

MDBS 中基于模式映射树的查询分解和优化

李瑞轩 卢正鼎 肖卫军 王治纲

(华中科技大学计算机科学与技术学院)

摘要: 提出了一种多数据库系统的模式映射方法,并对水平映射、垂直映射、混合映射、相关映射和局部映射给出了一种统一的形式化描述.使用模式映射树存储和表达用于查询分解的全局模式以及全局模式到输出模式及局部模式的映射信息,使用查询树来表示多数据库查询的内部结构.通过合并全局查询树与模式映射树,给出了将全局查询转化为中间查询的全局查询分解方法.利用模式映射信息中的相关映射性质,给出了多数据库查询中全局连接的优化处理策略.

关键词: 多数据库系统; 查询分解; 查询优化; 模式映射

中图分类号: TP311.13 **文献标识码:** A **文章编号:** 1671-4512(2003)11-0022-03

1 模式映射

多数据库系统(MDBS)的模式结构决定了查询分解处理的流程^[1].现有的多数据库系统大多采用四级模式结构:局部模式、输出模式、全局模式和外模式^[2].多数据库用户只能访问全局模式,它的实际数据必须从各局部数据库系统中获得.全局模式包含一组全局类和模式映射信息,下面给出其相关定义.

定义 1 多数据库的类模式是一个多元组 $C(U, D, I, Q, M, P)$,其中, C 是类名; U 是组成 C 的有限属性集; D 是 U 中属性的值域; I 是类 C 的对象所响应的消息的集合; Q 是类 C 的对象所满足的限定条件集; M 是类 C 的模式映射信息的集合; P 是类 C 继承的父类的集合.类 C 有主键 $K \subseteq U$.

全局类模式描述了全局类的结构及语义约束,它可以按一定的条件 Q 转换成中间类模式.类是类模式在某一时刻的当前值.为了简单起见,本研究不讨论属性的值域、消息和继承情况,将类模式简化为 $C(U, Q, M)$,有时也用 $C(U, Q, M)$ 或类名 C 表示类.

定义 2 类 $C(U, Q, M)$ 模式映射 M 是全局类与中间类、中间类与局部类之间联系的集合.

模式映射描述了多数据库中全局类的对象最终是如何从局部数据库中获取数据的.它又可分为

为中间映射(EM)和局部映射(LM).其中,中间映射是全局类与中间类间联系的集合,局部映射是中间类与局部类之间联系的集合.

根据全局类与中间类之间的关系,中间映射可分为两大类:**a.** 基于类本身的映射,包括水平映射、垂直映射和混合映射;**b.** 基于类之间的关联的映射,又称为相关映射.

水平映射(HM)是全局类的对象按对象 ID 横向以某些条件映射成中间类的对象.记为 $C(HM)\langle P \rangle = \{C_1, C_2, \dots, C_n\}$,式中 $P = \{P_1, P_2, \dots, P_n\}$ 为一组给定的条件.

垂直映射(VM)是全局类的对象按类的属性纵向以属性组映射成中间类的对象,记为 $C(VM)\langle A \rangle = \{C_1, C_2, \dots, C_n\}$,式中 $A = \{A_1, A_2, \dots, A_n\}$ 为一组给定的属性.为了保证全局类的可重构性,应将键属性映射到各个中间类中.

混合映射(MM)是水平映射和垂直映射的混合操作,记为 $C(MM)\langle AP \rangle = \{C_1, C_2, \dots, C_n\}$,式中 $AP = \{\langle A_i, P_i \rangle \mid i = 1, 2, \dots, n\}$.视应用需要,可先水平映射后垂直映射,也可先垂直映射后水平映射.

相关映射(CM)是指一个全局类是根据另一个与其有关联性质的类的属性来进行映射的,该相关类已水平映射为 H_1, H_2, \dots, H_n .相关映射记为 $C(CM)\langle H \rangle = \{C_1, C_2, \dots, C_n\}$,式中 $H = \{H_1, H_2, \dots, H_n\}$.它是一种半连接操作.在多数据库查询处理中,经常用半连接操作以实现连接

收稿日期: 2003-01-13.

作者简介: 李瑞轩(1974-),男,博士研究生;武汉,华中科技大学计算机科学与技术学院(430074).

基金项目: 国家高性能计算基金资助项目(99319);国家“十五”科技攻关计划资助项目(2002BA103A04).

