

面向对象 关系数据库 数据库系统 (21)  
OODB

90-93

# 对象关系数据库系统与技术

Object Relational Data Base Systems and Technology

阳国贵 吴泉源 TP311.13

(国防科技大学计算机系 长沙 410073)

**摘要** Based on the discussion of the influence of OO technology on the data base and the analysis of the problem of current OODBMS, the ORDB(Object Relational Data Base) systems and technology are studied.

**关键词** ORDB, Parallel Processing, OODB

## 一、引言

数据库早已从实验室走出,进入了社会生产力市场,成为众所周知的技术之一,仅关系数据库市场每年的销售额就高达80亿美元<sup>[1]</sup>,在[7]中,它还预言,到2005年,对象-关系数据库的市场可望比关系数据库市场规模大50%。

面向对象数据库一词产生于八十年代初,经过十年多的研究和发展,取得了丰硕的研究成果,但其中也存在不少问题和分歧。近年来,随着对象关系数据库技术的发展,特别是商业产品在此方面的努力,致使人们看好对象关系数据库技术的未来。为使读者了解OO(面向对象)概念对数据库的影响及面向对象数据库技术与对象关系数据库技术间的一些渊源,在背景一节中从应用需求与OO对数据库技术的影响、已有OODB系统中存在的一些主要问题方面进行了介绍,作为了解对象关系数据库技术发展的基础。

### 1. 应用需求与OO对数据库技术的影响

早在10多年前,一些新的应用领域包括CAD/CAM、CIM(计算机集成制造)、CASE、OIS(办公信息系统)、GIS(地理信息系统)等就对数据库技术提出了许多新的要求,如工程对象的构成关系较复杂,要求存储和管理复杂对象;这些领域中的数据类型较多,要求具有用户自定义数据类型的可扩展能力,支持新应用领域中的数据类型,如多媒体、空间数据、科学数据、地理数据等。对这些数据类型的处理较传统数据复杂,典型的如对空间多维乃至高维数据的查找与索引技术,基于图像内容的查找技术。特

别地,还提出了对常规内存的对象管理以及对大量对象的存取和计算;支持长事务和嵌套事务的处理;由于设计过程中的交互修改,还要求对模式的动态修改能力;为克服所谓“阻抗失配”的问题,还提出了程序设计语言与数据库语言无缝集成的要求,等等。

这些广泛而新型的应用领域和特殊要求在整体上推动着数据库技术特别是关系数据库技术的发展,在这种情形下,数据库技术,以至关系数据库技术如何发展成为数据库界关注的最大热点,正在那时,面向对象中的封装、继承、对象标识等概念备受人们的关注,用对象可以自然、直观表达工程领域复杂结构对象,用操作封装来增强数据处理能力。这样,人们开始尝试以面向对象概念为基本出发点来研究和建立数据库系统,导致了在数据库技术中全面引入对象概念的面向对象数据库的诞生。以[1]为代表,刻画了这类系统的主要特征,以OO或OO模型为出发点的面向对象数据库,确实推动了一批批原型系统的研究和开发,如Gemstone、Orion、Iris、Ontos、O2、ObjectStore、Objectivity/DB、Versant、Itasca等等。

与此同时,扩展关系数据模型的工作也得以展开,如以Michael Stonebraker为首的Postgres系统的研究, Won Kim领导的UniSql系统的工作,等等,特别地,以[2]为代表,对第三代系统提出了构想,使第三代能够支持和满足新型应用领域的要求,这个宣言的提出,也得到了关系厂商阵营的支持,它们的基本思想就是要将关系技术与对象技术融为一体,近年来出现的对象关系一词系统全面地体现了这种思想,在[7]中,Stonebraker认为它是数据库技

术发展的下一个大浪潮,人们普遍地看好 ORDB,认为它是数据库技术发展的主流,这是建立在对 OODB 问题的深入分析的基础上的<sup>[4]</sup>。

## 2. 已有 OODB 系统中存在的一些主要问题

首先,面向对象数据库系统的研究和开发经过了十余年之后,在各个技术层面上都取得了许多成就,遗憾的是,目前在面向对象数据库系统的任何方面都远未取得完全一致,许多面向对象数据库产品在程序设计接口、实现方法(手段)、对查询的支持等方面都存在许多差异,尽管 ODMG-93 的颁布到今天已有四年了,但许多厂家并未把它作为一个完整的标准来真心支持它,它被厂家的意愿随意支解,一些厂家支持这部分,而另外的厂家支持那部分,例如,我们可以注意到目前仅 O2 支持了 ODMG 标准中的对象查询部分 OQL。

