

数据库技术的新进展

——参加第十六届VLDB国际会议总结

王能斌(东南大学) 王 珊(中国人民大学) 周立柱(清华大学)

第十六届超大型数据库国际会议(以下简称VLDB'90)在澳大利亚布列斯班举行。VLDB是重要的数据库国际会议。澳大利亚动员了主要大学和工业界的数据库学者,参加了会议的筹备工作。澳大利亚总督出席了开幕式。会议于8月13~18日进行,共报告了60篇论文,进行了六次辅导报告和三次专题讨论,澳大利亚工业界也在会上做了两次介绍。会议反映了当代数据库的发展水平。

这次会议共收到论文约300篇,经评审录用了其中60篇,录用率为1/5,这在国际会议中算是比较严格的。

自八十年代初、中期以来,从不同的计算机应用领域提出了许多数据库的非传统应用的课题(例如多媒体数据、空间数据、时态数据、复杂对象、知识、超文本(hypertext)等的管理)。为了适应这类应用的需要,提出了不少新的数据模型、系统结构和新的概念,经过几年的实践,逐步形成了面向对象数据库(以下简称OODB)和知识库(以下简称KB)两个生长点。这次会议也明显地体现出这样两个重点。除此以外,还有相当多的论文论述数据库管理系统(以下简称DBMS)的实现技术,这包括传统的和非传统的、集中的和分布的DBMS实现技术。由于硬件的飞速发展,为数据库技术的发展提供了新的物质基础,由此也引出了许多新的研究课题,这个趋势在本届VLDB会议上也体现得十分明显。兹就上述诸方面的情况,分述如下:

1. 知识库和演绎数据库

关于知识库方面的论文只有一篇由法国人C. Esculier发表的“Non-Monotonic Knowledge Evolution in VLKDBS”。这是一篇带综述性的论文。在前半部分,作者提出了非单调知识进化的特征,并简述了在程序

设计语言、数据库以及人工智能中目前使用的一些处理非单调知识进化的方法,论文后半部分提出了“语义容忍”(semantic tolerance)的概念,讨论了如何将它用于非单调知识进化,说明了非单调知识进化与传统的工程技术、用户接口、规则管理、长事务以及数据库设计等技术之间的关系。论文着重强调非单调知识进化这一问题的重要性与复杂性,提出应当将它作为超大型数据库及知识库的基础研究。

本届VLDB会议上共发表了九篇有关演绎数据库的论文。从数量上讲,没有任何一个专题的论文可与之匹敌。在内容上,除了一篇是有关多重集(multiset)的形式化语义以外,其余八篇全部是有关递归查询的算法研究,就内容的集中性而言,也不是任何其它专题可与之比拟的。

最近几年里,演绎数据库的递归查询一直是人们十分感兴趣的问题。这方面的研究大体上分为两个方面:其一是研究递归查询表达方式,其二是寻找复杂度较低的查询处理算法。目前尚未找到一个一般的效率较高的处理算法,但对其中的某些子类却有了一些研究成果。例如,在对以Datalog为基础的查询研究中,已经有了对于可分离(separable)程序、右线性(right linear)程序、左线性(left linear)程序、可交换(commutable)程序、可分解(factorable)程序以及其它各类程序的优化算法。与此同时,将递归查询表达成传递闭包(transitive closure),对传递闭包的算法研究也取得了很大进展。这

次VLDB大会上发表的递归优化论文情况大致如下:

基于图的递归查询优化论文一篇。论文题目为“Factoring Augmented Regular Chain Programs”。在过去若干年里,论文作者P.Wood一直从事基于图的语言(graph-based language)的研究。这种语言的特点是将DB查询表达成有向图与路径,并采用正则表达式说明对路径的某些限制。这种表示方法允许对正则表达式中只含常量的查询进行优化。在这一基础上,作者对含有变量的正则表达式所表示的查询进行了研究,论文阐明了属于“扩充正则链”(augmented regular chain)这一类程序,总是可以转换成可分解的(factorable)Datalog程序,从而存在有效的算法。