操作中操作数的缩减,减少连接和传输开销.

对于局部映射(LM),这里只考虑中间类到局部类的简单转换关系,如属性名、属性类型的转换等.其他复杂情况的处理方法同中间映射类似.

以上 5 种映射操作可用一个统一的形式描述如下: $C(O_j)\langle S \rangle = \{C_1, C_2, \dots, C_n\}$,式中, $O_j \in \{HM, VM, MM, CM, LM\}$; $S = \{S_1, S_2, \dots, S_n\}$.其中, a. 当 $O_j \in \{HM, VM, MM\}$ 时, $S_i = \{\langle A_i, P_i \rangle\}$ | 若 $O_j = HM$, 则 $A_i = U$, 若 $O_j = VM$, 则 $P_i = True$; b. 当 $O_j = CM$ 时, S_i 表示相关类; c. 当 $O_j = LM$ 时, $n = 1, S = S_1$ 表示转换函数 f .

操作的结果是将全局类或中间类 $C(U, Q, M)$ 按操作 O_j 分解成一组满足条件的子类 $C_1(U_1, Q_1, M_1), C_2(U_2, Q_2, M_2), \dots, C_n(U_n, Q_n, M_n)$, 式中 $U_i = A_i; Q_i = P_i; S_i = \emptyset; i = 1, 2, \dots, n$.

2 模式映射树与查询树

多数据库模式映射定义了全局类、中间类及局部类之间的关联,它主要用于多数据库全局查询的分解和优化处理.可以有多种方式来存储和表达模式映射信息.由于多数据库全局查询处理过程中需要将全局查询转换为查询树,为了便于模式映射信息在查询树中的使用,在此使用模式映射树来存储和管理多数据库模式映射,它存储了用于查询分解的模式信息.

定义 3 模式映射树是一棵树 $T = (V, E)$, 其中 V 和 E 含义如下.

A. V 是节点集.每个节点用 $C(U, Q, M)(O)$ 表示,其中 $C(U, Q, M)$ 表示类, O 表示该节点的操作,且 $O \in \{HM, VM, MM, CM, LM\}$.如果该节点是根,则 $Q = True, M = T$.

B. E 是边集.对于任意操作 O ,若有 $C(O)\langle S \rangle = \{C_1, C_2, \dots, C_n\}$,其中 $S = \{S_1, S_2, \dots, S_n\}, i = 1, 2, \dots, n$, 则:

a. 如果 $O \in \{HM, VM, MM, CM\}$, 那么, $S_i = \langle A_i, P_i \rangle$, 在模式映射树中有从节点 $C(U, Q, M)(O)$ 到 $C_i(U_i, Q_i, M_i)(O_i)$ 的边,且 $K_i = K, U_i = A_i, Q_i = Q \wedge P_i, M_i$ 是以 $C_i(U_i, Q_i, M_i)(O_i)$ 为根节点的子树;

b. 如果 $O = LM$, 那么, S_i 是转换函数 f , 在模式映射树中有从节点 $C(U, Q, M)(O)$ 到 $C_i(U_i, Q_i, M_i)(O_i)$ 的边,且 $K_1 = f(K); U_1 = f(U); Q_1 = f(Q); M_1 = \emptyset$.

从定义 3 可得,在模式映射树 T 中,对于任一节点 $C(U, Q, M)(O_j)$,如果它是根节点,那么它所表示的类一定是全局类,且 M 是整个模式映射树 $T, Q = True$;如果它是叶节点,那么它所表示的类一定是局部类,有 $M = \emptyset$;如果它既非根节点也非叶节点,那么它所表示的类一定是中间类,且 M 是以该节点为根的子树.

多数据库用户使用全局查询语言来表达全局查询,但要得到查询结果,必须对数据库中的类进行具体操作.为了将查询表达式转换成类的操作系列,使用查询树来表示查询的内部结构^[3].

定义 4 查询树是一棵树 $B = (V, E)$, 其中, a. V 是节点集,每个非叶节点是类操作符,叶节点是类名(即查询涉及的类); b. E 是边集,当且仅当 V_2 是 V_1 的操作分量,两节点有边 (V_1, V_2) .

例如,对于查询表达式 $E = \Pi_A(C_1 \cup (C_2 \circ C_3)) \cap \sigma_Q(C_1 \circ \Pi_B(C_2))$, 其查询树如图 1 所示.

