POLYBASE 系统中的查询处理技术*

王国仁　于　戈　单吉弟　郑怀远

（东北大学计算机科学与工程系　沈阳　110006）

摘要　POLYBASE 系统是一个集成的多数据库系统. 本文主要讨论了 POLYBASE 系统的体系结构、集成数据模型，提出了一个基于加权查询树的查询表示模型，设计并实现了基于加权查询树查询表示模型的一个有效的导航查询处理算法.

关 键 词　多数据库系统，集成数据模型，加权查询树，查询处理.

近年来，在许多新的计算机应用领域中，如计算机综合(集成)制造系统(CIMS)、计算机辅助工程设计与制造(CAD/CAM)、工程信息系统(EIS)、计算机辅助软件工程(CASE)、飞机订票系统和信息检索系统等，其计算环境是由分布的、异构的和自治的软、硬件系统组成的. 在这些应用系统中既有关系数据库系统，又有层次和网状数据库系统，甚至还存在大量的文件系统. 从数据管理和应用的角度来看，在这些应用领域中的数据是分布的和异构的. 而且除了要保持各个应用系统中信息处理的自治性外，它们还需进行信息的交换与共享. 因此，如何把这些应用系统中的异构数据库系统集成起来以实现信息的集成与共享是当前许多应用系统所迫切需要解决的实际问题.

POLYBASE 系统是一个集成的多数据库系统，它的主要目标是实现传统的数据库系统的集成，以实现这些系统之间信息的交换与共享. POLYBASE 系统采用基于→1NF 数据模型和源标签集成机制的异构数据库集成方法[1~6]，源标签集成机制[1]有利于集成冲突问题(命名冲突、模式冲突和数据冲突)[7~9]的解决、查询处理的优化与分解，以→1NF 数据模型作为集成数据模型对于集成传统的数据库系统是比较有效的.[2~6]POLYBASE 系统采用一个基于客户/服务器模型的开放式体系结构，有利于提高系统的扩充能力和改善系统的适应能力. 有关 POLYBASE 系统的体系结构和集成数据模型将在本文的后面详细讨论. 本文的另一个主要贡献在于针对→1NF 数据模型提出了一个基于加权查询树的查询表示模型，并在此基础上设计并实现一个高效的导航查询处理算法，该算法经实验结果表明对于基于→1NF 数据模型的查询处理是非常有效的.

本文第 1 节介绍 POLYBASE 系统的体系结构；为了文章的完整性，第 2 节简要叙述

* 本文研究得到国家 863 高科技项目基金资助. 作者**王国仁**，1966 年生，讲师，主要研究领域为分布式数据库与多数据库系统. **于戈**，1962 年生，副教授，主要研究领域为分布式数据库及多数据库系统. **单吉弟**，1936 年生，副教授，主要研究领域为体系结构，数据库，网络. **郑怀远**，1931 年生，教授，博士导师，主要研究领域为数据库.

本文通讯联系人：王国仁，沈阳 110006，东北大学计算机科学与工程系

本文 1995－04－17 收到修改稿

POLYBASE系统中采用的集成数据模型;第3节详细讨论POLYBASE系统的查询处理技术,包括查询表示模型的定义、查询处理流程和导航查询处理算法;第4节总结全文.由于篇幅有限,有关多数据库系统的其它技术将在另文讨论.

1　POLYBASE系统的体系结构

在POLYBASE系统中采用基于客户/服务器模型的开放式体系结构,有利于系统的扩充性.该系统结构由多个客户器、多个数据服务器、一个事务服务器、一个字典服务器以及一组开放客户/服务器接口组成,系统中的各个模块和它们之间的关系如图1所示.

