

再探多连接查询优化方法^{*}

彭建平¹, 王变琴²

(1. 中山大学管理学院, 广东 广州 510275;
2. 中山大学信息科学与技术学院, 广东 广州 510275)

摘 要: 研究发现给定条件下的全局优化算法可用于具有非等连接的多连接查询中, 并且与 join 操作产生的关系大小的估算模型无关. 对于星状查询, 应用 join 产生关系的大小估算模型, 证明全局优化算法所生成的 join 序列具有任意 3 个相邻的 join 操作序列是最优的特性.

关键词: 关系数据库; 多元连接查询; 查询优化

中图分类号: TP311.132 **文献标识码:** A **文章编号:** 0529-6579 (2001) 02-0027-04

随着信息时代的来临, 数据库中的数据量急剧增大, 怎样迅速地从海量数据中挖掘出有用数据以满足查询的需求成为人们所关注的话题. 目前在关系数据库中, SPJ 查询是使用得最频繁的一种查询. 而查询优化的关键点之一都是围绕多元连接查询进行, 即: 确定 join 操作次序, 以使查询响应时间最短. 为此, 文 [1] 通过研究提出了一种能生成具有全局优化特征的 join 操作序列算法. 该优化算法 Determining sequence 所基于的条件是: ①等连接; ②文 [2] join 产生的关系大小估算模型; ③文 [2] 的 join 代价估算模型. 基于此优化算法, 对于多连接查询, 只要使用 join 代价估算模型或类似的估算模型, Determining sequence 仍能生成相应的全局优化 join 操作序列. 另外, 在文 [1] 的优化算法 Determining sequence 所基于的条件下, 对于多连接星状查询, Determining sequence 所产生的 join 序列具有任意 3 个相邻的 join 操作构成的操作序列是最优的特性. 使得算法 Determining sequence 具有更宽的使用范围.

1 问题描述

文 [1] 已对此做过详细的描述, 为便于阐述作者的新发现, 本文重新进行描述.

查询条件是由连接 2 个关系的 join 谓词项构成的, 其形式为: $\sigma_F(R_1 \times R_2 \cdots \times R_m)$ 其中, σ 表示选择操作 select, π_r 表示在 r 上的投影操作 project. 条件 F 是 $C_1 \wedge C_2 \wedge \cdots \wedge C_n$ 形式的合取公式. $C_i (i = 1, \cdots, n)$ 的形式为: $R_k.A$

comparison-op $R_j.A (k, j = 1, \cdots, m)$. 其中, comparison-op 是比较符 $=, >, <, >=, <=, <>$.

查询优化器所处理的查询由查询图表征, 查询图 $G = (V, E)$. 其中, V 是表征关系的结点集, E 是表征 2 个关系相连接的边集. 若在合取公式 F 中存在关于关系 R_i 和关系 R_k 的合取项, 则存在连接两结点 R_i 和 R_k 的边 (R_i, R_k) . 图 1a 和图 1b 均为查询图.

查询计算需要使用叉积时, 查询图存在多个连通分支. 根据文 [3, 4] 的处理, 可将叉积尽可能地推迟到各连通分支优化完毕后, 再将它们生成的结果叉积起来. 因此, 在本章中, 仅考虑查询图是连通图的情况.

若将边 (R_i, R_k) 从查询图 G 中删除, 并将结点 R_i 和 R_k 并为一个结点, 则称对图 G 就边 (R_i, R_k) 进行收缩操作. 对关系 R_i 和关系 R_k 进行 join 操作, 实质上等价于对图 G 就边 (R_i, R_k) 进行收缩操作. join 操作的结果关系记为 $R_i \theta R_k$. 由于 join 操作的交换性, 在此约定: 关系 $R_k \theta R_i$ 与关系 $R_i \theta R_k$ 表征同一个关系. 例如: 对于图 1a 中的边 (R_1, R_3) 进行收缩后, 图 1a 变成图 1b. 图 1b 中的 $R_1 \theta R_3$ 表征由关系 R_1 和关系 R_3 进行连接后产生的中间结果. $R_1 \theta R_3$ 与 $R_3 \theta R_1$ 表征同一个关系.