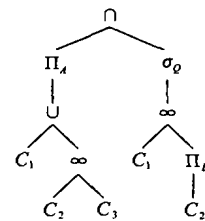


图 1 E 的查询树

3 全局查询分解

由于多数据库系统用户只能在全局类上完成查询,而其实际数据又必须从各个局部数据库中获得,在全局查询和局部查询之间还需要经过中间查询的转换.全局查询分解的过程实际上是利用模式映射信息将全局查询分解为中间查询,然后转换为相应的局部查询的过程,在分解的过程中逐步实现查询优化^[4].可按以下两步实现全局查询到中间查询的分解.

a. 全局查询转换成查询树.多数据库全局查询树和一般数据库的查询树类似,常用的逻辑优化技术在这里都可以适用.例如,对全局查询树中的一元操作尽量下移到叶节点,若查询树中有二元操作,则应尽量先缩减二元操作的操作数.若查询树同时具有选择和投影操作,则应尽量先做选择再做投影.

b. 全局查询树转化为中间查询树.由于模式映射树是由全局类经过中间映射操作形成的,是一组代数操作,全局查询树是对全局类的查询操

作,亦是一组代数操作,可以用下述算法进行合并,从而将全局查询树转化为中间查询树.

算法 1 全局查询树转化为中间查询树

输入:全局查询树和模式映射树.

输出:一组中间查询树.

a. 从模式映射树根节点开始,若节点上操作 $O_j = HM$ 或 CM ,则将其转换为并操作(\cup)节点;若节点上的操作 $O_j = VM$,则将其转换为连接操作(\bar{K})节点, K 为连接属性;若节点操作 $O_j = LM$,则不必转换. b. 按某一顺序遍历分解树,使用与第一步相同的方法处理其他节点. c. 输出对中间类的查询树.

在全局查询树转化为中间查询树的过程中,可以消去谓词运算具有矛盾的子树,即可消去选择操作结果为空的子查询树,也可以消去连接操作结果为空的子树.在得到了中间查询树后,还应按一般数据库查询优化技术进行优化,如尽量使一元操作下移,二元操作的操作数尽量减少,对同一中间类的多个选择、投影应合并成一个选择操作后接一个投影操作,尽量使查询树上相连的一元操作最多只有两个.

中间查询到局部查询的转换与全局查询到中间查询的转换类似,需要完成从中间类的类名、属性、关键字、类型等向局部类的转换,得到与中间查询树类似的局部查询树,再将局部查询树转化为各局部数据库能够接受的查询语言,并发送到局部数据库去执行.

4 全局连接的优化

在多数数据库查询处理过程中,当有两个全局类 C 和 S 进行连接时,对 C 和 S 的所有实例都应进行比较.当这两个全局类的中间类不在同一个场地上时,就必须经过通信在多数数据库中形成全局连接.图 2 是一全局连接的连接图,图中节点表示全局类经映射后得到的中间类,边表示两节点间连接不为空.图 2(a)是 C 和 S 的全连接图,即 C 的所有中间类(C_1, C_2, \dots, C_n)与 S 的所有中间类(S_1, S_2, \dots, S_m)进行完全连接.对于多数数据库来讲,这种连接的代价是极大的.所以,在考虑多数数据库的查询处理时,应充分利用模式映射信息对其进行优化.对于完全连接的优化处理有两种方法:a. 部分全局连接,如图 2(b),其中部分节点间没有连通,使完全连接图形成多个子图; b. 化简为简单全局连接,如图 2(c),每对节点间

只有一条边.

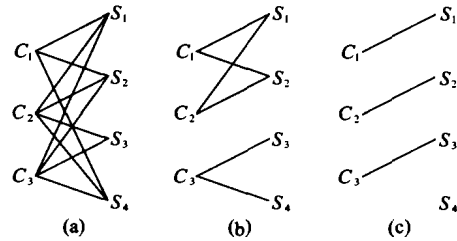


图 2 完全连接、部分全局连接和简单全局连接

多数数据库模式映射中支持的相关映射实际上是这种简单全局连接, C 和 S 的中间类之间具有相关映射关系,它们只存在一对一的连接,这就可以先做局部连接,再做全局连接,即先在中间类之间完成连接,然后再合并各局部连接的结果,这样可以大大降低通信开销.

