www.cqvip.com

数据库系统 面向对象 Windows32 事务处理

计算机科学 1999Vol. 26№. 2

一个基于 Win32 平台的面向对象数据库系统*)

An Object Oriented Database System Based on Win32 Platform

30-34

王意洁 钟 武 章文嵩 胡守仁

TP311.13

(国防科技大学计算机系 长沙 410073)

Abstract KDOODB is an object oriented database system based on Win32 platform. In this paper, storage administration, query processing and transaction processing of KDOODB are explained in detail. The application show that KDOODB is an advanced practical object oriented database system.

Keywords Object-oriented database, Storage administration, Query processing Transaction processing

针对我国信息产业快速发展对国产面向对象数据库系统的迫切需要,我们从应用需求出发,采用在面向对象程序设计环境(C++)中加入数据库功能的方法,参照 ODMG—93 国际标准[1],研制开发了具有新型开放结构的、能支持面向对象程序设计语言和支持数据一致性的面向对象数据库系统 KDOODB,重点研究实现了合理有效的存储策略、实用的查询优化算法[2]和循环查询处理策略[3]以及灵活有效的事务管理机制。基于 KDOODB 系统,我们开发了城市道路路面技术数据管理系统和项目合同管理系统,在实际应用中,应用系统得到了用户的一致好评。

1 KDOODB 的总体结构

KDOODB 的总体结构如图 1 所示,它由服务器端和客户端两大部分构成。KDOODB 数据库是由一系列体(volume)组成,体被等长度分割成若干页,对象被存储在各页中;若是大对象,则可跨页存储。客户端一个应用程序可有多个事务操作,但一次只能执行一个事务。客户端有一个缓冲区来缓存从服务器获得页面,而服务器主要负责页面级的并发控制、日志维护和页面存取等工作。

KDOODB 参照 ODMG-93 标准对 C++ 文法进行了扩充,并合理有效地实现了 C++ 对象的持久化、对象的引用以及对象之间的关系。利用 KDOODB 进行数据库应用开发的基本步骤是:(1)建立数据库阶段,即利用对象描述语言 ODL 描述数据库模式(类描述);(2)处理数据库阶段,即利用 C++ 风格的 API 和高层语言实现对数据库进行各种处理。KDOODB 的 ODL 是 C++ 的扩充,我们利用自行设计的 ODL 预编译器对其进行预处理,从而获取数据库的模式。KDOODB 为用户提供了三类实现处理数据库的手段。第一类是支持应用开发者利用 C++ 风格的 API,灵活地建立新的数据库应用;第二类手段是支持应用开发者综合利用已扩充的 C++(如:对象查询语言 OQL)高效地开发应用;第三类手段是支持用户使用图形用户界面方便自如地进行交互式操作。

KDOODB 页服务器结构的实现如图 2 所示,它采用了多进程多线程技术,每个客户机上有一个客户进程,KDOODB 的核心在服务器上,服务器上有一个服务进程,服务进程由一个服务管理线程和多个服务线程组成,服务线程的数目对应于客户进程的数目,服务线程由服务管理线程控制。

*)本文得到九五国防科技预研项目的资助。
王意洁 博士生,研究领域为面向对象数据库和并行分布处理技术;钟武,获博士学位,研究领域为数据库及分布处理技术;章文嵩 博士生,研究领域为面向对象数据库;胡守仁 教授,博士生导师,研究领域为面向对象数据库、虚拟现实、高性能机体系结构及并行分布处理技术。

