

新一代对象数据库系统

□郭宜斌

分布式的对象计算要求存储和检索服务,即需要数据库管理系统(DBMS)。满足这种需要可以采用两种技术:关系数据库(RDBMS)技术和对象数据库(ODBMS)技术。但这两种技术都有各自的严重缺陷。第三种选择是对象—关系数据库(ORDBMS)系统。ORDBMS既可以提供RDBMS具有的安全性、完整性、可靠性和SQL功能,又可以提供ODBMS具有的管理复杂数据的能力。

一、未来的数据库技术

大约三年以前,面向对象理论对于数据库管理来说还只是一种热心的推销而已。但现在许多公司已理解对象技术所能产生的好处,并理解了基于对象的存储机制的优势。他们同样清楚RDBMS的长处所在:即大多数RDBMS支持基于事务的客户/服务器应用。

现在,许多用户意识到他们既需要ODBMS技术,同时也需要RDBMS技术。其原因是:他们要做的是建立起关键任务(mission-critical)的应用,因而在数据

库中需要对复杂数据建立模型并进行操作。所谓复杂的数据包括影像、声音、复合文件、动画、数组、地理信息、复杂对象(如由多口油井组成的油田)等。关系型产品显然不擅长处理复杂数据,而面向对象的数据库又不适应多用户、关键任务的环境。另一个原因则来自于日渐流行的面向对象的编程语言(OOPL),如C++或SmallTalk。用这些语言开发出的应用要求有对象存储机制,既能支持数据的对象化,又能提供良好的运行性能、健全的数据管理功能以及对SQL的支持。RDBMS和ODBMS都无法同时满足上述要求。

因此,现在留给许多用户的问题已不再是关系技术与对象技术是否应该融合在一起,而是怎样实现这种融合。

幸运的是,市场上目前已有几种这样的产品,可把这两种技术结合起来,一般将其称为对象—关系数据库系统(ORDBMS)。这样的产品有:UniSQL公司的UniSQL、Illustra Information Technologies公司的Illustra、HP公司的Oadapter等。这些产品使用户在继承

式环境下作为终端查询语言使用,该语言能嵌入C及C++中,O2还具有较高的存取效率,提供Client-Server结构。O2另一个重要特色是它提供了包括图形界面、菜单界面及第四代语言等多种开发工具,为使用、开发O2提供了很大的方便。O2适合于熟悉关系数据库及SQL者的使用,它在欧洲较为流行,它的使用面也较为广泛,它的销售量在面向对象数据库产品中占13%,在欧洲占37%。O2近年来致力在我国的发展,1993年在我国举行产品发布会,其香港代理公司已将产品打入我国,预计近年来会有重大进展。

3. ONTOS

ONTOS是美国Ontologic公司于1989年推出的产品,它的前身是著名的系统Vbase,它提供了与C++有相同语法语义构造的C++扩充语言,同时它还提供交互式环境下使用的类SQL语言——Object SQL语言。ONTOS是一个功能齐全的面向对象数据库管理系统,它支持复杂对象、对象标识、持久性类定义等面向对象概念,同时它也提供良好的并发机制、嵌入事务、数据恢复技术,此外它具有多Client-多Server

结构。ONTOS具有较好开发与调试工具,包括如ONTOS DB Designer, Classify, Cplus, DBAtool等。ONTOS的不足之处是其执行速度尚不够理想。ONTOS公司产品目前我国已有应用。

4. Open Object DB

open object DB是HP公司在其关系数据库产品ALLBASE上扩充而成的面向对象数据库产品,其前身是著名的Iris,它具有关系数据库的所有功能并有面向对象数据库的主要特点,它提供一种扩充的SQL语言叫ISQL并能嵌入C及C++,它并具有Client-Server结构。由于该系统建立在关系数据库管理系统上,因此其运行速度较慢,其面向对象功能的扩充受到一定限制,但此类产品是目前关系数据库管理系统成熟产品的发展方向,具有重大的潜在应用价值。

以上介绍几种较为流行并有一定代表性的面向对象数据库产品。这些产品在我国均有引进与应用,这表示面向对象数据库的应用在我国已开始起步,预计面向对象数据库的应用将随着计算机事业的推进,近几年来将在我国获得飞跃发展。