关于“传递闭包”算法的研究共有三篇。“传递闭包”被认为是下一代数据库系统应当具备的十分重要的操作,“传递闭包”可用于解决与“材料单”、“最短路径”、“关键路径”、“路径源点”、“路径端点”等应用有关的各种递归查询。有关“传递闭包”的算法研究最近两年十分活跃。

在题为“Hybrid Transitive Closure Algorithms”的论文里,作者R.Agrawal等人将目前有关“传递闭包”的算法分为三大类。第一类称为“迭代算法”。这类算法包括“半纯真”(seminaive)、“对数”(logarithmic)算法等。它们的特点是重复地计算某一关系代数表达式,直至没有新的结果产生为止。第二类被称之为“直接算法”。它包括基于矩阵的算法、基于图的算法等。基于矩阵的算法将传递闭包的计算化成矩阵的表示与操作,而基于图的算法则将传递闭包的计算变为图的遍历问题。“直接算法”的特点是对每一个元素(例如一个节点或一条边)只计算一定次数(通常只考虑一次)。第三类是“混合型算法”(hybrid algorithm)。它是综合其它算法而形成的。在“Hybrid Transitive Closure Algorithms”论文里,作者提出了

一种将基于矩阵与基于图形这两种算法综合而成的混合型算法,主要目的仍然是为了减少传递闭包的计算时间。

另两篇有关传递闭包的论文是M.A.Houtsma等人的“Distributed Transitive Closure Computations;The Disconnection Set Approach”,与J.P.Cheiney等人的“A Parallel Strategy for Transitive Closure Using Double Hash-Based Clustering”。前一篇主要讨论分布环境下传递闭包的计算,提出根据数据划分,将查询分解并采用非联结集(disconnection set)方法计算传递闭包的算法。在后一篇论文里,作者提出了对传递闭包的一种并行算法。这种算法使用磁盘上的双散列丛集技术,利用有限的内存而避免过多的I/O开销。当P个处理器用于传递闭包的计算时,它可以使计算时间降低为原来的 $1/P$ 。

有关演绎数据库的其余四篇论文全部是关于自底向上(bottom-up)的递归优化算法的研究。自底向上也称之为前向链接(forward chaining)或定点(fixpoint)计算。它由已知事实出发,不断地应用规则,直至获得查询结果。自底向上的计算一般都采用“一次一个集合”的方法,它与PROLOG通常所采用的自顶向下,“一次一个记录”的方式正好相对应。目前的自底向上算法中有相当一部分是前几年提出的魔集(magic set)方法、半纯真方法的扩展。

在“How to Forget the Past Without Repeating It”的论文里,作者J.Naughton等人针对自底向上计算中的空间开销较大的问题,设计了一种使用“滑动式窗口制表”(slidding window tabulation)的算法,既保持了自底向上算法原有的时间效率,又降低了空间开销。

在“Rule Ordering Bottom-up Fixpoint Evaluation of Logic Programs”的论文里,作者R.Ramakrishnan等人探讨了如何利用规则的顺序来优化逻辑程序的自底向上

不定点求值,提出了三种利用顺序进行优化的算法。另一篇论文“Efficient Implementation of Loops in Bottom-Up Evaluation of Logic Queries”也对利用顺序的优化进行了探讨。论文对半纯真算法进行了改进,它将相互递归的规则进行分裂,对分裂产生的子循环按照特定顺序计算,减少了连结运算的次数。

在题为“Right-, Left-, and Multi-Linear Rule Transformations that Maintain Context Information”的论文里,D.K-empt等人针对较早提出的右线性、左线性以及多线性规则优化算法进行了扩充,克服了原有算法只能对自变元是常量的规则进行优化的局限性,使之能够处理自变元为变量的更为一般的情形,并且加速了递归问题的求解速度。

2.面向对象数据库(OODB)

在这次会议上,OODB也是论文的集中点之一,其内容大致集中在提高性能、OODB程序设计语言和实现技术等三个方面。OODB的论文已不再是介绍一些OODBMS原型或面向对象数据模型的设计,而是比较深入地提出和研究OODB一些实现技术问题和概念问题。会上还举行了一次OODB的专题讨论和一次OODB辅导报告。通过以上学术活动,我们获得如下的印象:

