

并行数据库复杂关系查询优化的一种方法

洪晓光 王新军 董继润

(山东大学计算机系 济南 250100)

摘要 查询优化不仅是顺序数据库系统的重要组成部分,也是并行数据库的重要组成部分,而多个 Join 操作的复杂关系数据库的查询优化又是目前研究的主要课题。本文给出了一种通过修改、更新 Join 图的查询优化方法。

关键词 并行数据库 查询优化 连接

TP311.13

并行数据库查询优化问题与顺序数据库不同。在顺序数据库系统中,给定一个查询 Q,查询优化算法只需要找到 Q 的一个具有最小工作量的执行计划,这样的计划必然具有最快的响应时间,因为在具有单处理器的系统中,一个计算任务的响应时间与这个任务的工作量成正比。在分布式数据库系统中,虽然它与并行数据库系统有许多相似之处,但其查询优化目标主要是减少结点之间的通信费用,局部处理时间则不是主要的优化目标。在并行数据库中,由于数据传递已不是大问题,并且并行处理器数量也可以达到很多,因此其查询优化目标则是对给定的查询 Q 寻找具有最小响应时间的计划,而执行计划的工作量则不是最重要的。

对于一个复杂关系查询(Complex Multi-Join Query,简称 CMJQ),在并行环境中就意味着 Q 中不同的操作可以由不同的处理器簇执行,即不仅每个 Join 可以由多个处理器操作,而且多个 Join 也可以并行执行。因此 CMJQ 的不同 Join 序列将导致 CPU 和 I/O 处理费用的不同。这样 CMJQ 的查询优化主要有两个方面:连接序列的计划模式和处理器分配方案。

连接方法能影响查询的优化过程,采用 hash 方法和采用 sort-merge 方法,其优化过程不同,我们在这里重点讨论如何构造一个连接序列和对这个序列如何分配处理器,因此忽略 Join 的实际连接方法。

一、基本概念和假设

首先我们假设所采用的并行环境是 SN 结构,每个处理器都有各自的磁盘存储器和主存储器,它通过最小化共享资源来避免系统因竞争资源带来的干扰,其 SN 结构又具有很高的可扩充性。本文的方

法就是基于 SN 结构的。

1990 年, Schneder 等研究了查询树模型,提出了左线性树(left-deep trees),右线性树(right-deep trees)和浓密树(busy trees)的概念。浓密树连接序列由于以前受到并行处理中 CPU 个数及处理能力的限制并没有引起很多注意,现在这两个因素已不再是障碍,因此浓密树的并行查询处理已引起很大的注意。文[1]中给出了一种执行最小结果关系连接形成浓密树的启发式方法 GMR:

Scheme GMR(G)

```
begin
  1. repeat until |V|=1
  2. begin
  3. 从图 G=(V,E)中选择连接  $R_i \bowtie R_j$ ,使得
      $|R_i \bowtie R_j| = \min_{i,j \in V} \{|R_i \bowtie R_j|\}$ .
  4. 执行  $R_i \bowtie R_j$ .
  5. 合并  $R_i$  和  $R_j$  生成  $R_{\text{sub}(i,j)}$ . 做相应修改.
  6. end
end
```

GMR 的时间复杂度是 $O(|V||E|) < O(|V|^2)$, 接近于 $O(|V|^2)$ 。

这只是对连接序列的一种计划模式,象前面提到的那样,为了减少 CMJQ 的执行时间,还必须合理分配给每个连接相应数量的处理器,并且它们之间的关系可以象文[3,4]中提到的曲线那样(见图 1)。从图中可以看出,大致上增加参与连接的处理器将减少执行时间,但存在一个临界点,达到这一点,再增加处理器相反会增加执行时间。

文[2]中给出了以下三种启发式处理器分配原则:

(a) 顺序执行原则(SE). 给每个连接分配所有处

理器,连接顺序执行,这样的设计就使得连接序列变得十分重要。

(b)最小时间点原则(MT)。这种方式基于操作曲线上的最小时间点选择,见图(1)中 P_M 。

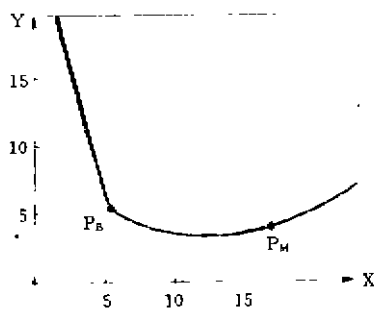


图1.