其次,面向对象的数据库产品在一些方面仍落后于关系数据库产品,例如,还没有 OODB 支持视图机制,并且在这方面的先期研究也未真正展开;模式演化在 OODB 中更为落后,许多 OODB 产品仍然停留在类似 CODASYL 风格的模式/应用编译周期上;OODB 与它们的应用程序语言紧密集成,许多系统仅提供一种语言(大多为 C++);除此之外,OODB 系统的鲁棒性、可伸缩性和容错性都难以赶上关系数据库系统。

在应用开发工具、C/S 计算环境相关的一些方面也存在不少问题。如在工具方面,OODB 明显缺乏最终用户工具及应用开发工具,而这方面的工具在关系数据库中却得到了广泛应用。

最后,OODB 的商业市场的增长速度比预想要慢得多,一些应用领域,如 CAD,原指望很快就会从文件系统转移到 OODB 上来,可现实并非如此;另外值得指出的是,一些大型的数据库厂商并未转移到 OODB 上来,它们仍热衷于关系数据库技术。

[4]也对对象概念对数据库技术带来的各种影响做了较全面的分析,并指出:在面向对象数据库系统领域广泛开展研究工作的同时,扩展的关系数据库系统趋于成熟,其产品可以从几个提供商那里得到,如 CA-Ingres、IBM、Illustra、Unisql 等,事实上,这些系统已采用了 OODB 中在数据模型和查询语言方面的一些长处,这个趋势无疑将继续下去。IBM 的 DB2/CS V2 支持用户定义的基本类型和函数,规则和大对象,并且认为,它所采用的面向用户的 Extender 方式是让 ORDB 技术得到接受的一种好形式。对象-关系数据库(ORDB)系统的道路已由许多

研究者基本铺平,如[3]、[7]。

## 二、对象关系数据库技术

对象-关系系统和面向对象数据库系统在许多方面不同,它从关系模型和它的查询语言 SQL 出发,从对象的特征上讲,现在的产品提供两种类型的对象的支持,即 ADT(抽象数据类型)和 Row Type(或称 Composite Types),容许增加新的数据类型,使 DBMS 管理事实内容,如职员的照片信息。Row 类型是对元组的直接而自然的扩充,使得表中的元组具有类似于对象的特征(如命名类型、函数/方法),除了可以具有类型属性外,“Row Object”还容许含有引用 Reference-Value 属性,另外,还支持多值属性(SETS、BAGS、ARRAYS、LISTS),在 Row Type 之间还支持继承。

Stonebraker 在[7]中,把那些既有复杂数据处理又有查询功能要求的应用称为对象关系数据库应用,开发这样的应用,强烈要求对象关系数据库系统的支持。一个对象关系数据库系统必须具有以下四个基本特征:(1)容许基本数据类型扩充。(2)支持复杂对象的管理,复杂对象指由基本的或用户定义的类型所构成的对象。为支持复杂对象的构造,一般应提供三个类型构造子,即组合(记录)、集合和引用类型构造子。(3)支持结构、数据和函数的继承,包括多重继承。(4)对产生式规则的支持,若规则的一般形式为:ON 事件 DO 动作,则应支持更新—更新、查询—更新、更新—查询和查询—查询四种基本规则类型。

对象-关系数据库模式的最顶层仍是一组命名关系,然而,关系中的对象现在可以和 OODB 系统中所支持的一样丰富,SQL 对对象查询,如路径表达式、类似于方法的函数引用格式,在 FROM 子句中支持嵌套的 SETS 等方面的扩展均已具备,对对象查询的支持,可以从 OODB 那里学到一些有用的经验。

在系统实现方面,许多 ORDB 系统承诺提供用任意(许多种)编程语言实现它们的 ADT 和应用的能力。在支持中间层和桌面应用方面,这些服务器将和高性能、面向对象、具有缓存(Caching)的前端一道工作,以提供一个开发环境,使得同一对象模型可在不同层上描述数据库,支持 SQL 查询和导航式编程。在功能实现方面,方法可在客户端的缓存数据上运行,或在服务器方运行,这取决于在哪边运行更便宜。对于触发器、参照完整性以及其他完整性限制,

如商业规则,可以 SQL 或强制式的面向对象的语言加以描述,而由系统决定在服务器或客户端上运行。

对象-关系数据库系统在实现和研究上的挑战有以下几方面:

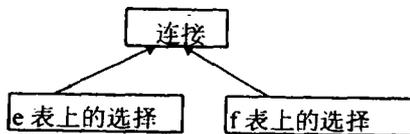
### 1. 对象-关系数据库优化技术

由于对象-关系数据库模型中不同于关系模型的一些特点,关系优化技术需要做许多扩充和改进。如支持类属的 B-树,支持用户定义的比较操作符、选中率函数等;支持基于函数的数据存取方法;对高价函数的特殊优化;用户定义的存取方法;支持属性集合上的索引;对继承层次连接的优化,等等。

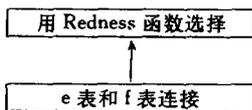
如对高价函数的特殊优化,有如下查询:

```
select e.name,f.name
from emp e,emp f
where e.salary=f.salary
and redness(e.picture)<0.1
and redness(f.picture)<0.1
```

如依旧按关系数据库中的做法,则选择操作下推,以减少连接操作的规模,操作树为:



上述做法对 SQL-92是合式的,但在对象-关系数据库(ORDB)中,由于 redness 函数的执行代价较大,上述策略导致该函数的两次计算,可能就不如以下的查询方案:



### 2. 对象-关系数据库服务器功能的完善

对象关系服务器将继续从今天的关系服务器及早先的对象关系服务器上做进一步发展,SQL 的查询处理将得到进一步加强,并对今天的系统做更多的改进。但是,做为其中的 SQL 语言如何以最好的方式对象化(Objectified SQL)的工作也有待研究和加强,如如何支持视图,对象修改,各种完整性限制,触发器等等;另外,对象-关系数据库还应从 OODB 那里吸收养分和研究成果,如路径索引,对象聚集;在 ADT 方面,一个很重要的问题是对可扩展存取方法的支持,这个话题由来已久,但仍没有找到一个好的解决办法。

在客户端,一个好的映射或捆绑办法及编程接

口是需要的,以用于对对象关系中的对象被 C++, Smalltalk、JAVA 及其他语言所编程;除导航外,还应完全集成对象查询的能力,这方面的问题还将涉及对客户端缓冲(Caching)的查询及数据库的查询(以一种智能的方式进行)。

### 3. 并行处理技术的引入及相关问题

并行关系数据库技术的研究已开展了相当长的时间了,随数据库容量剧增,如超大型数据库的容量目前已达到 TB 量级,查询操作复杂度的提高,如支持在线事务分析处理(OLAP),使得一些关系数据库系统已不得不采用并行处理以提高性能。研究原型如 Gamma、Bubba、EDS,在数据库产品中,Oracle 公司推出的 Oracle7. 2、Informix 公司推出的 Informix8. 0、Sybase 公司推出的 Sybase SQL Server 11等都开始扩充了并行处理能力。对并行关系数据库系统而言,已有工作主要集中在以下一些方面:在并行体系结构方面,对 SMP(对称多处理)、SM(共享存储器)、SN(无共享)以及混合结构进行了分析比较,认为 SN 是支持并行关系数据库的最好结构;在查询优化模型和算法方面,对各种树型结构做了研究;另外,对数据库的物理组织和并行连接算法也做了大量研究。简而言之,关系数据库中的并行来源于数据划分带来的水平并行和操作间的流水并行。

对 ORDB 中的并行处理,在这方面基本上还没做什么工作,原因之一是 ORDB 发展的主要动力在于对它的数据库模型的表达能力和查询处理能力的增强,随着这方面研究的明朗,ORDB 系统的出现和应用,将导致对并行处理能力的要求,因为大型企业中的数据管理需要并行数据库系统,这对 ORDB 的最终成功是至关重要的。关系查询技术可以扩展到对象查询,这对并行查询似乎也应如此。然而,一些 ADT,如多媒体(图像,视频,音频)以及 GIS 数据类型,将包括昂贵的操作,这些操作本身就应当并行处理,否则,装载的不平衡及在这些类型上的查询所需的大量时间将严重影响系统整体性能和限制并行性的发挥。然而,这方面的工作在关系系统中是不曾遇到的,对如何并行化 ADT,怎样从 I/O 及 CPU 方面开展工作都是一个未定论的问题。另外,对 ORDB 中的集属性的处理同样有待研究。

### 4. 基本结论

ORDB 系统将会成熟,并最终提供可扩展(伸缩)的和鲁棒性强的数据库产品,同样,也会提供对于 Row Type 的面向对象的完全支持,把 SQL 在此方面的扩展特性与 SQL 的其他重要特性,包括面向