(1) OODB的方向是肯定的

OODB是为了满足各种新的数据库应用的需要而产生的数据库技术。诸如多媒体数据、空间数据、复杂对象、超文本、知识、时态数据等的管理都可以用OODBMS支持。面向对象程序设计是软件工程的重要发展方向之一。它可以提高程序设计的生产率、重用性以及可扩充性。发展面向对象程序设计必然要求相应地发展OODB。OODB可以统一地管理数据和各种表示形式的知识,OODB与基于知识的系统相结合,也是一个重要的发展方向。在OODB专题讨论会上,在五个发言人中,只有西德Karlsruhe大学的Lock-

emann教授开始持对OODB持一些消极观点,其他几个发言人都是肯定OODB的发展方向。其实Lockemann所提出的反对理由都是人所共知的,例如性能不好、缺少理论基础和形式化描述、没有标准、没有新的内容而是一种时髦等等。这些都是OODB发展中的问题,而不是否定其发展的本质问题。最后,Lockemann又做了一个补充发言。第一句话就是“我是一个OODB的坚定的信仰者”。在会上,未闻其他怀疑OODB的论调。OODB的发展是必然的。

(2) OODB的发展水平及趋势

目前,OODB核心系统的一些技术问题都已基本解决。诸如数据模型、数据语言、查询处理、对象管理、消息管理、版本管理、事务管理、物理结构等等都找到了具体的解决办法并有了实践经验。目前已出现了一些商品化的OODBMS,例如Gem Stone(Se-rvio Logic)、Ontos(Ontologic System)、ORION-II(Itasca Systems)、STATICE(Symbolics)、Vision(Innovative System Techniques)等。ANSI也成立了一个OODB标准化的工作组。当然,所谓OODBMS商品是指其在软件市场上可以买到,并不意味着它们已被广泛地使用。发行这些OODBMS产品的都是些小公司、新公司,他们多少希望通过出售OODBMS另辟蹊径,寻找机会。IBM等大公司迄今还未看到这方面的动向。目前,OODB主要用于特定的领域,还不能说已被工业界和广大用户所接受。不过它的应用范围将不断地扩大。如果面向对象程序设计方法有了大的发展,面向对象程序设计语言成为主流的程序设计语言,OODB将获得广泛的应用。

从目前的情况来看,OODB与关系数据库(RDB)的关系不同于七十年代初关系数据库与网状数据库的关系。那时的争论是在同一主要应用领域(即事务处理),究竟用关系数据库,还是用网状数据库,也就是谁取代谁的问题。现在OODB的提出,是在肯定关系数

数据库基本适合事务处理的前提下,对某些非传统的应用,用OODB补其不足。这是相辅相成的关系,而不是对立或取代的关系。在会上,未闻有取代之说,也没有OODB与RDB谁优谁劣的议论。事实上,这次会议上还有很多关于RDB的文章。

(3) 可研究的问题

OODB技术还不能算成熟,有不少问题仍在研究解决中。从会上的论文来看,下列问题有待进一步研究:

- 改善实现技术,提高OODBMS性能。
- 精确定义OODB的一些概念,发展OODB形式化理论。
- 与基于知识的系统相结合,增加演绎功能。
- 与面向对象程序设计语言相结合,把数据库和程序设计语言集成为一个应用软件开发环境,解决两者不匹配的问题。
- 发展OODBMS的开发工具和OODB的设计工具。
- 解决OODB与传统数据库的集成问题。

8. DBMS实现技术

这方面的文章相当多,遍及查询处理、事务管理、缓存区管理、物理结构以及完整性约束等。有传统数据库的,也有非传统数据库的;有集中数据库的,也有分布数据库的。这类文章的特点是工作做得比较深入细致。不停留在概念化的论证上,而且还做了许多分析、模拟或实验。这点是值得国内数据库界借鉴的。下面将这方面的有关问题分述如下:

(1) 查询处理

连接的方法,特别是分布连接方法在论文中讨论很多。有关基于散列(hash-based)的连接论文还作为一个专门的分组进行报告。这方面的内容将在后面介绍。

会上也有两篇文章讨论嵌套关系。一篇是西德N. Sudkamp等人的“Elimination of Views and Redundant Variables in an SQL-