例如,设有图 3(a)所示的查询树,其中虚线框表示由模式映射树转化而来的查询子树, S_1, S_2, S_3 是按 C_1, C_2, C_3 进行相关映射所得到的中间类,该查询树可以依等价转换规则转换成图 3(b),图 3(c)表示简单全局连接的查询树,和图 3(b)比较不同之处在于它先做连接操作再做并操作,有利于进一步优化查询树.在多数数据库查询中具有全局连接时,应将连接下属的并操作尽量上推.

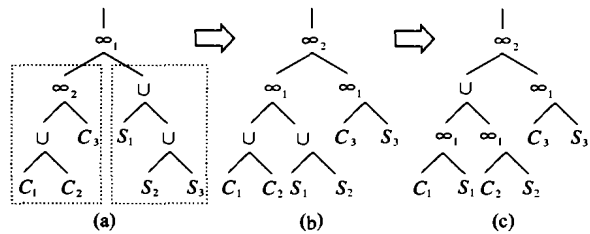


图 3 全局连接的优化处理

由于多数数据库环境的异构性和分布性以及各局部数据库系统的自治性,多数数据库系统中的查询分解和优化是十分复杂的.在多数数据库原型系统 Panorama 中对本文所给出的查询分解和优化方法进行了部分实现.

参 考 文 献

[1] Kevin C, Hector G. Boolean query mapping across heterogeneous information sources. IEEE Transactions on Knowledge and Data Engineering, 1996, 8(4): 515 ~ 521
[2] Reddy M P, Prasad B E, Reddy P G, et al. A methodology for integration of heterogeneous databases. IEEE Transactions on Knowledge and Data Engineering, 1994, 6(6): 920 ~ 933

(下转第 30 页)

张重复网页,因此集成搜索引擎在理想情况下将提供 219 张网页. Integration 综合考虑了两搜索引擎的排序结果,其结果具有比成员搜索引擎更好的相关性,很大程度上改善了搜索的查准率.

参 考 文 献

[1] 张晓辉,邵 华,常桂然. WWW 上的信息发现与搜索

引擎技术. 小型微型计算机系统, 1998, 19(6): 66~71

[2] Arasu A, Cho J, Hector G M, et al. Searching the Web. ACM Transactions on Internet Technology, 2001, 1(1): 2~43

[3] Lawrence S, Lee C. Context and page analysis for improved Web search. IEEE Internet Computing, 1998, 2(4): 38~46

Optimization of the ranking of the Web search results in meta-search engines

Li Yongping Wen Kunmei

Abstract: After full comprehension of the meaning of the relevancy, an improved ranking method was given based on the value. The need of the user, including interest value, people value and location value was considered and the pattern of the fixed page content was adopted. A small meta-search engine prototype was implemented. The experiment shows that the actual performance of this method is good and it also optimizes the ranked result.

Key words: meta-search engine; relevancy; optimal ranking; value

Li Yongping Dr.; College of Computer Sci. & Tech., Huazhong Univ. of Sci. & Tech., Wuhan 430074, China.

(上接第 24 页)

[3] 李瑞轩,卢正鼎, Sarem M 等. 多数据库查询处理中的查询树表示, 南京大学学报(自然科学版), 2001, 37: 174~178

[4] 卢正鼎,李兵,肖卫军等. 基于 CORBA/XML 的多数据库系统研究与实现, 计算机研究与发展, 2002, 39(4): 443~449

Query decomposition and optimization based on schema mapping tree in multidatabase systems

Li Ruixuan Lu Zhengding Xiao Weijun Wang Zhigang

Abstract: A method of schema mapping in multidatabase systems was introduced and an uniform formalized definition for horizontal, vertical, mixed, correlative and local mappings was given. The schema mapping tree was used to store and express global schemas and the mapping information from global schemas to export schemas and local schemas. The query tree was used to represent the internal structure of multidatabase queries. An approach of global query decomposition that the global queries were transformed into middle queries was presented through combining the global query tree with schema mapping tree. By employing the correlative mapping, the query optimization corresponding to global joins in multidatabase queries was also discussed.

Key words: multidatabase systems; query decomposition; query optimization; schema mapping

Li Ruixuan Doctoral Candidate; College of Computer Sci. & Tech., Huazhong Univ. of Sci. & Tech., Wuhan 430074, China.