第5卷第6期
1994年6月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 5, No. 6
June 1994

分布数据库系统内核TX的设计与实现*

李北星　　王小京

(清华大学计算机科学与技术系,北京 100084)

摘要　本文介绍由清华大学计算机科学与技术系数据库组于1987—1990年开发的一个开放系统——分布数据库系统内核TX的设计思想和实现技术.从其部分实现开始,TX就在他们的分布数据库、知识库、超文本数据库等研究中发挥着重要作用.本文着重于TX的体系结构、数据组织、查询处理、数据字典及并发控制.

关键词　分布式数据库.

分布数据库的概念自诞生之日起已经有10多年的历史了.在我国有关研究的开展大致从80年代初开始,并且在数据库技术研究领域保持了相当一段的繁荣时期.随着对分布数据库研究的深入,它对整个数据库技术,特别是关系数据库在我国的发展产生了深刻的影响,大大地促进了国内数据库管理系统的研制工作.

本文将全面介绍在设计与实现分布数据库系统内核TX时的各种技术考虑,内容包括TX的系统结构,数据组织与存取机制,查询处理,数据字典管理,并发控制.

TX系统已于1990年起在SUN 4/260 UNIX和HP 9000/840 UX上运行,整个系统由C语言写成.

1　TX总体结构

正如前面指出的,TX作为灵活的开放系统,力求达到以下3个目标:

1.作为独立的关系数据库管理系统,支持即席用户和应用程序访问数据库.

2.作为数据库服务器,与其它系统结合,如与知识库管理系统、分布数据库管理系统结合.

3.支持应用程序和应用系统以程序方式访问数据库数据.

为此,TX提供3层数据库服务接口.第1层为标准SQL语言[1],包括SQL的各种嵌套形式和聚集函数查询,提供丰富的非过程的查询手段.第2层为关系代数语言[2,3],尽管关系代数本来作为SQL语句的转换形式供查询处理,关系代数自身完全可以作为查询语言使用.第3层为数据存取函数,提供元组级访问服务.TX的结构由图1给出.

* 本文1991—07—03收到,1992—04—10定稿

作者李北星,34岁,讲师,主要研究领域为数据库.王小京,女,34岁,讲师,主要研究领域数据库.

本文通讯联系人:王小京,北京 100084,清华大学计算机科学与技术系

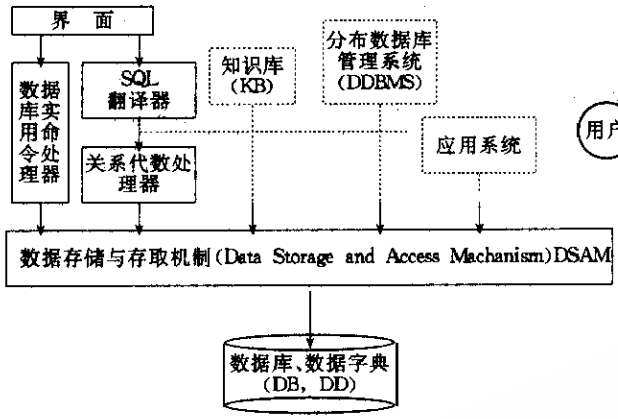

图1 TX的总体结构

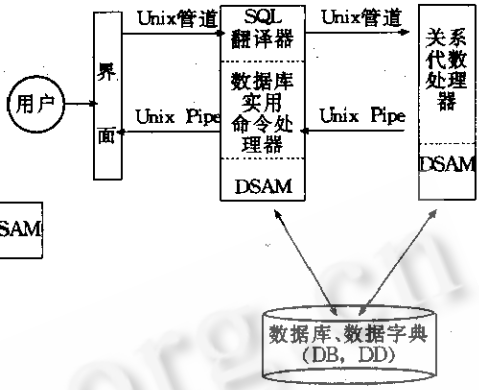

图2 TX 的进程结构

TX 界面是一个接收 SQL 语句和数据库实用命令的屏幕管理系统,用户可以在此编辑和执行命令文件.数据库实用命令是指那些涉及数据字典的操作,如建立关系、建立索引、改换存储组织等.数据库实用命令由数据库实用命令处理器解释执行.

SQL 语句先由界面接收并调整其内部形式,然后送给 SQL 翻译器将其转换为关系代数表示.转换方法的依据是文献[2].