对于集合 K , 用 $|K|$ 表示集合 K 的基数. 对于关系 $R_i \theta R_k$, $|R_i \theta R_k|$ 表示关系 $R_i \theta R_k$ 的元组数. 对于关系 R_i , 用 $\|R_i\|$ 表示其大小. 如: $\|R_i \theta R_k\|$ 表示关系 $R_i \theta R_k$ 的大小.

尽管利用启发式方法和随机方法对多连接查

^{*} 收稿日期: 2000-06-27; 作者简介: 彭建平 (1961—), 男, 讲师.
(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

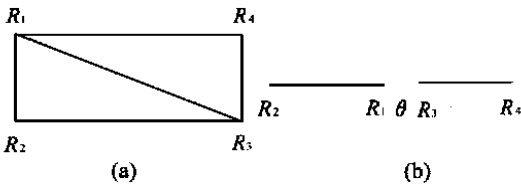


图 1 查询图

Fig. 1 Query graphs

询进行优化的问题得到了广泛的讨论,但在代价估算模型及其测试基准方面至今还缺乏统一的认识.随之带来的问题是在不同的前提假设下的实验分析结果的相互矛盾^[4,5].为承接文[2]的研究成果,本文将采用该文献中所使用的代价估算模型,讨论多连接查询的优化处理的问题.

不同的 join 执行算法具有不同的执行代价.这里采用的 join 执行方法为排序—归并算法.该算法具有稳定的性能特性,且能处理等连接 (equi-join) 和非等连接 (nonequi-join) 的查询.因此,该算法是当前数据库产品中使用比较频繁的算法^[3].用排序—归并算法对关系 R_i 和 R_k 进行连接的计算代价为^[3]: $|R_i| + |R_k| + |R_i \theta R_k|$.

该代价估算式是通过分析和简化文[6]的代价公式而获得,并在其多连接查询优化的分析研究工作中得到应用.

要说明的是:当关系 R_i 和关系 R_k 进行连接的计算代价为 $|R_i \theta R_k|$ 或 $\|R_i \theta R_k\|$ 或 $\|R_i\| + \|R_k\| + \|R_i \theta R_k\|$ 时,也不难获得与本文类似的结果和优化算法.对于多个 join 操作,若它们之间存在一个公共的关系 R ,则称它们是围绕关系 R 的相关 join 操作.例如,在图 1a 中,与 join 操作对应的 3 条边 (R_1, R_2), (R_1, R_3) 和 (R_1, R_4) 就分别表征了围绕关 R_1 的相关 join 操作 $R_1 \text{ join } R_2$ 、 $R_1 \text{ join } R_3$ 和 $R_1 \text{ join } R_4$.

2 BEFORE 关系和 EQUAL 关系

对不相关的 join 操作 $R_1 \text{ join } R_2$ 和 $R_3 \text{ join } R_4$, join 操作序列 $R_1 \text{ join } R_2, R_3 \text{ join } R_4$ 的计算代价显然与 join 操作序列 $R_3 \text{ join } R_4, R_1 \text{ join } R_2$ 的计算代价相同.因此, join 操作 $R_1 \text{ join } R_2$ 和 join 操作 $R_3 \text{ join } R_4$ 在执行次序上对总的查询执行代价无影响.其任一 join 操作序列均是最优的.

对于围绕关系 R 的 2 个相关 join 操作 $R_1 \text{ join } R$ 和 $R_2 \text{ join } R$,由于它们的执行次序要影响总的查询执行代价,因此,下面我们将讨论 join 操作

的执行次序与执行代价的关系,并由此确定它们的最佳执行次序.

定理 1 对于围绕关系 R 的 2 个相关 join 操作 $R_1 \text{ join } R$ 和 $R_2 \text{ join } R$, $R_1 \text{ join } R$ 先于 $R_2 \text{ join } R$ 执行所构成的 join 操作序列的总执行代价低于 $R_2 \text{ join } R$ 先于 $R_1 \text{ join } R$ 执行所构成的 join 操作序列的总执行代价的充要条件为: $|R_1 \theta R| < |R_2 \theta R|$.