现有的关系技术中的技巧和知识的同时,也得到了开发需要复杂数据的关键任务应用的能力。特别是 UniSQL,它可支持分布式异质数据库,并可与其他主要的 RDBMS 集成在一起。这种共存能力使得拥有其他数据库系统的用户可以很方便地移到 ORDBMS 上来。

数据库技术今后的发展方向,将集中在把对象技术和关系技术中的优点集合到一个共同的产品中。尽管 ODBMS 厂商和 RDBMS 厂商仍分别坚持自己的阵营,ORDBMS 厂商却已预见到了未来市场的要求,并已可以做到在一个 DBMS 产品中提供这两者的功能。

在详细分析对象—关系技术之前,我们先来分析一下 RDBMS 和 ODBMS 各自的局限。这将有助于理解对象—关系厂商致力于解决的问题及他们所面临的挑战。最后,我们以一个技术领先的产品——UniSQL 为例,来展示一下现在这种技术所能提供的能力。

二、关系数据库(RDBMS)的局限

关系数据模型及其商业化 DBMS 从产生到被广泛接受用了 10 多年的时间。RDBMS 现在已成为大多数组织中的标准数据库结构,而 SQL 也成为最常用的数据访问语言。

RDBMS 的成功并非侥幸。它被设计用来有效地处理大量涉及简单字符/数字数据的短事务。它可以方便地处理特定(ad hoc)查询和新的数据关系,通过事务管理、完整性限制、备份/恢复等手段对数据完整性进行严格的控制,对多用户并发处理也很重视。这些都是现在开发客户/服务器应用所需的关键特性。

但是越来越多的公司也在开发 RDBMS 现在仍支持得不够的一些应用。通常这些应用要求使用复杂数据,而这些数据无法规整地放入表格格式中。到目前为止,RDBMS 厂商基本上仍忽视复杂数据,并且在支持面向对象的开发方面明显乏力。

虽然 RDBMS 厂商声言可以支持二进制大对象(BLOB)数据类型,但他们对 BLOB 内部结构却知之甚少,并且无法解释其内容。这是因为 RDBMS 基本上是用来处理字符/数字数据类型的,如定长或变长字符、数字、日期和时间等。要使所有 RDBMS 的组成部分(分析程序、优化程序、访问方法等)重新认识一种新的数据类型,则需要分解上百万行的代码。所以 RDBMS 的扩充性很差。

至少在两年以内,RDBMS 厂商不会提供在服务器端对复杂数据应用的解决方案。大部分厂商在等着看 SQL3 标准。据说它会在 SQL 语言中包含面向对象方面的扩充。但即使 SQL3 在明天即可明朗化,要实现

所需的扩充程度仍要对核心做大量重新构造工作。

RDBMS 另一个主要局限是它很难模仿现实世界中商业过程和它们之间的关系。如果我们将实体—关系(E—R)模型(经常用于 RDBMS 中)与面向对象理论作一个比较,就会发现 E—R 模型只包含实体(表)和关系,却没有方法和函数。

将 E—R 模型转化为关系数据模型,会失去数据关系的大部分信息。DBMS 只保留了那些可以用简单的主键/外来键方式表达的关系。这样保留的实际只是—些关系句法,而不是丰富的语义。为了实现全部关系,用户必须对许多表做关联(join),而这种操作却代价极大。

此外,RDBMS 产品在用户需反映大量层次对象时显得力不从心。它要求用户将这种关系挤进简单的二维表关系中。打一个简单的比方:假设你的车棚是一个个的小格子组成的。当你骑车下班回家后,存车时要把车子拆开,将各个零件分别放到相应的格子中。第二天,你要先把零件攒成车子,然后再骑着去上班。将对象数据拆分到关系表中的做法与此是相同的道理,既困难又费时。但最严重的问题还在于运行时的效率。每当应用存储/检索对象时,都必须将对象分解/重建,这个过程极大地降低了性能。

综上所述,尽管 RDBMS 在需要处理数字/字符数据的大多数事务处理应用方面取得了成功,但却无力应付新出现的那一类功能强大且要求较高的应用。而这些应用却正是许多公司赖以产生竞争优势的根本。