(c)最佳效率点原则(TE)。这种方式基于可独立执行连接的最佳效率点,见图(1)中 P_B 。

(b),(c)都有固有的缺陷。对于(b),有些连接操作在最佳效率点可能由于给的处理器个数太少而花费很长时间,而(c)中又可能使某些处理器不能充分利用。因此(b)(c)的组合则是一种有效的处理器分配方法,换句话说,处理器分配的个数是 $\alpha * P_M + (1 - \alpha) * P_B$,其中 $0 \leq \alpha \leq 1$ 。

上面叙述了目前有关CMJQ中连接序列产生和处理器分配的一些方法,下面,笔者将给出一种更有效的连接序列产生原则和处理器分配方法。

二、Merge-Update 方法

前面叙述的GMR算法是在连接序列中每次执行两个关系的连接使结果关系最小的方式,很明显它每次只执行两个关系的连接,但在实际应用中,我们知道,查询时经常会出现一个关系R与多个关系进行连接的情况,这时若仍然采用GMR方法效率就会降低。为此我们在复杂查询图的基础上给出一种产生连接序列的方法,并同时给出其处理器分配原则。首先我们给出产生连接序列的规则,然后根据规则得出相应算法。

设给定查询Q,Q所涉及的数据库为D,由此我们给出查询图 $G=(V,E)$,其中 $V=\{Q\}$ 所涉及的所有关系名的集合, $E=\{(i,j) | i \neq j, Q\}$ 的某个操作同时涉及 R_i 和 R_j 这里我们假设操作是自然连接。

首先我们给出Merge-Update方法的基本思想。它是基于分而治之的策略,通过修改、合并查询图和

动态分配处理器的方法来完成查询的,其过程如下:

1. 对给定的查询Q,在Q中选择一个关系 R_i ,把它从Q中删除,当然选择关系有一定的准则。删除后查询Q被分解为 Q_1, Q_2, \dots, Q_k ;

2. 递归并行执行这k个子查询,方法如1中描述的那样;

3. 由2得到k个结果关系 $R_{E1}, R_{E2}, \dots, R_{Ek}$,然后完成 R_i 与这k个结果关系的连接。

上述过程有很多细节问题没有说明,象(1)中如何选择关系 R_i ,处理器如何动态分配等。这些问题都直接影响算法的性质,下面我们就来讨论这些问题。

首先我们讨论选择关系 R_i 的标准。设从Q中除去 R_i ,剩余的查询变成 Q' ,从并行的角度,也就是从减少响应时间的角度来考虑, R_i 有两种选择方案:(a)Q被分解的部分越多,就越能提高并行处理的响应时间。(b)选择基数最大的关系作为 R_i ,这样可以使参与运算的数据量变得很少,除去 R_i 后剩余的结果关系较小。

现在来看一下(a)方案的正确性。假定关系 R_i 被删除后Q被分解成的独立部分为 Q_1, Q_2, \dots, Q_M ,而任一关系 R_j 被删除后Q被分解为 Q'_1, Q'_2, \dots, Q'_N ,这里 $M > N$,我们只需证明 R_i 的响应时间比 R_j 的响应时间短。由于 $M > N$,因此M个子查询中的每一个数据处理量就其平均来说比N个子查询中的每一个要小(也不排除M中某些要比N中的大,对此我们可以通过动态分配处理器的方法加以解决)。由于处理器个数没有限制,它采用前面提到的公式 $\alpha * P_M + (1 - \alpha) * P_B$,因此很显然处理M个子查询的时间要比处理N个子查询的时间短。(b)方案则类似于前面我们提到的GMR方法,在此不多赘述。