Like Database Language for Extended NF Structure”,另一篇是比利时的G. Hulin的“On Restructuring Nested Relations in Partitioned Normal Forms”。前者主要讨论支持嵌套关系的数据模型中的查询优化,后者则主要讨论如何将一类特定的嵌套关系——“划分范式”关系转换成具有泛关系形式的平面关系,从而用平面关系的语义来描述嵌套关系。论文定义了一种新的算法及操作,可用于进行上述转换并使转换中的信息丢失最少。

(2) 事务管理

传统的事务模型要求事务的作用原子化,事务的执行可串行化。这在一般的事务处理中还可适用,但在长事务或嵌套事务处理中,这些要求成为提高性能的障碍。近几年来,有不少论文提出修改事务模型,以弱化这些要求。在这次会议上H. F. Korth等人提出了用补偿事务进行恢复的形式化理论。一般的事务是不允许其他事务访问其未提交的数据的。否则当该事务撤销时,会引起其他事务的连锁撤销。要等一个长事务提交后,才允许其他事务访问其有关数据,往往会导致长时间的等待。在合作的事务中,例如在合作设计中,有时还希望了解其他合作者的未提交的数据。为了满足这些需要,一般用一个作用相反的补偿事务来消除夭折的事务的影响,但不是在所有情况下都能奏效的。本文在理论上分析了这些条件,也讨论了一些实现问题。另外,也有两篇论文讨论在分布数据库中利用多副本进行恢复的技术。

在并发控制方面,IBM公司的C. Mohan发表了两篇论文,其中一篇是用记录序号(Log Sequence Number)来确定某页数据是否提交,以提高并发度。这项技术已在AS/400、SQL/DS、DB2、IMS中成功地得到应用。但令人不解的是,这项成果迟到1990年才正式发表,或许是商业原因。另一篇是关于B树的一个封锁方法,称之为ARIE

S/KVL(Algorithm for Recovery and Isolation Exploiting Semantics Using Key-Value Locking)。这两篇论文所介绍的并发控制技术实用、具体、可供国内研制DBMS者参考、其基本内容也可反映在有关教材中。

(3) 缓存区管理

本届VLDB论文中,有不少文章讨论缓存区的管理。随着内存的增大,DBMS的缓存区也相应地扩大。在有些系统中以缓存(cache)相称。缓存区扩大后,可以显著地减少访盘的次数,是提高DBMS性能的主要措施之一。论文中主要讨论缓存区的调度和更新算法、保持缓存区数据的一致性以及命中率分析等。

(4) 物理结构

在DBMS的物理结构方面主要讨论索引和散列技术。在索引方面主要讨论多属性索引、时态数据索引等特殊问题。在散列技术方面主要讨论利用散列技术进行连接操作。

(5) 完整性约束

有些论文涉及到完整性约束问题。从论文所反映的情况看,新的大型DBMS都或多或少地加些完整性约束措施,尤其引用完整性约束(referential integrity constraints),几乎主要的DBMS都有。其表示方法不外乎三种:1.用非过程语言描述完整性约束。2.用规则描述完整性约束。3.用触发(Trigger)的条件来体现完整性约束。在发生违反完整性约束的事件时,或禁止于事务执行之前,或撤销于事务执行之后。S.Ceri提出一个修补法,就是在发生违反完整性约束事件以后,采取修补(repair)措施,把数据库恢复到某一一致状态(不一定是执行前的状态)。

4. 硬件发展所带来的数据库研究的新课题

在本届VLDB会议上,虽无此标题或提法,但许多论文所研究的问题属于这个范畴。为了强调硬件的发展对数据库技术的影响,下面将在此标题下综合介绍有关的内

容。众所周知,计算机技术有两个发展的动力:一是应用的需求;二是硬件的发展。前者是需要,后者是物质基础。数据库技术的发展也是如此。计算机应用对数据库技术提出的要求,上面已有介绍。下面根据本届会议的论文内容,介绍硬件发展所带来的数据库研究的新课题。