三、对象数据库(ODBMS)的局限

ODBMS 出现于 80 年代中期,主要是用来为 C++ 或 SmallTalk 等面向对象语言所建立的对象提供一种永久的存储机制。它们也可用来处理 RDBMS 无法胜任的复杂数据和某些真实世界中的应用。ODBMS 首先是由工程单位用于技术应用方面,如 CAD、GIS 和网络管理应用。但最近以来,ODBMS 已进入商业市场,用于处理复杂数据的应用。

ODBMS 的一大优点是其面向对象理论本身的丰富性。这使得用户可以根据他们的理解去建立商业活动及其关系的模型。其中有一个基本的结构——即对象——可以被定义为各种各样的概念:从商业级对象(顾客、定单、雇员、股票行情、保险应用、专利记录等)到平台概念(如 GUI 窗口或打印机)。关系模型要求用户将真实世界中的数据看做一组二维的关系表。这对大多数用户来说不甚灵活。由于对象理论利用实体(对象)、关系和行为等概念,因此它可以更好地反应实际环境中的复杂性,并且将数据以一种用户所熟悉的方

式反映出来。

但是, ODBMS 由于缺少对多用户的支持和缺少标准查询语言和安全措施, 因此未能如人们原来所预计的那样成长。

首先, ODBMS 主要是针对单用户或工作小组设计的(如 CAD/CAM, CASE 方案及特殊的操作系统)。因为所有处理都是在客户端的编程语言中完成的, ODBMS 对多用户的支持不够理想(只有少数例外)。在 ODBMS 中, 方法(methods)只在客户端执行。这就使得网络因数据流量过大而变得不堪重负。同时要求用户配备具有庞大内存的昂贵的桌上工作站。

其次, ODBMS 通过对象标识的指针来寻找(navigate)唯一的对象, 一般不会支持工业标准的 SQL。其对象间的关系定义仅限于模式(schema)级。然而对于动态地建立满足查询条件(并需计算累计值)的对象小组而言, 查询语言是必须的。与此相对照, RDBMS 则可以在通过值访问数据的应用中取得较高效率。

最后, 登录到 ODBMS 中的用户通常可以访问数据库中的所有数据。而许多公司都希望通过用户、组或域以及数据类别(型)等方面的限制来控制这种访问。很多 ODBMS 由于安全性上的原因迟迟不能进入商业生产环境。

总之, 上述缺陷妨碍了 ODBMS 对商业环境的渗透, 并且在可预见的未来仍会造成妨碍。1994 年 ODBMS 市场约 1.3 亿美元。虽然这个市场每年都在以翻番的速度增长, 但它仍然只是 RDBMS 市场逾 20 亿美元的营业额的零头而已。

四、关系一对对象数据库(ORDBMS)填补了鸿沟

ORDBMS 的厂商吸收了上面两种数据库技术的优点, 并将它们集中到一个产品中。其目的是让用户在前端真正使用到对象模型, 同时在数据库服务器上提供所需的功能——安全性、备份、恢复等——从而有效支持真正的生产应用(见图 1)。

目前还有许多应用是 RDBMS 和 ODBMS 各自能力无法满足的。比如网络管理应用、模拟复杂金融手段的贸易系统、大规模事务的面向对象应用以及使用多媒体数据的复杂文件管理系统等。开发这些应用的用户会发现, ORDBMS 也许是一个有效的解决方案。

当前是 ORDBMS 成功的一个绝好的机会。ORDBMS 的主要特点是:

●**可扩充的数据类型。**ORDBMS 比 RDBMS 的数据类型要丰富得多。

●**可扩充的函数。**ORDBMS 允许用户撰写在

客户端、服务器端或两端都可以执行的函数。

●**扩充的数据访问方法。**ORDBMS 允许用户自定义访问方法, 提高对用户定义的数据类型的访问效率。

●**对数据的寻找(navigation)和 SQL 访问。**ORDBMS 支持对象访问方法和关系访问方法。也就是说, 用户可以通过 SQL 或唯一的对象标识在对象中间作寻找。

●**按内容查询。**ORDBMS 容许用户对复杂数据类型的内容进行查询和操作。例如在一个图片库中, 查询命令可以是“找出类似这所房子的房子的照片”或“找出具有这种性质的草地的照片”。