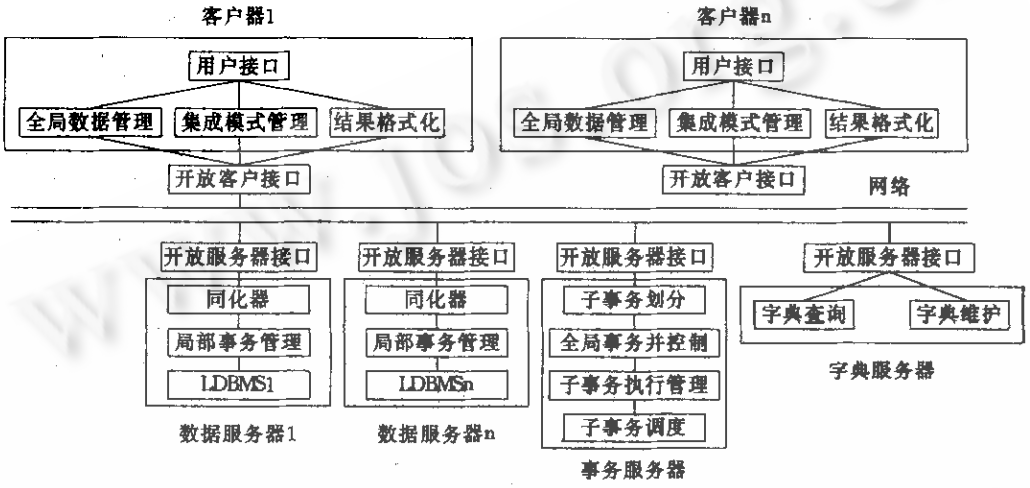

图1 POLYBASE系统的体系结构

客户器主要包括用户接口模块、全局数据管理模块、集成模式管理模块和结果格式化模块.用户接口模块主要负责命令的接收、词法与语法分析,并生成语法分析树;全局数据管理模块主要进行全局数据的管理,包括语义检查、权限检查,并负责将语法分析树翻译成公共数据语言;集成模式管理模块负责模式的集成与维护.全局结果格式化模块对各个局部返回结果进行转换、合成并格式化为全局数据格式后返回给用户.

数据服务器主要包括同化器和局部事务管理2个模块.同化器主要负责将从网上接收到的公共数据语言翻译成局部数据库系统所支持的数据语言;局部事务管理模块主要是配合全局事务服务器来完成全局事务的局部执行.

事务服务器包括子事务划分、全局事务并发控制、子事务执行管理、子事务调度4个模块.子事务划分模块主要负责将全局事务划分成多个子事务,并为每个局部事务管理器设置子事务集.全局事务并发控制模块将控制全局事务的执行顺序和全局模式的加(解)锁.子事务执行管理模块负责子事务的提交、恢复与补偿.子事务调度模块负责子事务间的调度.

字典服务器包括字典查询和字典维护2个模块.字典查询模块负责查询字典信息的检索,字典维护模块负责增加、删除和修改字典中的信息.

开放客户/服务器接口模块负责公共数据语言、中间结果在网络上的发送与接收工作.

2 POLYBASE 系统的集成数据模型

POLYBASE 原型系统采用基于源标签集成机制的→1NF 数据模型作为全局集成数据模型. 基于→1NF 数据模型的多源数据库集成技术特点是便于对传统数据库(如关系、层次和网状数据库)中的数据进行集成,以源标签方法来解决数据间的异构性,以→1NF 数据模型来处理工程应用中数据复杂性问题.

目前,用于数据库集成的数据模型主要有关系模型[1]、语义数据模型(如 ER 模型、EER 模型、OSAM* 模型、OERA 模型等)[7,8]、→1NF 数据模型[2,6]和面向对象的数据模型.[10]关系数据模型作为集成数据模型的特点是便于集成关系数据库系统,实现起来比较简单,但它的集成能力是极其有限的. 语义数据模型和面向对象数据模型作为集成数据模型的特点是集成能力强,便于集成存在于应用当中的语义,但实现起来比较困难,且还有许多问题尚处在研究阶段. POLYBASE 系统采用→1NF 数据模型作为集成数据模型是基于以下几点考虑的:①→1NF 的数据模型便于集成传统数据库系统,如关系、网状和层次数据库系统,这是大多数应用系统所迫切需要的;②→1NF 的嵌套关系模型具有相当的描述复杂对象的能力,这一点对于满足某些特殊应用领域(如工程应用领域)的复杂需求是非常重要的;③系统实现起来比较简单;④→1NF 数据模型对于其它模型而言是比较容易接受和理解的. 为了使→1NF 数据模型适合作为集成系统的数据模型,解决多数据库系统中的数据异构性和集成冲突问题,本文对它进行了必要的扩充,增加了源标签集成机制. 集成数据模型可定义为:

集成数据模型=→1NF+源标签

集成数据模型建立在→1NF 数据模型和源标签概念的基础之上. →1NF 数据模型是一种嵌套结构,即关系中还可以套关系,它主要用来描述和处理工程应用领域中的复杂对象. 而源标签主要用来描述一个集成数据模式中数据的来源,并为各类冲突问题提供了解决的基础. 有关→1NF 数据模型的定义请参考文献[2,3,6,11],有关源标签概念的定义及利用源标签如何解决各类集成冲突问题请参考文献[1,6,11],本文在下节中将重点讨论 POLYBASE 系统中的查询处理技术.