维普资讯 http://www.cqvip.com

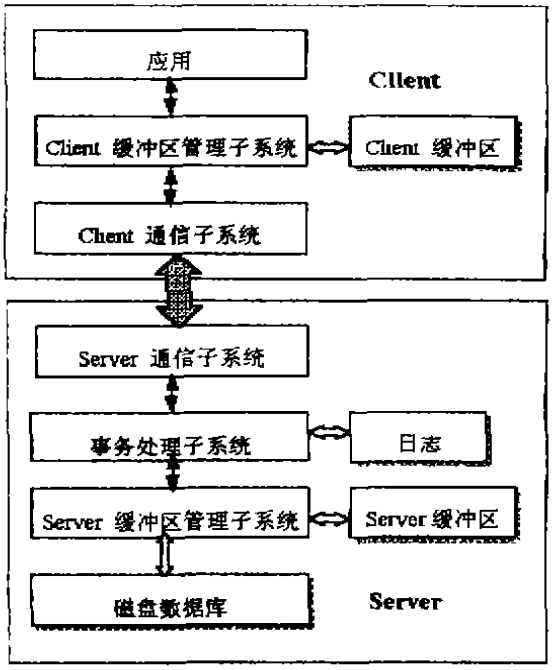

图 1 KDOODB 的总体结构

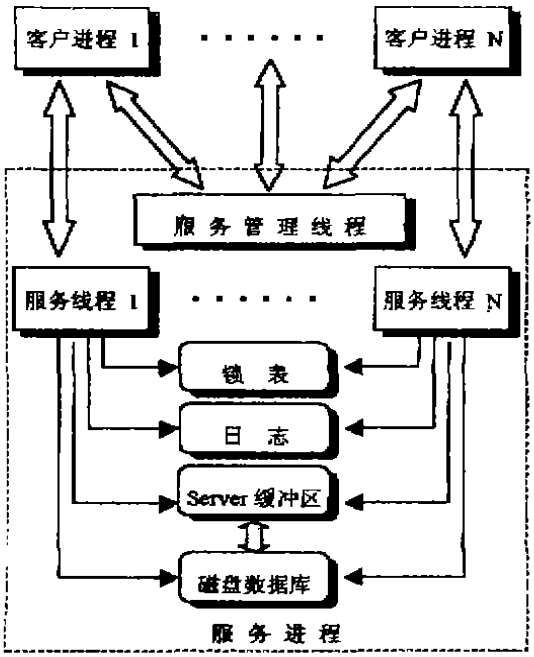

图 2 KDOODB 的页服务器结构

2 存储管理

KDOODB 存储子系统有效管理磁盘数据库，客户缓冲区和服务器缓冲区。

2.1 利用虚存映射机制实现大对象访问

在 KDOODB 中，我们利用现代操作系统的虚存映射机制进行内存缓冲区的管理。在访问大对象时，我们没必要一次把对象全部装入内存进行访问。我们为大对象在数据库的缓冲区中分配其所有空间，注意大对象分配到的空间都是虚的，它们没有与物理内存对应；应用程序对它的任何读写操作，都会引起内存访问异常，此时 KDOODB 的异常处理程序获得访问异常的内存地址，将虚页提交分配物理内存，从服务器调入相应的页装入内存并返回。应用程序的读写操作可以继续进行，直到对象的另一个页面被访问，再按以上过程进行，这样做的好处是按需调页，降低首次等待时间。当应用只对大对象的部分内容访问时，这样处理很有效。

2.2 利用灵活指针实现对象间的引用

面向对象数据库中最基本的操作是对象引用，通过一个指针找到目标对象。因此数据库系统面临的问题是指针在内存中，而所指的目标对象还在数据库文件中，一旦该指针引用目标对象时，如何尽快地把目标对象调到内存中？这可以利用虚存机制实现。但在客户新调入一页时，可能遇到指针引用问题，而指针倒换（Pointer Swizzling）方法依赖于具体的 C++ 编译器，故其很难实现为任意 C++ 提供持久性的对象数据库。在 C++ 中允许创建灵活指针（smart pointer）类，它包含指针数据成员，并通过重载指针的运算符在指针操作上增加新的功能。再利用模板类就能生成任意类型的灵活指针类。因此，我们可以用灵活指针方法解决永久对象引用问题，每个永久对象有一个逻辑上唯一的对象标识 OID，当对象引用时，利用 OID 查表（一般是 Hash 表）找其内存地址，若在则返回地址，否则从数据库中装入该对象，加入表中，返回内存地址。对于相同永久对象进行多次访问，每一次对象引用意味着一个查表操作，这是耗时的，但如果 Hash 表并不庞大，Hash 表中元素分布比较均匀，平均一次多点就能命中，这种操作是完全可以忍受的。使用灵活指针的方法不破坏 C++ 语义的完整性，使用比较灵活，实现也比较简单。

3 查询处理

3.1 查询优化

维普资讯 http://www.cqvip.com

KDOODB 的对象查询语言 OQL 符合ODMG—93 标准，与 SQL 风格一致，采用 select-from-where 的语法结构。针对 OODB 及其查询的特点，我们对 OODB 的查询优化进行了较深入的研究[2]，以此为基础，提出了以下实用有效的优化准则：