●**对数据库有多种界面。**ORDBMS 使开发人员及工具可以用多种方式访问服务器, 包括调用级界面、嵌入式 SQL 预编译器、C++ 和 SmallTalk 界面, 以及 ODBC 等。

●**健全的数据库管理功能。**数据库管理功能包括安全性、事务管理、备份/恢复、可用性、分布式数据库能力、数据复制、并行处理等。

●**异质数据访问。**某些 ORDBMS 提供对多种数据来源的异质访问能力。

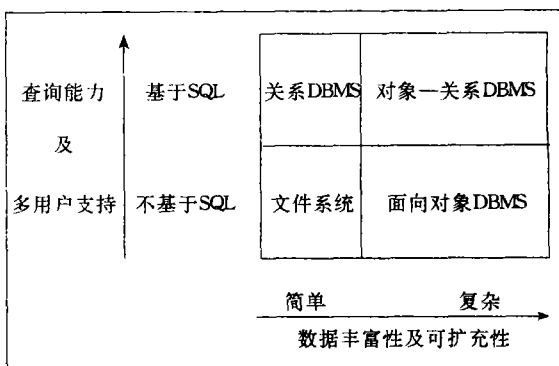


图1 各类数据库比较

ORDBMS 也面临着技术和市场的双重挑战。要获得成功, 则必须使其产品与 RDBMS 产品共存, 从而使用户的技术、技巧和应用方面的投资得到保护。他们还必须提供在服务器端的数据处理功能, 以便同 RDBMS 的预存程序、完整性限制等功能相匹敌。

ORDBMS 厂商还提供可集成到企业级管理系统中去的复杂管理、监控及安全性工具。对 SNMP 和 DCE 的支持也是很重要的。同时还应在服务器上提供对分布式数据的支持, 并在多处理器平台上优化性能。

总之, ORDBMS 满足了当今复杂应用问题所提出的需求。它使得用户有效地对复杂数据建立模型和进行操作, 并可使用面向对象的工具开发应用, 而同时又

不失去关系技术所具备的 SQL 语言、数据安全性、参照完整性和其他能力的优点。

五、UniSQL 代表了 ORDBMS 的趋势

UniSQL 是由韩裔美籍学者 Dr. Won Kim 在 1990 年建立的一家软件公司。Dr. Kim 曾参与 IBM 和其他厂商的 RDBMS 和 ODBMS 的开发工作,在数据库领域有着丰富的经验。UniSQL 公司 1994 年的营业额约为 1500 万美元。由于没有投资返还和债务方面的顾虑,该公司的运行和成长都很顺利。在 UniSQL 创建之初,NTT data Communication Syetem 公司便为其投入数千万美元的“种子资金”。现在该公司在 UniSQL 中只占很少份额,但它得到了 UniSQL 在日本的独家代理权。NTT Data 是日本电报电话公司(NTT)的下属企业,也是日本最大的系统集成商(约 20 亿美元/年)。UniSQL 的得名来自于它结合了(unify)面向对象和关系理论,并且有自己扩充的 SQL 语言(SQL/X)。UniSQL 的设计前提是“对象技术是对关系技术的一种扩充”这个观念。实际上,UniSQL 是一种与关系模型兼容的 ODBMS。其设计完全支持对象管理集团(OMG)所采用的“核心对象模型(Core Object Model)”。

某些公司也曾尝试建立 ORDBMS,但却没有成功。因为它们用现有的关系模型核心再加上一个简单的对象层。其失败的原因要么是对对象模型的支持能力严重不足,要么是完全可以支持但性能很差。因此,通过简单的翻新是无法弥补对象及关系数据库系统之间的巨大差异的。而 UniSQL 的做法却与众不同:它从零开始建立了一个全新的对象—关系数据库核心。

作为对象数据库,UniSQL 支持所有应有的对象特点:用户自定义数据类型和函数、唯一的对象标识、封装、多重继承、多形性和其它特性。UniSQL 的定义及操作语言 SQL/X 则包含了对 SQL 语言在面向对象方面的扩充。

从关系型的角度来看,UniSQL 与 RDBMS 类似,具有视表、并发、安全及访问控制,也支持预存程序和触发器。目前 SQL/X 符合 ANSI SQL92。其函数用 C 写成。用 C++ 写函数的功能也即将实现。UniSQL 目前尚不支持版本控制和长事务,但正在向此方向努力。