3 POLYBASE 系统中的查询处理

集成的多数据库系统 POLYBASE 的全局数据语言是 PSQLNF.[2,5,6,11]由于 POLYBASE 系统采用的全局数据模型是以→1NF 为结构基础的,因此 PSQLNF 的查询语言相应也采用了嵌套的 SFW 结构. 由于一个→1NF 关系的属性既可以是原子的,也可以是非原子的,在 PSQLNF 的 SELECT 子句中的属性既允许是原子的(即简单属性),也允许是非原子的(即另一个嵌套查询语句). 这样查询结果仍然是一个→1NF 关系,为实现语言的封闭性提供了基础.

由 MIT[1]提出的基于源标签集成机制的多源查询处理机制,是基于关系模型的,提出了一套基于源标签机制的关系代数,查询处理过程采用 2 遍扫描算法. 该查询处理机制和查询处理算法,虽然对关系模型比较适合,但却不适应→1NF 数据模型. 针对→1NF 数据模型和源标签集成机制,我们也提出了一套扩充关系代数[2,6],但其实现算法效率不高. 为了高效

地实现查询,本文试图提出一套简洁的代数系统和高效的查询处理算法.

3.1 查询表示模型

在POLYBASE系统中,其查询表示模型的基础概念是加权查询树.在给出加权查询树的定义之前,我们先来看看一个PSQLNF查询的具体例子:

```
select pcname, pcno,
      (select pdno, pdname,
            (select peno, pename
            from pemp
            where psal>1000)
      from pdept
      where pdno=301),
      (select pbname, pbno
      from pbran
      where pbno=2000)
from pcorp
where pcname="IBM";
```

上述查询查找IBM公司中301部门以及该部门中工资高于1000的雇员的情况以及分公司号为2000的分公司的有关情况.

对上面的嵌套查询的查询树可用如图2来直观地表示:

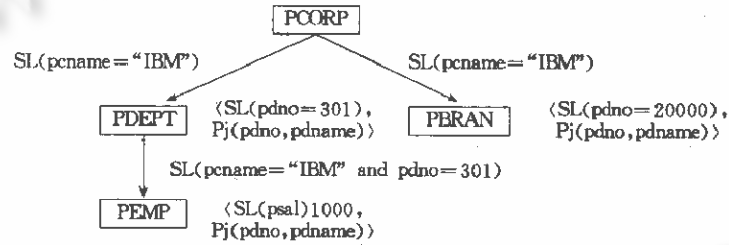

图2 一个加权查询树的例子

上面的查询树有3个基本构成要素:(1)结点.每个结点代表一个关系、片段或记录类型;(2)有向边.代表导航操作的方向;(3)权.在查询树中,总的来讲,加权动作表示施加操作,即权代表某类动作或操作.结点上的权是一个二元组$\langle SL_n, PJ_n\rangle$,表示对该结点施加的2个操作.有向边上的权为$SL_m$.它是一个导航操作,是由父结点产生而作用于其直接子结点的操作.

基于一1NF模式的查询处理系统可以定义为如下的一个代数系统:$QPS=(V,E,VF,EF)$,其中V是结点的集合;E为有向边的集合;VF是一个结点加权函数.$VF:V\rightarrow S_{11}\times S_{12}\times\cdots\times S_{1n}$.$S_{11},S_{12},\cdots,S_{1n}$分别是结点权值中各个分量值的集合;$EF$是一个有向边加权函数.$EF:E\rightarrow S_{21}\times S_{22}\times\cdots\times S_{2m}$.$S_{21},S_{22},\cdots,S_{2m}$分别是有向边权值中各个分量值的集合.

一个加权查询树可定义为满足QPS定义的如下代数系统:$WQT=(HA,NR,HF,NF)$,其中HA是查询树中所涉及到的所有高序属性(High_order Atribute)的集合.NR是查询树中所有有向边的集合.它表示了查询树中高序属性之间的嵌套关系(Nested Relationship).HF是一个结点加权函数:$HA\rightarrow SL_{11}\times PJ_{11}$.$SL_{11}$表示在高序属性上所有可能的选择操作的集合.$PJ_{11}$表示在高序属性上所有可能的投影操作的集合.$NF$是一个有向边加权函数:$NR\rightarrow SL_{21}$.$SL_{21}$表示查询树中各种嵌套关系之间所有可能的导航操作的集合.如图2所示的查询树是一棵满足WQT定义的加权查询树.