证明 $R_1 \text{ join } R$ 先于 $R_2 \text{ join } R$ 执行的 join 操作序列为: $R_1 \text{ join } R, R_1 \theta R \text{ join } R_2$, 其执行代价和 $C1 = |R_1| + |R| + |R_1 \theta R| + |R_1 \theta R| + |R_2| + |R_1 \theta R \theta R_2|$;

$R_2 \text{ join } R$ 先于 $R_1 \text{ join } R$ 执行的 join 操作序列为: $R_2 \text{ join } R, R_2 \theta R \text{ join } R_1$, 其执行代价和 $C2 = |R_2| + |R| + |R_2 \theta R| + |R_2 \theta R| + |R_1| + |R_2 \theta R \theta R_1|$;

$C1 - C2 = 2(|R_1 \theta R| - |R_2 \theta R|) < 0 \Leftrightarrow R_1 \text{ join } R$ 先于 $R_2 \text{ join } R$ 执行所构成的 join 操作序列的总执行代价低于 $R_2 \text{ join } R$ 先于 $R_1 \text{ join } R$ 执行所构成的 join 操作序列的总执行代价.

定理 2 对于围绕关系 R 的 2 个相关 join 操作 $R_1 \text{ join } R$ 和 $R_2 \text{ join } R$, $R_1 \text{ join } R$ 先于 $R_2 \text{ join } R$ 执行所构成的 join 操作序列的总执行代价等于 $R_2 \text{ join } R$ 先于 $R_1 \text{ join } R$ 执行所构成的 join 操作序列的总执行代价的充要条件为:

$$|R_1 \theta R| = |R_2 \theta R|$$

(证明略).

定义 1 对于围绕关系 R 的 2 个相关 join 操作 $R_1 \text{ join } R$ 和 $R_2 \text{ join } R$, 若存在 $|R_1 \theta R| < |R_2 \theta R|$, 则 $R_1 \text{ join } R$ 必须在 $R_2 \text{ join } R$ 之前执行以获取最小的代价和, 记为: $R_1 \text{ join } R \text{ BEFORE } R_2 \text{ join } R$.

定义 2 对于围绕关系 R 的 2 个相关 join 操作 $R_1 \text{ join } R$ 和 $R_2 \text{ join } R$, 若存在 $|R_1 \theta R| = |R_2 \theta R|$, 则就获得最小代价和而言, $R_1 \text{ join } R$ 和 $R_2 \text{ join } R$ 的优先执行级别相等, 记为: $R_1 \text{ join } R \text{ EQUAL } R_2 \text{ join } R$.

关于 BEFORE 和 EQUAL 的传递关系, 有以下若干结论.

定理 3 对于围绕关系 R 的 3 个相关 join 操作 $R_1 \text{ join } R$ 、 $R_2 \text{ join } R$ 和 $R_3 \text{ join } R$, 若存在 $R_1 \text{ join } R \text{ BEFORE (或 EQUAL) } R_2 \text{ join } R$ 和 $R_2 \text{ join } R \text{ BEFORE (或 EQUAL) } R_3 \text{ join } R$, 则 $R_1 \text{ join } R \text{ BEFORE (或 EQUAL) } R_3 \text{ join } R$ (证明略).

定理 4 对于围绕关系 R 的 3 个相关 join 操作 $R_1 \text{ join } R$ 、 $R_2 \text{ join } R$ 和 $R_3 \text{ join } R$ ，若存在 $R_1 \text{ join } R \text{ BEFORE (或 EQUAL) } R_2 \text{ join } R$ 和 $R_2 \text{ join } R \text{ EQUAL (或 BEFORE) } R_3 \text{ join } R$ ，则 $R_1 \text{ join } R \text{ BEFORE (或 EQUAL) } R_3 \text{ join } R$ (证明略)。

推论 1 对于围绕关系 R 的 2 个相关 join 操作 $R_1 \text{ join } R$ 和 $R_2 \text{ join } R$ ，它们的 EQUAL 关系的等价类分别为 $[R_1 \text{ join } R]_{\text{EQUAL}}$ 和 $[R_2 \text{ join } R]_{\text{EQUAL}}$ 。若 $R_1 \text{ join } R \text{ BEFORE } R_2 \text{ join } R$ ，则对任意一个 join 操作 $R_{1i} \text{ join } R \in [R_1 \text{ join } R]_{\text{EQUAL}}$ 和任意一个 join 操作 $R_{2i} \text{ join } R \in [R_2 \text{ join } R]_{\text{EQUAL}}$ ，必存在 $R_{1i} \text{ join } R \text{ BEFORE } R_{2i} \text{ join } R$ 。