UniSQL 可通过其多数据库(multidatabase)特性提供数据集成能力。原来 UniSQL 将主要产品分为两个:UniSQL/X(数据库核心)和 UniSQL/M(多数据库系统)。现在它将这些产品打包为 UniSQL Server。这种重新打包的工作并未影响到此两种产品的实质。UniSQL 的多数据库特性满足了沟通关系—对象阵营

所需的以下几个重要要求:

●**与其它数据库共存。**它提供了对其他主要 SQL 数据库(如 Oracle, Sybase, Informix 和 Ingres)的读/写访问。并可通过 Information Builders 公司的 EDS/SQL Gateway 访问 50 多种 SQL 或非 SQL 的数据来源。

●**数据模型间的透明度。**UniSQL 可以把诸如用户定义类型、方法和继承等面向对象的能力加到本身并不支持这些功能的系统中。例如,UniSQL 可以使 Oracle 看上去像一个面向对象的数据库。

●**跨异质数据来源的单一逻辑视表。**UniSQL 可以在异质数据来源之间提供模式集成、冲突解决、事务上下文和 SQL 透明性(模式变化除外)。以冲突解决为例,它可以不管底层数据库将金额存储为里拉、英镑还是美元,在为应用提供金额时全部折合成美元计算。

●**针对关键任务环境的、健全的分布式数据库功能。**UniSQL 还对 UniSQL 或非 UniSQL 数据库提供了分布式数据库能力。这其中包括地点透明性,透过两阶段提交实现的分布式事务(由底层数据库管理器支持),以及通过分布式查询优化器提供的分布式关联能力。

UniSQL 也提供了一个工具集,称为 UniSQL/4GE。它包括应用开发工具(Object Master)、数据库管理(DBA)工具(Visual Editor)和报表工具(Media Master)。与其他 ODBMS 厂商不同的是,UniSQL 非常重视工具。因为它在准备着与 RDBMS 厂商在未来几年中进行竞争(一旦这些厂商也实现了支持对象数据库管理的结构调整后)。

UniSQL 提供了许多界面,包括 C, C++, SmallTalk, 还有 ODBC 及 UniSQL 自己的扩充了的 SQL 界面(SQL/X)。SQL/X 既可以通过解释器直接访问(ISQL/X),也可通过嵌入在其他语言界面中的方式访问(ESQL/X)。

有充分的理由相信 UniSQL 在商业应用领域会取得成功。但它也面临着一个潜在威胁,这就是,倘若某个大规模的 RDBMS 厂商某天也拿出一个正经的 ORDBMS(即经过重新构造,而非拼凑出来的),那时 UniSQL 也许会发现自已将被挤垮。

当然这种情况至少在近期是不可能出现的。如在 1994 年中,最有可能提供真正的对象扩充功能的 RDBMS 厂商 Oracle,与 UniSQL 签定了一项协议,购买了在石油/天然气行业转销 UniSQL 的权力。很显然,Oracle 公司在解决石油/天然气勘探与开发领域所

要求的复杂数据模型问题中,将得力于 UniSQL 的支持。

UniSQL 将自己的产品定位在具有很多 RDBMS 特点的 ODBMS 上。这使得它隐含地宣布了与 ODBMS 和 RDBMS 两方面的竞争。它所提供的功能要比这两种数据库中的任何一种所能提供的都多。当然,由于 UniSQL 的历史比它的两种竞争者要短,它在某些方面的不成熟也就在所难免。

●**不支持多处理。**UniSQL Server 在一个单一的操作系统进程中运行,利用多线程实现并发控制。尽管 UniSQL 在典型的对象数据库应用中比 RDBMS 要快得多,它的处理结构使它在处理极大量同时用户时不如 RDBMS 那样方便。UniSQL 的处理结构不是产品设计上的问题,应可以在短期内得到解决。

●**市场形象。**许多面向对象的开发人员把 UniSQL 误解为具有对象扩充的或带有对象层的 RDBMS,而事实上 UniSQL 是可以支持关系功能的真

正的 ODBMS。这种误解对 UniSQL 不利,因为它目前希望把面向对象的开发人员争取为它的客户。

UniSQL ORDBMS 有足够的理由既被认为是完全的 RDBMS(全面支持 SQL),同时又被认为是彻底的 ODBMS。对于那些被 RDBMS 在处理复杂数据模型上的局限所困扰,和对 ODBMS 无法提供数据库功能(安全性、访问控制、恢复/备份)感到失望的商业应用市场来说,这一点很有吸引力。在此情况下,由于 UniSQL 保证现有 SQL 代码的轻易移植,已很成功地将自己树立为一种稳妥的解决方案。