上述查询处理机制，由于结合了¬1NF 模型结构的特点，考虑并综合了网状和层次数据库中查找路径是非透明性的特点，所以在加权查询树的操作中仅需 2 个基本的关系代数操作.

(1)$SL_F(HA)=\{t|t\in HA \text{ AND } F(t)=\text{TRUE}\}$

(2)$PJ_{A_1,A_2,\cdots,A_n}(HA)=\{t^{A_1A_2\cdots A_n}|t\in HA\}$

3.2 查询处理流程

POLYBASE 系统的查询处理流程如图 3 所示.

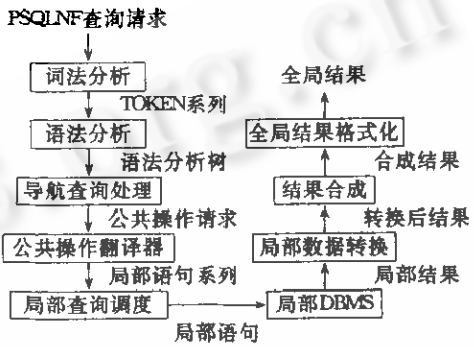

图3 POLYBASE系统的查询处理流程

1 条 PSQLNF 查询语句经过词法分析和语法分析后产生 1 棵语法分析树，该语法分析树是以加权查询树为基础的. 两者之间的区别仅仅是语法分析树中的有向边上的权为空，即导航信息并未产生. 导航查询处理模块则进一步对语法分析树进行导航查询处理，产生各条有向边上的导航权值，并将各有向边的导航权值作用于该有向边所指向的结点. 导航查询处理模块产生一公共操作请求系列，该系列是以公共数据语言为基础的. 公共操作请求经过网络调用接口被传送至公共操作翻译模块，该模块实际上是同化器中的查询处理模块. 公共操作翻译模块负责将公共操作请求逐一翻译为相应的局部 DBMS 所支持的操作语句系列. 局部查询调度模块实际上是局部事务管理中的一个子模块，局部语句系列经局部查询调度，由局部 DBMS 执行. 局部执行结果需要进一步进行格式转换以后进行全局结果的合成并进行格式化输出.

3.3 导航查询处理

对于一个复杂的嵌套查询来讲，其查询处理与优化过程也将是十分复杂的，在执行过程中可能要借助于中间结果才能完成比较复杂的查询. 在具体的过程中中间结果可借助于临时表来存放，这样一个复杂的查询在处理和执行过程中可能要产生不少的临时关系表. 但通过建立临时关系表来处理查询的方法有以下不足之处：

(1)降低查询处理的效率. 在大多数数据库管理系统中，模式的建立与删除操作(如关系数据库中的建表与删表操作)比其它操作(如关系库中的 DML 操作)要慢得多.

(2)使得分布式事务变得不可控制. 这主要是因为关系数据库系统对一个事务中的 DDL 语句进行特殊处理的缘故. 如在 SYBASE 数据库系统中，一个事务不能包含 DDL 语句，而在 ORACLE 数据库系统中，一个事务如果包含有一条 DDL 语句，则将这个事务自动分成 3 个独立事务来处理，即 DDL 语句之前的所有执行语句，DDL 语句本身和 DDL 语句以后的所有执行语句被分成 3 个独立的事务，ORACLE 系统对 DDL 语句的处理方式与 SYBASE 系统的处理方式大同小异. 这样的话，如果在一个分布式事务中包含有这样 4 条语句：(1)Begin transaction；(2)DML 语句；(3)Query 语句；(4)Rollback，在这样一个分布事务中，如果 Query 语句的处理需要产生中间结果表的话，就迫使相应的局部子事务在未接到全局提交命令之前就必须(自动)提交而使得 Rollback 语句不能正确执行.

为了克服上述缺点,提高查询处理的效率,我们必须设计一个不能产生临时关系表(至少中间结果不借助于创建临时关系来实现)的快速而简明的查询处理.导航查询处理机制正是基于上述考虑而进行设计的.