证明 对于任意一个 $R_{1i} \text{ join } R \in [R_1 \text{ join } R]_{\text{EQUAL}}$ ，有 $R_{1i} \text{ join } R \text{ EQUAL } R_1 \text{ join } R$ 。又因 $R_1 \text{ join } R \text{ BEFORE } R_2 \text{ join } R$ ，由定理 4 $\Rightarrow R_{1i} \text{ join } R \text{ BEFORE } R_2 \text{ join } R$ ；对于任意一个 $R_{2i} \text{ join } R \in [R_2 \text{ join } R]_{\text{EQUAL}}$ ，有 $R_{2i} \text{ join } R \text{ EQUAL } R_2 \text{ join } R$ ，由定理 4 $\Rightarrow R_{1i} \text{ join } R \text{ BEFORE } R_{2i} \text{ join } R$ 。

由于 EQUAL 和 BEFORE 关系的确立以及它们的传递性均与 join 操作所产生的关系的大小的估算模型和 join 操作的性质（等连接、非等连接）无关，而文 [1] 的优化算法 Determining sequence 只取决于 EQUAL 关系和 BEFORE 关系的传递性。因此，可以断定算法 Determining sequence 的优化条件只与 join 操作代价估算模型有关，而与 join 操作性质和 join 产生的关系的大小估算模型无关。

另外，算法 Determining sequence 在锁边时，有定理 5 来保证它不会在我们指出的更宽的条件下锁住查询图中所有的边，而导致算法 Determining sequence 不能正常工作。

定理 5 在 Determining sequence (G) 算法中，对图 G 的每一个结点 R 执行 $\text{set lock}(R)$ 不会导致图 G 中的所有边被锁住。

证明 在图 G 中任意选取一条边 (R_1, R_2) 。若 $(R_1, R_2). \text{lock} = \{R_1, R_2\}$ ，则可在 R_2 关联的边中找到一条边 (R_2, R_3) ，有 $(R_2, R_3). \text{lock} = \{R_3\}$ 或 $\{\}$ 。若为 $\{\}$ ，则存在没有锁住的边；若为 $\{R_3\}$ ，可按如下方法继续搜索：

- ① $i \leftarrow 3$;
- ② 搜索关联于结点 R_i 的边 (R_{i+1}, R_i) ， R_i 对边 (R_{i+1}, R_i) 没有上锁或 (R_{i+1}, R_i) 是自由边；
- ③ 若 (R_{i+1}, R_i) 是自由边，搜索结束；若 $(R_{i+1}, R_i). \text{lock} = \{R_{i+1}\}$ ，则 $i \leftarrow i + 1$ ，转第 ②

步。

上述搜索过程不会陷入无限循环之中，它会找到自由边而终止搜索。因为：由于图 G 中的结点数是有限的，因此，搜索要进入无限循环的条件是：结点 $R_3, R_4, \dots, R_n$ 必构成回路(如图 2)，且 $(R_3, R_4). \text{lock} = \{R_4\}$ ， $(R_4, R_5). \text{lock} = \{R_5\}$ ， $\dots$ ， $(R_{n-1}, R_n). \text{lock} = \{R_n\}$ ， $(R_n, R_3). \text{lock} = \{R_3\}$ 。由该条件，可获得不等式组：

$$\begin{cases} |R_3 \theta R_4| > |R_4 \theta R_5| \\ |R_4 \theta R_5| > |R_5 \theta R_6| \\ \vdots \\ |R_{n-1} \theta R_n| > |R_n \theta R_3| \\ |R_n \theta R_3| > |R_3 \theta R_4| \end{cases}$$

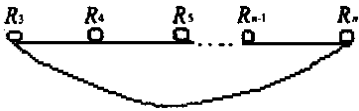


图 2 循环搜索示意图

Fig. 2 Circular searching sketch map

由上面的不等式组，可推得不可能成立的不等式 $|R_3 \theta R_4| > |R_3 \theta R_4|$ 。因此，无限循环的条件不可能成立。Q. E. D.