对于熟悉关系理论的用户,一旦他们意识到 RDBMS 无法满足其调用复杂数据模型、结构和关系的要求时,他们会首先选择 UniSQL,而不是其他传统的 ODBMS。由于 UniSQL 提倡的是一种“改良”而不是“革命”的方式,因而它极大地迎合了商业市场的保守本性。在此背景下,UniSQL 的快速成长,并跃升为 ODBMS 市场的领导者之一,就是一件很自然的事了。

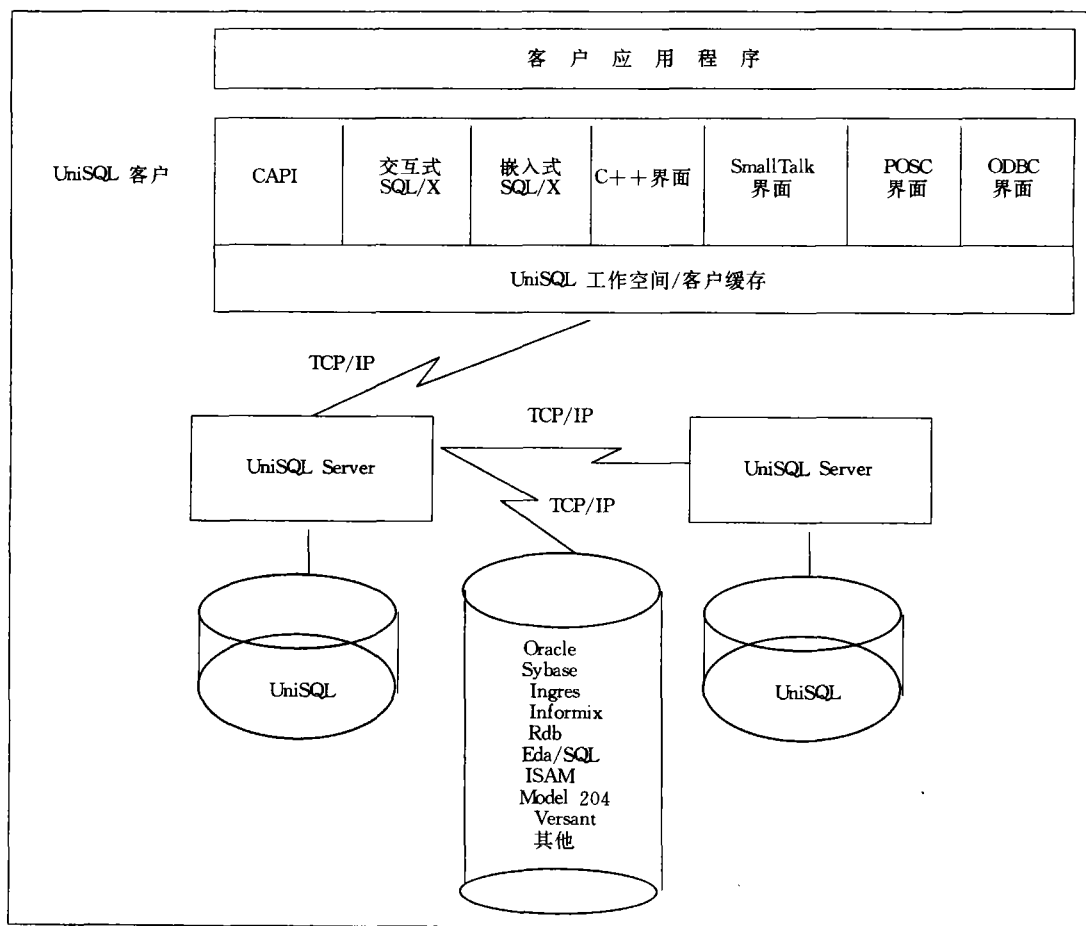


图2 UniSQL 数据库结构