关系代数处理器支持扩展的关系代数.关系运算,即投影(project)、筛选(select)、联接(join)等均包括在内.与 SQL 中聚集函数与分组操作相对应增加了 FN 运算.此外,利于分布处理的半联接(semijoin)也纳入进来.这些关系代数运算的定义可参见文献[3].为了保持 SQL 中对查询结果的重复元组的控制,即 SELECT 子句中出现 ALL 或 DISTINCT 限制词,引入不去除重复的投影.对于 SQL 中的数据操纵语句,即插入(INSERT)、删除(DELETE)和修改(UPDATE)语句,引入了 3 个相应运算 INS、DL、UP.

关系代数处理器对关系代数实施两步优化.首先是表达式等价变换,找出最优表达式,然后再对表达式中每个关系运算选择存取路径并确定各运算操作执行的次序.

数据存储与存取机制(DSAM)负责数据库的数据组织,提供一次一个元组的存取服务.

TX 的总体结构由 UNIX 的 3 个进程实现,如图 2 所示.前端进程为界面管理;数据库实用命令处理器、SQL 翻译器及 DSAM 组成第 2 个进程;后端进程包括关系代数处理器和 DSAM.进程之间通过 UNIX 管道通信.

2 数据组织与存取方法

数据存储与存取机制(DSAM)是建立在 UNIX 文件系统上的,它所管理的基本对象是关系,每个关系对应一个 UNIX 文件.数据存放和取出以页为单位,一页长度固定,可以在系统生成时定义,如定为 1KB、2KB、4KB 等.

DSAM 支持 4 种数据组织,即到达顺序组织、关键字排序组织、Hash 组织和 B+树组织.在到达顺序组织中,元组按照加入的顺序插入,新元组总是加入到最后一页.当最后一页

容纳不了时则增加一页,使之成为最后一页.关键字排序组织中元组按关键字值的顺序加入到相对应的页中.元组在 Hash 方法中的位置则是由一个 Hash 函数计算得出.对于关键字排序组织和 Hash 组织都可能出现具有某一关键字值的元组所要存放的页容纳不下的时候,这时需要进行溢出处理,增加溢出页来存放这些元组.B+树组织是专门用于索引机制的.由于 B+树优越的均衡性和顺序排列,经索引的任何方式查找都能迅速定位.一个关系上能够对不同的属性或属性组建立多个索引,从而满足多种属性值的相关查找.对元组的迅速定位依赖于元组标识符(TID),它是由元组所存放的页的号码和页内位置构成.索引项就是由关键字值和 TID 组成.

DSAM 提供的在关系上的操作有打开(open)、关闭(close)、插入(insert)、删除(delete)、修改(update)、限界定位(scanrange)和取元组(get).要对一个关系操作,先要将其打开.插入、删除和修改操作能够改变关系内容.取元组操作按照给定的 TID 取出元组.通常 TID 值通过调用 scanrange 来确定,当给定关键字取值范围后,scanrange 可以找到起始 TID 和终止 TID.在 TX 这样的关系系统中,扫描关系的一个部分或整个关系是必须的操作,这在 Get 里 TID 可以自动连续增值以达到连续取的目的.通过 scanrange 的定位只能是对关键字排序或 Hash 组织的关系起作用.对于到达顺序组织,只能从头至尾一个一个取.作为关系的一种,索引总是关键字排序的,故经索引的定位总是有效的.索引一经建立,对它的插入、修改和删除都由系统自动完成.

为了提高数据存取的效率,DSAM 保持着若干页的内存缓冲区.当读取数据请求到来时,首先搜索缓冲区.若缓冲区中可以找到,则不必要实际存取一页.在缓冲区找不到的情况下,才调入一页进入缓冲区.同样,修改数据也先对缓冲区实施.缓冲区管理按 LRU 方式进行.

3　查询处理

对于 SQL 的支持需要设计较为精巧的处理策略来保证系统性能.在 SQL 中,一个查询语句只指出要查找什么,而不说如何找到.典型的查询块为

SELECT　〈选择列举表〉
FROM　〈关系名表〉
WHERE　〈条件〉

其中,〈选择列举表〉给出了要检索的项,〈关系名表〉指定了要引用的关系,〈条件〉说明了检索结果应满足的条件.由于在 SQL 中条件的表达可以包含新的查询块,从而形成多重相互引用的嵌套查询,故处理起来不是容易的事情.