准则 1：确定最佳循环嵌套关系，并在此循环嵌套关系上进行循环不变式外提。

所谓确定最佳循环嵌套关系是指：在满足由循环变量间的依赖关系所决定的循环嵌套关系的条件下，确定一种循环嵌套关系 G，使得：对任何一种可能的循环嵌套关系 H，通过在 H 上对原子谓词公式进行不变式外提的变换后，程式的计算代价不会低于在 G 上进行不变式外提变换后的计算代价。

准则 2：对最靠循环外层的原子谓词公式尽早做出判断，化简谓词公式。

准则 3：当最靠循环外层的原子谓词公式不止一个时，计算代价最小的原子谓词公式首先得到计算和判断。

准则 4：反复利用准则 1、2、3 进行变换，消除外提的无用的原子谓词公式，直至不能再降低计算代价。

3.2 循环查询

循环查询是一种既特殊又复杂的查询。所谓循环查询，是指其查询图中存在环路(cycle)的查询。针对 OODB 及其循环查询的特点，我们依据"分而治之"的原则对循环查询进行处理，具体的处理策略[3]如下：

· 识别循环成分。识别查询中的循环成分，也就是识别查询图中的环路。根据查询图中的各种环路的具体特点，充分利用图论的有关算法进行循环成分的识别。

· 处理循环成分。对各循环成分分别进行处理是"分而治之"原则的具体体现。在查询对象图中，与查询图中的某个环路相匹配的对象和对象之间的关系也构成一个环路，我们称其为对象环路。对循环成分进行处理的目的就是要找出查询对象图中的所有对象环路。

对于查询图中的自圈，我们可以采用传统的正向遍历与嵌套循环检索相结合的方法[4]。这种方法可以在多项式时间内得到处理结果。

通过对查询图中的各种双连通成分进行分析研究，并结合有关的图论知识，我们提出了具有一定普遍性的双连通成分处理算法，算法利用标记传递机制在多项式时间内找出所有双连通成分对象环路，算法的具体描述及其正确性证明参见文献[3]。

· 循环查询的处理。我们将循环查询中的循环成分视为一个特殊的类，该类的特性由循环成分中各类的特性共同组成，该特殊类的对象即为与循环成分相匹配的对象环路，利用前面的处理算法可以得到特殊类的对象，这样，循环查询就转化为非循环查询。然后，利用非循环查询的有关算法(包括优化算法和执行算法)对转化后的查询进行处理以得到最终的查询结果。

与现有的循环查询处理方法[4]相比，我们提出的循环查询处理策略具有以下特点：(1)它适用于多种循环查询，而且对查询的目标类数目没有限制，具有一定的普遍性；(2)它能在多项式时间内得到查询结果，具有一定的实用性。

4 事务管理

参照 ODMG—93 标准，KDOODB 支持一种复杂度较高、灵活性较大、应用领域较广的事务模型——嵌套事务模型，采用经典的两段锁协议 2PL 进行并发控制。

4.1 事务标识分配策略

事务标识的分配是影响嵌套事务执行效率的一个重要因素。我们从事务处理对事务标识的需求入手，结合嵌套事务模型的具体特点，提出了一个实用有效的事务标识分配策略——基于位的事务标识分配策略，它将事务层次结构的有关信息有效地记录于事务标识中，与传统的基于树和哈希表的策略相比，它兼顾了存储空间和处理机时间的有效利用，提高了嵌套事务的运行效率。

在基于位的事务标识分配策略中，子事务的标识包含了其父事务的标识，一个事务标识由若干个元素构成，每个元素与事务层次结构中的一个层次相对应。对于顶层事务而言，事务标识由一个元素构成，随着事务层次的逐渐降低，表示事务标识的元素个数逐渐增加。每当一个新的子事务产生，它的事务标识由其父事务标识和一个用于与兄弟事务区分的新元素构成。许多已有的事务标识分配策略对存储空间的使用是以字节为单位来考虑的，导致了存储空间的浪费。基于位的事务标识分配策略对存储空间的使用是以位为单位来考虑的，从而节约存储空间。

在基于位的事务标识分配策略中，一个事务标识由若干个元素构成，每个元素与事务层次结构中的一个层次相对应。每个元素由一个基于位的编码

维普资讯 http://www.cqvip.com

序列表示，编码序列由编码单元组成，各编码单元的值的总和即为元素的值。编码单元的基于位的长度是预先定义的，所以一个编码单元能够表示的数值范围也是预先定义的。若元素的值超过一个编码单元的表示上限（设 n 是编码单元的长度，即：2^n-1）时，将该编码单元置为全满标志（即：全零），并分配一个新的编码单元，如果元素的值又超过了两个编码单元的共同表示上限（即：$2\cdot(2^n-1)$）时，则再分配一个新的编码单元，依此类推，直到满足要求为止；事务标识的第一个编码序列用于表示与元素相对应的编码单元的数目。