(1) 可写光盘的出现和使用

由于光盘可以高密度地存储大量信息,为多媒介数据库、文档数据库和时态数据库等的研究提供了物质基础。在本届VLDB会议上,时态数据库成为一个专门的分组。一般数据库只反映现实世界的当前状态,不反映历史状态。时态数据库不但记载数据本身,而且还记载数据的有效时间。过期的数据不必删除,可以作为历史数据保存。甚至在数据库中还可保留与未来有关的数据。数据在时间维展开以后,数据量大增,磁盘难以承受。光盘出现以后,为时态数据库的应用提供了物质基础。在本届VLDB会议上,关于时态数据库的论文共有三篇:第一篇是关于时间索引(time index)问题,提出按时间建立索引,作为时态数据的存取机构,可以有效地支持WHEN操作、时态聚集函数处理、时态SELECT操作以及时态连接操作等,从而改善时态数据库的性能。第二篇提出一个时态关系代数,可用于任何支持离散线性有界时间的时态关系数据模型。文中把五种基本关系代数运算推广到时间域,并定义了一种线性递归运算。这种时态关系代数和基于以until和since作时态操作的谓词时态逻辑的时态演算具有同样的表达能力,可作为时态关系完备性的基础。第三篇研究数据库系统对时态数据的支持,提出把时态数据当作复杂对象处理,并以西德Kaiserslautern大学所研制的PRIMA数据库系统的数据模型MAD为例,说明时态数据的表示,文中还讨论了减少存储开销和提高访问速度等问题。

(2) 大容量内存的出现

由于内存芯片集成度的提高和价格的下

降,单机内存容量已可扩充到100兆字节以上,停电时可用内接电池供电,可以成为不易失存储器。这项硬件的发展对数据库技术产生很大的影响。

过去,计算机系统只能向DBMS提供有限的缓存,在此基础上发展了嵌套循环和排序归并等基本连接方法。近年来,以散列技术为基础的连接方法有所发展。在磁盘上将关系按连接属性散列后再连接,是得不偿失的。如在内存中对整个关系或部分关系按连接属性散列,再进行连接,可以获得好的性能。在本届VLDB会议上,以散列为基础的连接方法成为一个专门分组。这些论文讨论内存的有效使用、溢出处理等问题。由于并行处理在数据库技术中用得愈来愈多,有三篇论文讨论在多机情况下的以散列为基础的连接问题。

单机的内存容量是有限的,如果采用多机系统,总的内存容量可与磁盘的容量相当,可在其上建立数据库。这称为内存数据库。内存数据库消除了I/O瓶颈,性能很高,但有其特殊问题,例如数据结构、存取路径等的设计均不同于一般数据库。这次有两篇这方面的论文。

(3) 小型温盘容量加大、价格降低、体积缩小

3.5英寸温盘的出现,这个趋势更加明显。数据库可以不用大容量磁盘组成,而代之以磁盘组(disk array)。每个磁盘的容量虽然不大,但由于磁盘的个数多,总的存储容量仍是很大的,可以在它上面建立大数据库,这带来几个好处:

- 磁盘操作的并行性增加,可望缓解乃至解决I/O瓶颈问题。

- 不同的磁盘,其失效机制是相对独立的。如采用多副本,有利于容错。

在这次会议上,有两篇关于这方面的论文。一篇研究数据在磁盘上的分布,另一篇研究在失效情况下磁盘组的性能分析。

(4) 微机和计算机网络的性能大幅度提

高

这种发展对数据库技术的影响反映在两个方面:

- 数据库的并行处理受到重视 目前,数据库并行处理结构主要有两种:一种是多处理机结构,也称为share everything结构,由多处理机共享内存和磁盘。另一种是多计算机结构,也称为share nothing结构,由多个独立的计算机通过高速网相联。由于多处理机并行处理的吞吐量(throughput)随着处理机数的增加而迅速饱和,目前用得更多的是share nothing结构。会上的有关论文也都以此结构为背景,其中三篇讨论并行处理时的以散列为基础的连接,另外两篇研究数据库并行处理中数据分布问题。