在 TX 中将 SQL 转换为关系代数再进行优化.这一方法有以下几个理由:

1.将嵌套查询转化为非嵌套形式利于优化处理[4];

2.关系代数为非嵌套的偏序结构,趋于过程化;

3.可运用已经提出的转换方法[2];

4.关系代数是关系级运算,更适合于分布数据库[3].

文献[2]给出的 SQL 到关系代数的转换方法中有一些限制.基于实现的需要,我们去掉了其中的几个,如对数据操纵语句的转换(insert,delete,update)、允许使用算术表达式、可

以查找重复元组等.

SQL 语句经过转换都化成了等价的关系代数表达式.这时的优化工作分两步进行.首先是根据启发式规则,进行表达式等价变换,找到最优表达式.所应用的启发式规则是:

1.尽早通过投影(Π)和筛选(σ)缩小关系的体积.

2.提取公共子表达式,使之只求值一次.

3.应尽量少产生中间关系.

下面通过一个例子说明一下运用上述规则进行的表达式变换,有关变换中用到的关系代数变换规则就不在这里给出了.

设有关系emp(name,e#,sal,d#,sex,age)

dept(name,d#,mgr#)

初始关系代数表达式

$$\Pi_{emp.name}((emp \underset{d\#=d\#}{\infty} \sigma_{mgr\#=373}(dept)) - (\sigma_{sal>500}(emp) \underset{d\#=d\#}{\infty} \sigma_{mgr\#=373}(dept)))$$

经提取公共子表达式

$$emp \underset{d\#=d\#}{\infty} \sigma_{mgr\#=373}(dept)$$

变换为 $\Pi_{emp.name}(\sigma_{sal\leqslant 500}(emp \underset{d\#=d\#}{\infty} \sigma_{mgr\#=373}(dept)))$,再经投影与筛选下移变为:

$$\Pi_{emp.name}(\Pi_{name,d\#}(\sigma_{sal\leqslant 500}(emp)) \underset{d\#=d\#}{\infty} \Pi_{d\#}(\sigma_{mgr\#=373}(dept)))$$

上述表达式中的投影和筛选并不是都要单独求一次的,可以将对 emp 和 dept 两个关系的 Π、σ 分别一次完成,这就是启发式规则(3)的运用.同理,最后结果 $\Pi_{emp.name}$ 的投影也能与联接同时进行.

优化工作的第二步是存取路径选择.这时的任务是确定各个关系代数运算的执行顺序,所运用的实现方法和使用的存取方法.对于联接运算,有两种实现方法即归并排序法(merge—sort)[5]和嵌套循环法(nested—loop)[6]供选择.存取路径选择是以开销估计为依据进行的,开销估计的目标函数为数据存取的页数.有关开销估计所需要的数据库统计信息以及估计公式在这里就不一一给出了.

4 数据字典/数据目录的设置与管理

在数据库管理系统中,数据字典/数据目录是系统的数据库.数据库管理系统所管理的对象,即数据、用户、资源都要在字典/目录中建立相应的描述信息.从某种意义上讲,数据字典/数据目录控制着系统的运行.

作为开放系统,数据字典/数据目录的灵活性同样有着重要意义.在 TX 现行版本中数据字典/数据目录包括有模式描述、子模式描述、存取方法描述、完整性描述、安全性描述、表查询描述、资源描述、数据分划描述及数据分配描述.后两个是关于数据分布的信息,将提供给分布数据库管理用.

数据字典/数据目录在 TX 里也是一类关系——系统关系.其组织方法和存取机制与用户定义的关系一样,当然不能对系统关系建立索引.用户可以使用查询语言查询数据字典的内容.为了适应开放系统的需要,系统人员可以增加数据字典/数据目录的内容,也可以设立新的数据字典.只要将初始化文件作些变动,新的系统关系就会产生.

5 并发控制

作为多用户系统,为了保证数据库的一致性和正确性,各用户同时对数据的操纵要加以协调.这一目标的实现以支持事务概念的事务管理为最终手段.现行TX系统中的事务概念相对简单,一个SQL语句或一个数据库实用命令为一个事务.在此我们着重介绍并发控制机制对事务管理的支持.