对象的视图、授权、触发器、限制等功能完整集成。并行处理将得到进一步加强,性能将会有更大的提高。目前,ORDB 中的一个问题是厂商产品的差异仍然巨大,但 SQL3 将会在消除这种差异方面起到积极作用。SQL3 具有 ADT、Row Object 及 Reference 等项内容,可喜的是,现在大的主要的数据库厂商都在朝此方向努力,如 IBM、Informix、Oracle 等。

### 三、ORDB 系统与实现途径

建立一个对象-关系数据库系统,既可以从头开始,如 Illustra、Omniscience 等。也可从已有关系系统或面向对象数据库系统加以发展。

从一个关系引擎过渡到对象-关系引擎大致要完成以下工作<sup>[1]</sup>。

1. 改写分析器,使之成为表驱动的对象-关系分析器。
2. 在改写后的对象-关系分析器中,应包括对计算代价高昂的函数的优化和对继承与复杂对象的支持。
3. 要求原有关系数据库执行引擎具有动态链接用户定义的操作函数的能力。
4. 对原有存取方法接口加以改进,使之能容纳用户实现的新的存取方法。
5. 对索引结构加以改造,如把 B-树变成类属结构。
6. 改写客户端的 API,以便执行更一般的查询,并且获得返回结果。

采用增量式的办法,把一个关系型的 DBMS 引擎变为对象-关系型的引擎,是许多关系厂商的演变策略,第一个采用增量式演变策略的是 Ingres 公司,它于 1988 年将它的关系引擎扩展为具有基本类型扩充功能和规则管理器的系统,这两个特征是对象-关系数据库四个特征中的两个。开始时,提供的功能尽管不够全面,但它较早开始了向对象-关系型系统迈进,1990 年被 ASK 公司购买后,这种演变过程基本上停止了。当 ASK 公司被 CA (Computer Associates) 公司收购后,采用了一种新的策略,即网关方式来得到对象-关系型的功能。IBM 已经采用增量式演变策略作为它们发展对象-关系功能的策略,并将 UNIX 上的关系引擎 DB2/6000 扩展为支持基本类型扩充功能和规则管理器的系统。微软也采用增量式演变策略,改写 Microsoft SQL Server 的某些部分,并开放各种各样的内部接口,它向 SQL 增加复杂对象。

扩展面向对象的数据库管理系统,使之成为一个对象-关系数据库系统,前面讲到,目前大多数的 OODBMS (面向对象数据库管理系统)都与一种特定的语言紧密集成,如与 C++ 或 Smalltalk,它们提供与用户程序相同地址空间的运行系统,ODMG 中的 OQL 有与 SQL3 相同的特征,但在与 SQL3 的关系上,存在两种可能的局面,即 OQL 和 SQL3 相互融合,形成一个单一的标准,或者 OQL 被 SQL3 埋葬,这主要基于推动 SQL3 的那些公司的力量所决定的,这些公司包括 IBM 和 Oracle。这样,OODBMS 有必要向 OQL 系统发展,并完成以下工作:1. 实现一个 SQL 分析器;2. 实现一个 SQL 优化器;3. 实现一个执行查询计划的执行引擎;4. 完成与原有运行存储系统的集成。

市场上的对象-关系数据库系统主要有以下一些: DB2/6000 C/S、Illustra、CA-Ingres、ODB、O-dapter、Omniscience、UniSQL 等。上述系统,尽管在支持对象-关系数据库的全面特征上存在差异,但都是开创 ORDB 市场的先锋,Oracle 8 的出台,进一步表现出 ORDB 发展的强劲势头,随着 Informix、Microsoft、Sybase 等的对象-关系产品的加入,无疑会促进 ORDB 市场份额的快速增长。

### 参考文献

- [1] M. Atkinson et al., The Object-Oriented Database System Manifesto, Proc. 1st DOOD Conf., Kyoto, Japan, 1989
- [2] Committee for Advanced DBMS Function, Third-Generation Database System Manifesto, SIGMOD Record 19(3), July 1990
- [3] H. Darwen and C. Date, Third Manifesto, SIGMOD Record 24(1), March 1995
- [4] Michael J. Carey et al., Of Objects and Databases, A Decade of Turmoil, 22nd VLDB, 1996
- [5] W. Kum, Object-Relational Database Technology, <http://www.unisql.com/tech-spot/tech-spot.html>
- [6] R. Cattell (ed.) (The Object Database Standard: ODMG-93), Morgan Kaufman Publishers, 1994
- [7] M. Stonebraker, (Object-Relational DBMSs: The Next Great Wave), Morgan Kaufmann Publishers, 1994
- [8] Claude Delobel, et al., (Databases: From Relational to Object-Oriented Systems), International Thomson Computer Press, 1995