- Client/server正成为数据库系统的标准结构 由于硬件价格的下降,数据库的处理和应用程序的处理趋向于由不同的计算机系统来担任。处理数据库任务的计算机系统称为数据服务器(data server),数据服务器实际上就是数据库机器。不过它是由通用的微处理机(例如采用RISC结构的微机)构成,其本身一般不止一台计算机,而是由多台计算机构成的并行处理系统。处理应用程序的计算机称为主机(host)。主机与数据服务器通过计算机网相联。主机如需访问数据库,可将此任务交给数据服务器,主机扮演委托者(client)的角色。而数据服务器接受各个主机的请求,专职地提供数据库服务,扮演服务者(server)的角色。这两者之间的界面如何划分,是论文中讨论的一个主题。另外,由于数据服务器是专用计算机系统,可按DBMS的要求进行设计,从而摆脱了通用计算机给DBMS带来的限制和不足。其操作系统一般是专用的。DBMS的一部分功能(例如并发控制、恢复、缓冲区管理等)可以下放给操作系统,而DBMS可以多做一些本来由应用程序所做的工作(例如检索、检索加速等),为用户提供一个更好的开发应用程序的平台(platform)。这次有

三篇论文讨论数据服务器的结构。主要是讨论数据存放 (data placement) 和平衡负载等的策略。并行处理的方式有三种, 即同一操作由多机并行处理 (intra-operator), 多个操作由多机分工并行处理 (inter-operator), 多个查询由多机并行处理 (inter-query)。这三种方式可用于同一系统中, 视操作和查询的复杂程度、查询的响应时间要求以及当时的负载情况选择使用。数据服务器的商品在会上提得比较多的是 Teradata 公司的 DBC 和 Tandem 公司的 NonStop SQL。

由于硬件的发展, 冯·诺依曼计算机给数据库系统所施加的三个瓶颈 (I/O、CPU、内存) 都将突破。这将大大影响 DBMS 的实现技术, 甚至数据库理论的发展。这个倾向在这次 VLDB 会议中表现得十分明显。

5. 其他

与往常的 VLDB 会议不同, 这次无分布数据库专门分组, 但有关分布数据库的论文很多。许多关于 DBMS 实现技术的文章, 都是以分布数据库为背景。分布数据库不是过时了, 而是成熟了。正象七十年代初期, 关系数据库可能成为当时学术会议的专题。但时至今日, 就很少有学术会议把关系数据库再列为一个专题, 而关于关系数据库的论文却很多。根据会上的论文所反映的情况, 同构的分布数据库和 client/server 结构的数据库已趋成熟, 有了这方面的商品。分布数据库剩下来的一个难题, 就是如何把已运行的、分布的、异构的数据库集成起来。这是一个迫切需要解决的实际问题。关于这个问题, 会上有两篇文章和一个辅导报告讨论这个问题, 但没有大的突破。其中一篇文章介绍普渡大学所研制的 Interbase。它已将 GU-

RU、INGRES、Sybase、dBASEIV 等四种运行于不同的硬、软件环境的数据库系统集成起来。文中只介绍了它的事务模型。这种事务模型有三个特点:

- 功能重复 (function replication) 一个子事务可以访问多个数据库, 例如订机票这个子事务, 如一个航空公司的票订不到, 就可以自动转到其他航空公司去订。

- 混合事务 (mixed transaction) 即允许一个全局事务既包含可补偿子事务, 又包含不可补偿子事务, 可补偿子事务可以在全局事务提交前提交, 而不可补偿子事务只能与全局事务一起提交。

- 标注事务的处理时间 有些事务对其处理时间有一定的要求, 例如订票的时间最好放在票价打折扣时。

多数据库的另一篇论文是 MIT 提出的 Ploygen 模型 [Ploygen 是作者杜撰的一个字, 表示多 (poly) 种来源 (gen)], 作者认为在实际应用中, 有时不但不需要位置透明 (location transarency), 相反地还要求标明数据的来源, 例如用以判断数据的可信度。有时同一数据也可以有多种来源, 更需要标明其来源。在作者所提的数据模型中, 加上数据来源标记, 并对这种模型发展了相应的一套代数运算规则。

可扩充 DBMS 结构近年来成为数据库的一个新的研究领域。在这次会议上, 除个别论文提到外, 没有一篇文章专门论述这个问题。

关于数据库应用, 历来不是 VLDB 会议的重点, 一般只接受刺激数据库技术发展的数据库高级应用 (advanced application) 方面的论文。这次有空间数据库 (spatial database) 这个分组, 并有这方面的辅导报告。