分布式数据库中半连接查询优化算法的改进①

Improvement of Semi-Join Algorithm in Distributed Database

张　伟　刘万军　(辽宁工程技术大学 电子与信息工程系 辽宁 葫芦岛 125105)

摘　要： 提出了一种新的查询优化算法“二次半连接算法”。通过对“二次半连接算法”的阐述和性能分析，证明了它能最大限度地缩减数据传输量，利用多站点并行性缩短查询响应时间，最大限度的降低系统的开销，特别是在广域网上的分布式数据库查询中，该算法有较明显的性能优势和可用性。

关键词： 分布式数据库 半连接算法 二次半连接算法 并行性 响应时间

1　序言

分布式数据库查询涉及到大量的数据在网络上的移动，再加之在远程通信网络中各站点之间数据传输速度比单机情况下，内存与磁盘间的数据传输速度要慢许多倍。在这种情况下查询的局部处理时间与通信所需时间相比，可以忽略不计，减少通信费用成为分布式查询优化的主要目标，而通信费用与所传输的数据量成正比。由于上述原因，为了缩短响应时间和减少系统开销，一种基本的方法就是采用半连接来缩减它的操作数，以降低通信费用和缩短响应时间[1]。

半连接算法虽然能缩减它的操作数，但是没考虑到查询结果的最终存放位置，而且不能达到很好利用网络各分布站点的并行性。当结果数据很大且需要传输时，这样的问题就会很明显。本文提出一种新的算法“二次半连接算法”来解决上述问题。

2　半连接算法

2.1 半连接(semi—join)算法

半连接(semi—join)算法的基本思想：是经过半连接操作减少操作关系，从而减少远程数据查询的数据传输费用和通信时间。

假定有两个关系 R(如表 1 所示)和 S(如表 2 所示)，求按属性条件 R.A=S.B 的半连接操作。这里“关系”也可指片段。因为在集中式数据库中，关系是一张二维表。而在分布式数据库中，关系一般被称为全局关系。当全局关系存放到各站点时，需要对它进行分片成若干互不相交的子集，每个子集为全局关系的一个逻辑片段，简称片段。这些片段也是一张二维表。采用半连接操作可表示为：

$$R \propto_{A=B} S = \pi_R(R \infty_{A=B} S) = R \infty_{A=B}(\pi_B(S))$$ [2] (1)

或 $$S \propto_{A=B} R = \pi_S(S \infty_{A=B} R) = S \infty_{A=B}(\pi_A(R))$$ [2](2)

其中符号 $\propto$，∞ 分别为两个关系(或片段)半连接和连接操作符号，π 为关系的投影，A，B 为两关系(或片段)的连接属性。在上式(1)和(2)中，等式左边是两关系半连接的公式表达形式，等式中间和右边的表达式为其结果的等价表示形式，可以用它们的关系运算求解过程，求解出半连接的结果，其表达式的意义和求解过程与集中式数据库的关系运算相同。

表 1　关系 R

C	A
c1	x1
c2	x1
c3	x1
c3	x3
c4	x4
c4	x3

表 2　关系 S

B	D
x1	d1
x2	d2
x5	d2
x5	d3
x6	d4
x7	d5
x8	d5

由上面的半连接操作公式可得 $R \propto_{A=B} S$ 的结果(如表 3 所示，R' 即 $R \propto_{A=B} S$)和 $S \propto_{A=B} R$ 的结果(如表 4 所示，S' 即 $S \propto_{A=B} R$)。由表 3 和表 4 所示，可知半连接操作具有不对称性，即：$R \propto S \neq S \propto R$ [2]。

① 收稿时间:2008-12-24

表 3 关系

C	A
c1	x1
c2	x1
c3	x1

表 4 关系 S'

B	D
x1	d1

若采用半连接方法表示这两个关系 R 和 S，在属性 R.A 和 S.B 上做连接操作，则有：

$R\infty_{A=B}S=(R\propto_{A=B}S)\infty_{A=B}S$ [1]

或 $S\infty_{A=B}R=(S\propto R)\infty_{A=B}R$ [1]

其中，等式左边表示两个关系的连接操作，等式右边括号内为“半连接”，其操作目的是先对一个关系进行缩减，即减少关系元组数。然后进行两个关系的连接，半连接操作具有不对称性。所以两关系半连接表达式有以上两种表达形式。但结果相等(如表 5 所示)

表 5 $R\infty_{A=B}S$ 或 $S\infty_{A=B}R$

C	A	B	D
c1	x1	x1	d1
c2	x1	x1	d1
c3	x1	x1	d1

当连接操作采用半连接方法表示时，由(1)和(2)代换也可以得到下面的表达式：

$R\infty_{A=B}S=(R\propto_{A=B}S)\infty_{A=B}S=(R\infty_{A=B}\pi_B(S))\infty_{A=B}S$ [1]

或 $S\infty_{A=B}R=(S\propto_{A=B}R)\infty_{A=B}R=(S\infty_{A=B}\pi_A(R))\infty_{A=B}R$ [1]