现在我们来查看有关处理器的动态分配问题。在方案(a)和(b)中,根据选择的关系 R_i 和被分解成的子查询 Q_1, Q_2, \dots, Q_k 估计 Q_1, Q_2, \dots, Q_k 的数据量,该量可以通过 Q_i 中参与运算的关系的数目和大小得出,根据这些参数将处理器分为K簇,每簇处理器负责处理相应的一个 Q_i ,这里我们并不要求每簇处理完的时间大约在同一个时间点上,即不要求同步,因为每个 Q_i 的结果关系与 R_i 的连接我们可以采用流水线方式,只要某个 Q_i 处理完成,其结果关系 R_{Ei} 便马上参与和 R_i 的连接,连接出来的行元组又马上可以与相继完成的 Q_j 的结果关系 R_{Ej} 进行连接,这里,完成一个连接的处理器个数我们限制在 $\alpha * P_M$ 。

小波变换与图像数据压缩评价^{*}

方涛 郭达志

(中国矿业大学测量与土地科学系 徐州 221008)

摘要 In this paper, sub-band bite allocation in the wavelet transform domain is studied, this posses important guidance meanings for data compression of vector quantization (VQ) based on wavelet transform. Unlike Fourier analysis, there are different wavelet bases in wavelet analysis. Especially, orthonormal bases of compactly supported wavelets are a important kind of wavelet bases. Consequently, the performances of data compression about those wavelet bases are analysed and evaluated.

关键词 Wavelet transform, Bite allocation, Orthonormal wavelet bases, Compactly supported wavelets, Data Compression.

小波变换, 图像数据压缩

小波变换与 JPEG 国际标准所采用的 DCT 相比, DCT 在高频和低频段具有相同的空间分辨率, 而小波变换在低频时具有较高的频率分辨率, 高频时具有较高的空间分辨率, 有助于实现图像中平稳

成分与非平稳成分的分离, 易于保护图像的细节和边缘, DCT 采用分块压缩编码, 缺乏整体性, 易于出现方块效应, 而小波变换对图像作全局分解, 具有整体连续性, 量化失真随机分布于整幅图像之中, 不易

*国家自然科学基金资助项目。方涛 博士后。郭达志 教授。

图像处理

TN 911.1 TN 919.8

$+(1-\alpha) * P_2$.

以上是该算法的基本思想, 下面我们给出它的一个形式化描述:

Procedure Merge-Update(Q, C, R_k)

// Q 为查询, C 为处理器个数, R_k 为结果关系 //

begin

1. select R_k // 采用方案(a)或(b) //
2. 将 R_k 从 Q 中删除
3. 查询 Q 变为 Q₁, Q₂, ..., Q_k
4. 并行执行 Merge-Update(Q_i, C_i, R_k)
5. for I=1 to k do
6. 完成 R_k 与 select 的连接(用流水线方式)

end

上述 Merge-Update 算法中, select 可以用方案(a)也可以用方案(b), 两种方案的时间复杂度均为 O(n), 而选择 R_k 与 select 连接中, 选择 R_k 的时间复杂度为 O(k) < O(n)。很显然 Merge-Update 算法的时间复杂度是多项式的。

讨论 到目前为止, 还没有一个并行查询优化算法

能够保证产生高效率的查询计划, 因此我们还需做以下几个方面的工作: (a) 具有强表达能力的并行查询计划表示模型的研究; (b) 并行数据操作算法的多变量复杂性模型的研究; (c) 以并行数据操作算法的多变量复杂性模型为基础的查询计划的复杂性模型的研究; (d) 具有产生高效率查询计划能力的并行查询优化算法的研究。

参考文献

- [1] M. S Chen et al., Scheduling and Processor Allocation for Parallel Execution of Multi-Join Queries, Proc. of the 8th Intl. Conf. on Data Eng., Feb., 1992
- [2] P. S Yu et al., Parallel Query Processing, Advanced Database System, p. 229-258, 1993
- [3] 李建中, 并行数据操作算法和查询优化技术, 软件学报, No10, 1994
- [4] 陈建荣、严隽永、叶天荣, 分布式数据库设计导论, 清华大学出版社, 1992