导航查询处理的基本思想是这样的:由于→1NF 的查询语言是一个嵌套的 SQL 查询语言,其查询树的表示也是基于→1NF 数据模型的.根据加权查询树,首先将广义高序属性表中满足条件的对象进行定位,这时广义高序属性表中的零序属性值已经确定,而其嵌套的高序属性的查询则将导航条件代入后可继续递归查找.导航查询算法可描述如下:

```
NavigateQuery(WeighedQueryTree, NavigateCondition)
/* WeighedQueryTree:相应查询的加权查询子树 */
/* NavigateCondition:相应加权查询子树 WeighedQueryTree 的导航权值或导航条件 */
BEGIN
/* 根据导航权值和该加权查询子树的根结点权值对根结点进行定位操作 */
{t1,t2,...,tn}=Location(WeighedQueryTree.RootNodeName,NavigateCondition),
/* 根据加权查询子树 WeighedQueryTree 查找其所有的加权查询子树 */
{st1,st2,...,stm}=GetST(WeighedQueryTree);
i=1;
while (i<=n) /* Loop for {t1,t2,...,tn} */
    BEGIN
      j=1;
      while (j<=m) /* Loop for {st1,st2,...,stm} */
          BEGIN
            /* 根据 ti 构造 stj 的导航权值,即给有向边<ti,stj>加权 */
            TNC=WeighEdge(ti,stj);
            NavigateQuery(stj,TNC);
            j=j+1;
          END
      i=i+1;
    END
END
```

上面的导航查询处理算法具有以下特点:(1)无论查询多么复杂,也无论查询嵌套的深度如何,在整个查询处理过程中无需建立任何临时表来存放中间结果,因为在导航查询处理过程中,每得到一个完整的→1NF 结果元组就进行格式化并组装输出.由于在关系数据库系统中创建一个关系表所耗费的时间比执行一条 DML 语句要大得多,所以在这层意义上导航查询处理算法比借助于临时关系表来暂存中间结果的查询处理算法的效率要高;(2)导航查询处理算法的另一个显著特点是以导航操作代替了连接操作.在关系数据库中连接操作是耗时最多的操作之一,而在导航查询处理算法中导航操作仅是一个选择操作,因此进一步提高了查询处理算法的效率;(3) 改善了分布事务的可控制性,这一点往往比算法效率更重要.

4　结束语

本文在介绍了 POLYBASE 系统的体系结构和定义了 POLYBASE 系统的集成数据模

型之后，详细讨论了 POLYBASE 系统中的查询处理技术，提出了基于加权查询树的查询表示模型的概念，在此基础上设计并实现了一个高效的导航查询处理算法，经过实验表明该查询算法对于¬1NF 的嵌套关系查询的处理是非常有效的.

参考文献

1 Richard Y W *et al*. A polygen model for heterogeneous database system: the source tagging perspective. Proc. of The 16th VLDB Conf., Brisbane, Austrilia, 1990.

2 王国仁，于戈等. CIMS 环境下多数据库的集成技术. 第2届中国计算机集成制造系统学术会议论文集，深圳，1992. 41～46.

3 Roth M A. Theory of non－first－normal from relational database. PH. D. Thesis, The University of Texas at Austin, 1986.

4 Roth M A *et al*. SQL/NF: a query language for ¬1NF relational language. Information Systems, 1991, **12**(1): 45～53.

5 Zheng Huaiyuan, Yu ge *et al*. Extending polygen paradigm with NF^2 data model to integrate multidatabase in CIMS environments. In: Proc. of International Conference on Computer Integrated Manufacturing(ICCIM'93), Beijing, 1993. 278～281.

6 张迎红. 多源数据库集成系统中集成机制的研究与实现[硕士论文]. 东北大学，1992.

7 王国仁，郑怀远. 基于 EER 数据库集成方法的研究. 计算机研究与发展，1993, **30**(12): 36～40.

8 王国仁等. 面向对象和关系数据库系统的集成方法. 见：何守才主编，第9届全国数据库学术会议论文集，上海：复旦大学出版社，1990. 291～299.

9 Deen S M *et al*. Data integration in distributed databases. IEEE Transtractions on Software Engineering, 1987, **SE－13**(7): 33～45.

10 于戈等. 一个 CIMS 信息集成平台系统的设计. 见：冯玉才主编，第12届全国数据库学术会议论文集，武汉：华中理工大学出版社，1994. 345～349.

11 寥卫东. 一个基于 NF^2 的多源数据库集成系统的研究与实践[硕士论文]. 东北大学，1993.

QUERY PROCESSING TECHNIQUES IN THE POLYBASE SYSTEM

Wang Guoren Yu Ge Shan Jidi Zheng Huaiyuan

(*Department of Computer Science and Technology Northeastern University Shenyang* 110006)

Abstract POLYBASE is an integrated multi database system. This paper mainly discusses the architecture and integration data model of the POLYBASE system, presents a query presentation model based on the concept of weighted query trees, designs and implements an effective navigative query processing algorithm on the basis of the concept of weighted trees.

Key words Multi database systems, integration data models, weighted query trees, query processing.