采用半连接方法表示连接操作过程(如图 1 所示)。

（1）$\pi_B(S)$　（2）发送 $\pi_B(S)$ 到站点 A

（3）$R'=R\infty_{A=B}\pi_B(S)$　（4）发送 R' 到站点 B　（5）$R'\infty_{A=B}S$

图 1　半连接算法连接过程

2.2 半连接算法连接过程

R 与 S 的半连接算法连接过程如下：

① 在站点 B 上,作关系 S 在 R 和 S 连接属性 B 上的投影 $\pi_B(S)$。当然也可以在站点 A 作关系 R 在 R 和 S 连接属性 A 上的投影,一般最好选投影结果少的，这样有利于减少网络上传输的数据量。

$\pi_B(S)$ 结果为：(x1,x2,x5,x6,x7,x8)。

② 把 $\pi_B(S)$ 结果(x1,x2,x5,x6,x7,x8)送到站点 A。

③ 在站点 A 上根据收到的投影值计算半连接,其结果为 R' (如表 3 所示)。

④ 把 R' 从站点 A 上送到站点 B。

⑤ 在站点 B 上执行 $R'\infty_{A=B}S$ 连接操作,形成最终结果集(如表 5 所示)。

从上面分析半连接算法连接过程中，可以看到半连接算法最终结果集(即表 5)放在其中一个站点上,它没有考虑到实际应用中的情况。在一些情况下,请求站点不是结果集存放的站点。这对于查询请求有大量结果集的情况下，将存在网络上有的结点上有大量数据传输而拥塞，有的没有数据传送。这将造成网络负载不均与并行性较差的问题。从表 5 中，还可以进一步看到：A，B，D 三个属性存在大量的数据冗余，这使网络上传输的数据量进一步增大。

3 二次半连接算法

半连接算法在上面所述的情况中，它存在不足。本文对半连接的算法进行改进，进行二次半连接来进一步减少通信数据量，同时也提高了各站点并行的可能性。二次半连接算法如图 2 所示：

（1）$\pi_B(S)$　（2）发送 $\pi_B(S)$ 到站点 A

（3）$R'=R\infty_{A=B}\pi_B(S)$　（4）$\pi_A(R')$

（5）发送 $\pi_A(R')$ 到站点 B

（6）在 B 站点执行 $\pi_A(R')\infty_{A=B}S$

图 2　二次半连接算法连接过程

3.1 二次半连接算法连接过程

R 与 S 的二次半连接算法连接过程如下：

① 在站点 B 上，作关系 S 在 R 和 S 连接属性 B 上的投影 $\pi_B(S)$。$\pi_B(S)$ 结果为：(x1,x2,x5,x6,x7,x8)。

② 把 $\pi_B(S)$ 结果(x1,x2,x5,x6,x7,x8)送到站点 A。

③ 在站点 A 上根据收到的投影值计算半连接,其结果为 R'(如表 3 所示)。

④ 计算 R'的投影 $\pi_A(R')$。

$\pi_A(R')$ 结果为：(x1)

⑤ 发送 $\pi_A(R')$ 结果：（x1）发送到站点 B。

⑥ 在 B 站点上执行 $\pi_A(R')\infty_{A=B}S$ 连接操作,得到结果 S'(如表 4 所示)。最终结果集(如表 5 所示)在请求站点形成。

注意：与半连接算法连接过程相比，二次半连接算法连接过程在请求站点上要多了一次连接过程。为了减少请求站点上连接开销，最好按 A、B 属性排序，并且在 A、B 属性所在站点也排序或索引。它有利于连接，特别是在向请求站点传输数据时，按 A，B 属性顺序传输可大大节省请求站点的连接时间。

3.2 二次半连接算法优点

二次半连接算法优点：查询请求对结果集的处理分为浏览操作、更新操作、删除操作。二次半连接算法在经过两次半连接后,结果集的处理按下列情况分类：

① 浏览操作

首先它减少了半连接算法第 4 步不必要的数据传输量(尤其数据量大时更明显)。这些数据在传输到请求站点(不是半连接结果集形成的站点)将会被第二次传输加大了网络传输数据开销。

其次将所得到的查询结果传送到请求站点，因为数据按某要求排好顺序进行传输，简单的连接(不必自然连接)即可满足操作的要求(此时要将两个或多个站点上所有重复的连接属性值去掉,最后只保留其中的一个属性值)。这里最重要的是两个或多个站点可同时向目标站点传送数据。当数据量较大时，两次半连接有利于网络的负载均衡和站点利用率。

② 更新操作

根据所要更新的属性值，请求站点可分别将任务分派到不同的站点上并行执行，而不是先访问半连接的结果站点，再对相关站点进行更新。例如：对其中一个站点的属性进行更新，只需要更新需更新站点上的数据。当两个或两个以上站点都进行更新,只需要同时向各站点发出数据更新的命令，各站点可以完全地并行执行更新。