从以上证明可以发现证明过程与所产生的关系的大小估算模型无关，而结果与文 [1] 的定理 5 相一致。因此，可得出结论：算法 Determining sequence 在锁边时，条件进一步放宽，也不会死锁。

3 星状查询的全局优化特性

本小节将证明：对于星状查询，算法 Determining sequence 能生成满足如下全局优化特性的 join 操作序列：在该序列中，任意 3 个相邻的 join 操作构成的子序列是最优的。

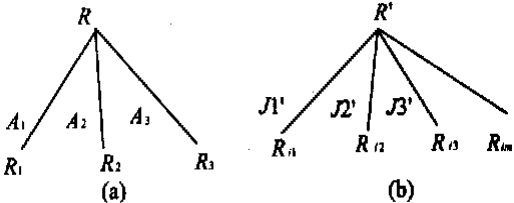


图 3 星状查询图

Fig. 3 Star query graph

引理 对于围绕关系 R 的 3 个相关 join 操作 $R_1 \text{ join } R$ 、 $R_2 \text{ join } R$ 和 $R_3 \text{ join } R$ (图 3a)，若算

法 Determining sequence 产生的 join 操作序列为 $R_1 \overset{A_1}{\theta} R \overset{A_1 A_2}{\theta} R_2 \overset{A_1 A_2}{\theta} R_2 \text{ join } R_3$, 则该序列是最优的 (证明略)。

定理 6 用算法 Determining sequence 为星状查询确定的 join 操作序列具有如下全局优化特征: 在该序列中, 任意 3 个相邻的 join 操作构成的子序列是最优的。

证明 对于任意一个星状查询图, 在对其所拥有的任意一条边进行收缩后所产生的查询图仍然是星状查询图。因此, 在算法 Determining sequence 构造出来的 join 操作序列中, 任意相邻 join 操作构成的子序列 J_1, J_2, J_3 必定是算法 Determining sequence 从由围绕某个关系 R' 的 3 个相关 join 操作 J_1', J_2' 和 J_3' 构造而来的 (图 3b)。由于 join 操作 J_1', J_2', J_3' 构成的查询图是星状查询图, 所以算法 Determining sequence 确定的 join 操作序列 J_1, J_2, J_3 是最优的。
Q. E. D.

本文针对文 [1] 提出的算法 Determining sequence 及使用条件, 通过研究发现: 算法 Determining sequence 既可用于等连接查询的优化又可用于非等连接的多连接查询的优化中。该优化算法与 join 操作所产生关系的大小的估算模型无关, 仅与 join 操作的计算代价的估算模型有关。

因此, 算法 Determining sequence 的约束条件可进一步放宽, 其使用范围更广。对于星状连接查询, 在文 [1] 研究优化算法的前提下, 本文得到: 算法 Determining sequence 可生成具有如下全局优化特征的 join 操作序列: 任意 3 个相邻的 join 操作构成子序列是最优的。

参考文献:

- [1] 钟武, 胡守仁. 一种改进的多连接查询优化方法 [J]. 软件学报, 1998, 9(2): 125—128.
- [2] CHEN M S, YU P S. Optimization of parallel execution for multi-join queries [J]. IEEE Trans Knowledge and Data Eng, 1996, 8(3): 416—428.
- [3] SWAMI A. Optimization of large join queries: Combining heuristic and combinatorial techniques [J]. Proc ACM SIGMOD Conf Management of Data, 1989, 367—376.
- [4] SWAMI A, GUPTA A. Optimization of large join queries [J]. Proc ACM SIGMOD Conf Management of Data, 1988, 8—17.
- [5] IOANNIDIS Y E, KANG Y C. Randomized algorithms for optimizing large join queries [J]. Proc ACM SIGMOD Conf Management of Data, 1990, 312—321.
- [6] IYER B, DIAS D M. System issues in parallel sorting for database system [J]. IEEE Proc 6th int Conf Data Eng, 1990, 246—255.

A Research on Optimization of Multi join Queries

PENG Jian ping¹, WANG Bian qin²

(1. School of Management, Zhongshan University, Guangzhou 510275, China;

2. School of Information and Technology, Zhongshan University, Guangzhou 510275, China)

Abstract: On the basis of the given condition, it is shown that the global optimization algorithm can apply to equi-join or nonequi join and has nothing to do with the estimation model of the join later size in muti join queries, and that for star queries, the algorithm can produce join sequences with the following global optimization property. In the join sequence, arbitrary subsequence formed by three neighboring joins is optimum.

Keywords: relational database; multi join queries; query optimization