4.2 基于页的恢复策略

针对页服务器和嵌套事务模型的特点，我们为 KDOODB 系统设计了一种基于页的恢复策略 WAL_P。WAL_P 以先写日志协议为基础，以页作为恢复粒度，能够有效地处理各种故障（即：事务故障、系统故障和介质故障）。它具有一定的灵活性，对日志类型（操作日志和值日志）和缓存管理策略都无严格的限制。它在嵌套事务的正常执行过程中进行了周密细致的准备工作，从而使嵌套事务的恢复准确有效。它较充分地考虑了恢复的效率问题，不仅提供对保存点和检查点的支持，而且在各阶段的处理工作都体现了"效率至上"的准则。另外，它有较好的可扩展性，能很容易地将其改为以对象为粒度的恢复策略，从而为基于对象的事务管理提供支持。

4.2.1 基于页的日志序号　基于页的日志序号是指：每个页有一个唯一的日志序号，这个日志序号将页的状态改变与日志记录联系起来。

4.2.2 数据结构

· 日志记录。日志文件由一系列记录构成，日志记录包括以下几个域：**LSN**：日志序号；**Tid**：事务标识；**Pid**：页标识；**Length**：记录的长度；**Data**：redo 和 undo 操作所需的数据；**Type**：记录的类型；**Subtype**：记录的子类型；**Last_LSN**：指向属于同一事务的前一个记录；**RedoNext_LSN**：在系统重启时被动态设置，它指向与下一个 redo 操作相对应的记录；它出现在 CLR 类型的记录中时，指向被补偿的记录；**UndoNext LSN**：只在 CLR 类型的记录中出现，指向事务卷回时需处理的下一个记录。

· 事务表。是事务管理的基础，在事务执行过程中，事务表是被动态更新的。在设置检查点时，事务表将被写入日志文件，等到系统重启时用于确定圆满事务和夭折事务。事务表包括以下几项：**Tid**：事务标识；**Last_LSN**：记录事务的最后一个日志记录的序号；**State**：事务的当前状态，active、pre-commit。

· 脏页表，管理已在缓存中被更新但尚未存入磁盘数据库的页（即：脏页）。每当某页被存入磁盘数据库时，就从脏页表中删除与其对应的项。在设置检查点时，脏页表将被写入日志文件。脏页表包括以下两项：**Pid**：页标识；**Dirty_LSN**：记录页被调入缓存后首次被更新的逻辑时间，即：与第一次更新对应的日志记录序号。

4.2.3 系统重启　当发生系统故障时，需要进行系统重启以使数据库恢复到一致状态，从而保证事务的原子性和持久性。基于 WAL_P 的系统重启可以分为三个阶段（如图 3）。

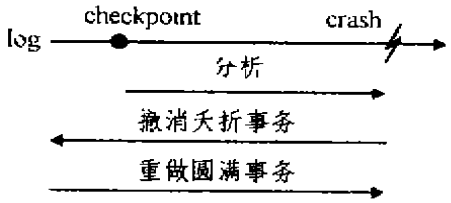

图 3　基于 WAL_P 的系统重启

· 分析阶段。是为下面两个阶段作准备的，它从磁盘的指定位置取出最后一个检查点的 LSN，从该检查点取出信息来初始化事务表和脏页表；然后，从该检查点开始依次读取每个日志记录并进行处理，直到日志文件的结尾。分析阶段结束时已能确定所有圆满事务和夭折事务。它为下面两个阶段提供的信息有：事务表、脏页表以及最后一个写入日志文件的记录 Last_record。

· undo 阶段。任务主要是：1）撤销夭折事务；2）找出需要重做的更新操作，为 redo 阶段作准备。从 Last_record 开始自后向前依次读取每个记录，并进行判断和相应的处理：如果该记录对应于夭折事务的更新操作，且更新结果已存入磁盘数据库，则进行 undo 操作，并产生一个 CLR 记录；如果该记录对应于圆满事务的更新操作，且更新结果未存入磁盘数据库，则将其加入前向链，为 redo 阶段作准备；另外，如果是 Start 记录，则从事务表中删除与该事务对应的项。