③ 删除操作

对不同节点同时发出删除命令,并在各个节点上并行执行。二次半连接的这个优点在网络质量不是很好的情况下，缩短响应时间方面的优点很明显。

4 算法性能比较与分析

在分布式数据库系统中，一个查询可能涉及多个站点。通常以两种不同的目标考虑查询优化：一种目标是总传输代价最小，另一种目标是以每个查询的响应时间最短。在一般远程网络分布式查询环境中，查询的局部处理时间与通信所需时间相比，可以忽略不计，减少通信费用成为分布式查询优化的主要目标。从减少通信费用和缩短响应时间方面比较两种算法的性能。传输代价模型可表示成一个线性函数，表示 X 字节的传输时间与数据的数量成线性关系可用下面关系式表示：

$$T = T0 + T1 * X \quad [2] \tag{3}$$

在半连接中传输的数据量 X：

$$X = SIZE(B) * VAL(B) + SIZE(R) * CARD(R') + SIZE(R\infty S) * CARD(R\infty S) \tag{4}$$

在二次半连接中传输的数据量 X：

$$X = SIZE(B) * VAL(B) + SIZE(A) * VAL(A) + \text{SIZE(R)*CARD(R')} + \text{SIZE(S)*CARD(S')} \tag{5}$$

其中，T 为数据传输时间，T0 为建立一个连接所需的固定时间，T1 是网络范围内的统一的传输速率，X 表示数据传输量， A，B 为连接属性，R，S 为连接关系，VAL 为所求属性上不同值的个数，SIZE 为属性或关系的字节数，CARD 为无重复的元组个数。

下面以一个简单的查询图来比较一下它们的数据传输和响应时间(如图 3)。

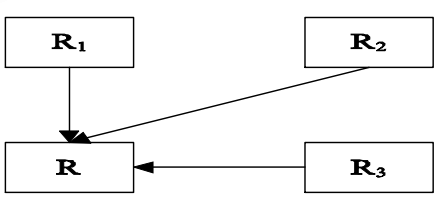

图 3　4 个关系的查询图

假定 R 是有 10^6 元组其中有 10^5 符合条件，其中 R1，R2，R3 都与 R 进行连接，符合条件元组数都为 10^4，并且每条记录的长度都为 100 字节，连接属性 10 字节且连接属性个数大约都是 10^4(箭头方向为结果集站点)。

① 半连接算法：

总数据传输量：

$X = SIZE(B)*VAL(B) + SIZE(R)*CARD(R')$

$+SIZE(R\infty S)*CARD(R\infty S)$ $=3*10*10^4+3*10^4*100+3*190*10^5$;

(注：190 是去掉共用属性，10^5;是结果数目)

结果集传输时间：

$T = T0 + T1*X = T0+3*190*10^5*T$

② 二次半连接算法：

总数据传输量：

$X = SIZE(B)*VAL(B) + SIZE(A)*VAL(A)$

$+SIZE(R)*CARD(R') + SIZE(S)*CARD(S')$

$=6*10*10^4+3*10^4*100+100*10^5$

(注：6 是有二次半连接)

R 结果集传输时间：

$T' = T0 + T1*X = T0+100*10^5*T$

R1 结果集传输时间：

$T1' = T0 + T1*X = T0+100*10^4*T$

R2 结果集传输时间

$T2' = T0 + T1*X = T0+100*10^4*T$

R3 结果集传输时间：

$T3' = T0 + T1*X = T0+100*10^4*T$

结果集传输时间：

$T = MAX(T', T1', T2', T3')$

从以上表达式，可以看出半连接算法数据传输量比较大和响应时间性能比不上二次半连接算法。特别是对于大型分布式数据库系统，响应时间性能上和数据传输量上会更明显。分布式数据库一个很大的特点就是可以用数据分散和冗余的特性进行数据的并行处理。这样可以提高整个系统数据处理的速度和效率。但是半连接算法在半连接处理后，它的结果集不能充分利用分布式数据库的特点进行并行处理，这使响应时间和数据传输等方面受到较大的影响。

5 结语

本文在半连接算法基础上提出了一种新的查询处理优化算法——二次半连接算法。二次半连接算法通过最大限度地缩减网络上的数据传输量，利用多站点并行性缩短查询响应时间，最大限度的降低系统开销。特别是在广域网上的分布式数据库系统查询中，该算法有较明显的性能优势和可用性。

参考文献

1 邵佩英.分布式数据库系统及其应用(第二版).北京:科学出版社, 2005.

2 Tamer M. zsu, Patrick Valduriez 分布式数据库系统原理(第 2 版).北京:清华大学出版社, 2002.

3 张时鹏,陶世群.大规模数据库的一种新的分布式查询优化算法:二分劈开缩减.计算机工程与设计, 1998, 19(4):62－65.

4 王意洁,王勇军,卢锡城.基于半连接的并行查询处理算法的研究.软件学报, 2001,12(2):219－224.

5 徐勋明,薛永生,王劲波,吕晓华.一种基于事先测试的分布式数据库优化联接查询技术.厦门大学学报, 2004,43(2):175－178.