TX的并发控制以封锁机制实现,有共享锁和互斥锁.读操作可以共享资源,而写操作要达到相互排斥.封锁粒度的选择是根据事务的语义分别设有临界区锁、页锁、关系锁和数据库锁.

我们把一次仅允许一个进程使用的资源称为临界资源.临界区锁就是为了控制进程对某些临界资源的互斥访问而设立的.例如,在建立一个新关系前,先要检查此关系是否已经存在.两个用户同时要求建立一个关系,当一个用户检测到此关系不存在之后,进程切换到另一个用户,同样检测到该关系不存在,这时两个用户对关系的建立将产生错误结果.为了解决这个问题,可以在建关系之前设立临界区锁实现互斥.若有其它用户在建立关系则等待,直到获得临界区锁.

页锁的封锁对象是输入输出的单位—页.页锁是为封锁系统关系设立的.当用户调用数据库时都要访问系统关系,但每个用户涉及的数据并不多,采用较小的封锁单位可以提高并发度.

关系锁和数据库锁都是较大范围的封锁.前者用于一般查询,而后者则是对数据库进行特别操作,如加载与建立后备时使用.

6 结束语

以上介绍了TX的设计思想和若干实现技术.作为一个开放系统,TX已成功地运用于我系开发的知识库系统QKBMS/75[7]和数据库管理系统生成器.目前,一个支持超文本数据库的存储组织正在基于TX的服务加以实现.

参考文献

1 Final draft ISO 9075—1987(E): Database Language SQL. ISO TC97/SC21 WG3—DBL/RENO—2, Nov. 1986.

2 Ceri S, Gottlob G. Translating SQL into relational algebra: optimization, semantics and equivalence of SQL queries. IEEE Trans. of Software Engineering, 1985,SE—11(4).

3 Ceri S, Pelagatti G. Distributed databases: principles and systems. McGraw—Hill Book Company, 1984.

4 Kim W. On optimizing an SQL—like nested query. ACM TODS, 1982,7(3):443—469.

5 Blasgen M W, Eswaran K P. On the evaluation of queries in relational data base system. IBM System Journal, 1977(4).

6 Pecherer R M. Efficient evaluation of expressions in a relational algebra. Proc. ACM Pacific 75 Conf. ACM, New York, 1975:44—49.

7 周立柱,范正平. 提高知识库管理系统效率的若干措施及效果. 清华大学学报, 1990,30(4).

THE DESIGN AND IMPLEMENTATION OF TX ——A KERNEL FOR DISTRIBUTED DATABASE SYSTEMS

Li Beixing and Wang Xiaojing

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

Abstract This paper presents the design considerations and implementation technics of an open system TX, a kernel for distributed database systems developed by the database group of the computer science department at Tsinghua University from 1987 to 1990. TX has played important roles in their further research on distributed database system, knowledge based system, and hypertext database since it was partially implemented. The focuses of this paper are the architecture, data organization, query, processing, data dictionary, and concurrency control of TX.

Key words Distributed data base.

中国计算机学会 1994、1995 年国际学术活动计划

会议名称	时间	地点	主办单位	联系人
The 3rd Pacific RIM International Conference on Artificial Intelligence 第三届太平洋地区国际人工智能会议	8.16—8.18	北京	CCF CAA	史忠植 2565533—419 Fax. 2567724
世界华人计算机科学技术及产业研讨会	8.28—9.1	北京	CCF CAS	李光华 2560911
HKICC(GZ)	10.3—10.4	广州	CCF HKCS	王介生 2565533—818
ICA'94 展览会	10.6—10.10	上海	CCF SCS B/I	徐桂珍 3726055
国际软件测试和软件可靠性研讨会	10.18—10.20	北京	IEEE SE 专委 CCF 容错专委	闵应骅 2565533—836
The 19th. Computer Software & Applications Conference 第 19 届国际计算机软件及应用会议	1995.10	北京	IEEE CS CCF	朱明远 6756956(O) 4919213(F)
ICYCS'95 第四届国际青年计算机学术会议	1995.8	北京	CCF 主办 智能中心承办	白硕 2565533—162
第四届 CAD & CG 国际学术会议	1995.10	武汉	CCF 主办 华中理工大学 (430074) 承办	华中理工大学 机一系 刘健