· redo 阶段。依据前两个阶段的结果选择性地进行某些 redo 操作，从而保证事务的持久性，如前所述，它的准备工作已由 undo 阶段完成（建立前向链），它只需获取前向链的头，然后依次重做前向链中的每一个操作。由于前向链是存于日志缓冲区中的，所以 redo 阶段不必访问日志文件，从而节约了

维普资讯 http://www.cqvip.com

大量的时间开销。

结论 从理论和技术两方面来看，KDOODB 系统的主要特点表现在：

(1)系统采用了合理有效的页服务器结构，并采用面向对象设计方法，从而使系统具有良好的开放性结构，便于今后对系统的维护和改进以及与其它应用软件的集成；

(2)参照 ODMG—93 国际标准，系统提供了灵活有效、语义表达能力强的标准化的数据模型；

(3)系统扩充当前国际上流行的面向对象编程语言 C++ 作为数据库应用编程语言，提供定义和操作永久对象的功能，使其与 OODBMS 无缝集成，有效地避免了"阻抗失配"问题；

(4)系统充分利用现代硬件和软件的处理优势，提供了对永久对象的高效存储，支持对永久对象的灵活快捷的操作，尤其是对大对象和复杂对象的快速存取提供了良好的支持；

(5)系统提供了使用方便的非过程化的查询语言，既可以嵌入编程语言使用，又可以作为交互式命令使用，实用有效的查询优化算法和循环查询处理策略为查询语言的实现提供了有力支持；

(6)系统针对页服务器结构和嵌套事务模型的特点，研究并实现了基于页的恢复策略和用于事务标识分配的基于位的事务标识分配策略，从而实现了有效的事务管理，为用户开发新型的数据库应用提供有力支持；

(7)系统为用户提供多种使用方式和友好的用户界面，基于 KDOODB 开发了城市道路路面技术数据管理系统和项目合同管理系统等应用系统，显示出该系统具有广阔的应用前景。

参 考 文 献

1 Cattell R, et al. The Object Database Standard: ODMG—93. Morgan Kaufmann, 1994

2 钟武，胡守仁. OQL 逻辑优化准则. 计算机科学，1998，25(2)

3 王意洁，胡守仁 面向对象数据库中循环查询处理技术的研究. 计算机研究与发展，1998，35(12)

4 Kim, et al. Cyclic Query Processing in Object-Oriented Databases. In: Proc. of 5th Intl. Conf. on Data Engineering. 1989. 564～571

(上接第 43 页)

采取自下而上的方式，仍不失是一种务实和可行的办法。

(5)关于 DW 的刷新与归档问题，每时每刻 DW 都需要保存它的信息长河的一个合适的"区段"以维持其可用性。为此就需要不断地刷新和归档。如何确定刷新与归档的时机是 DW 使用和维护中心必然要遇到的问题，时机取决于数据的可用程度。但是，数据的可用性不是一种可预测的事，只能事后处理。真正构成挑战的是在某些使用场合不允许事后处理，如何以事前处理方式解决数据的刷新和归档，仍是一个需要认真研究和解决的问题。

结束语 目前，数据仓库在各个领域已引起了很大的注意。随着信息量的增加，这项技术将会有更为广阔的前景，并给人类带来不可估量的益处。例如，在市场分析、决策支持、金融预测和医学诊断等各个方面，都可以大大地提高信息处理的效率，更有效地发挥数据库的潜在价值。但正如本文所述，开发一个灵活、可伸缩、高效的数据仓库系统，还需要进一步的研究。

参 考 文 献

1 Inmon W H. Building the Data Warehouse(Second Edition). Wiley Computer Publishing, John Wiley & Sons Inc.

2 Hammergren T. Data Warehousing: Building the Corporate Knowledge Base. Ventana Communications Group Inc.

3 Bischoff J. Achieving Warehouse Success. Database Programming & Design, 1996, 7(7)

4 姚卿达，王珊编译. 数据库研究：面向 21 世纪的机遇与成就. 计算机科学，1996，(4)：1～7

5 王珊，罗立. 从数据库到数据仓库. 计算机世界报，1996-7-15

6 周胜，王珊. 论数据仓库系统中工具的重要性. 计算机世界报，1996-7-15

7 姚卿达，黄晓春，刘向民. 数据仓库和数据采掘应用研究. 计算机科学，1996，23(6)：63～65

8 于戈，等 数据仓库管理中的若干关键技术. 计算机科学，1997，24(2)：31～34

9 李云峰，徐新国 OLAP 及其实现. 计算机世界报，1998-4-6