

汇总型多表连接查询的一种优化方法

杨 武 (重庆工业管理学院信息管理系 400050)

文守逊 (重庆大学工商管理学院 400050)

摘要:通过对影响多表连接查询效率因素的分析,在通常的临时关系法的基础上,提出了一种针对汇总型多表连接查询的优化方法,充分利用这种方法能提高汇总型查询的执行效率。

关键词:数据库 SQL 查询 优化

1. 多表连接对查询效率的影响

在开发数据库应用时,经常要进行多表连接(JOIN)查询(QUERY)获取需要的信息。目前大多数关系数据库管理系统(RDBMS)完成连接都采用基于成本的查询优化策略,即用穷举法列出查询的所有执行策略,考虑所有可能的连接顺序,再在其中选择估计成本最低的策略。由于查询执行策略的数目随参与连接维数的增加而成指数增长,尽管采用相关算法(如启发式算法)进行搜索,但选择最优者的开销也随之急剧增长。因此,多表连接往往对查询效率有重大影响。

2. 提高多表连接处理效率的临时关系法

从数据库实现连接的方式可以看出,常见的二表连接对性能的影响不大,只有当参与连接的表的数目达到三个以上、且表中有大量数据时,这种连接就可能成为影响性能的重要因素。即使最后结果的规模不大,其内部产生的中间结果的数量和占用内存的规模都相当可观,足以形成性能的“瓶颈”。在进行数据库设计时,应预先考虑到连接操作对系统性能的影响,通过适当增加表的冗余项的方法来减少连接操作。在编写应用程序时,可以通过不断优化查询语句等方法提高连接效率。目前,优化查询语句最常用的方法是临时关系法。临时关系法是将两个表连接到一个临时表中,然后再把该临时表与其他被连接表中的某个表连接并生成另一临时表,依次类推,最后把所有要连接的表连接到某个临时表中,这样就实现了将多表连接转化为多次两个表的连接,大大提高了处理效率。临时表的引入,使原来的一条 SQL 语句变成了多条顺序执行的 SQL 语句。

3. 汇总型多表连接查询优化方法

数据库应用经常需要产生大量的统计报表,统计报表的数据通常来自多个基本表,这些基本表中的记录数往往又很多,为避免基本表的直接连接对效率的影响,于是使用临时关系法,尽量减少参与连接的表的数量,以加

快报表生成速度。

由于报表的主要功能是统计,所以在报表生成中用到的查询多为带有 GROUP BY 子句的 MAX, MIN, SUM, AVG 等汇总型查询。如果将汇总操作尽量放在临时关系法中前面的 SQL 语句中执行,可较大地减少用于下一步连接的临时表中的记录数量,这就是本文提出的汇总型多表连接查询优化方法的基本思路。下面以一个实际例子来详细说明。

假设某医院管理系统的数据库中有三个表,分别存放药品、科室和药品消耗等信息如下:

Medicine(Medicine_No, Medicine_Name, Category, Base_Cost)

Dept(Dept_No, Dept_Name, Dept_Type)

Cousume(Medicine_No, Dept_No, Month, Year, Amount)

假设现在要统计过去2年中医院外科类科室抗菌素类药物的用量,写出的直接实现的 SQL 语句如下:

```
SELECT Year, Sum(Amount)
FROM Medicine, Dept, Cousume
WHERE Dept. Dept_No = Cousume. Dept_No
AND
```

```
Consume. Year >= '1997' AND Medicine. Category = '抗菌素' AND Dept. Type = '外'
```

若使用临时关系法分解多表连接,则该查询可分为两步实现,记为:

```
INSERT INTO Temp(Year, Amount, Dept_No)
Select Year, Sum(Amount), Dept_No
From Consume, Medicine
WHERE Consume. Year >= '1997' AND Medicine.
Category = '抗菌素' 语句1
AND Consume. Medicine_No = Medicine. Medicine_No
```

```
SELECT Year,Sum(Amount)
FROM Temp,Dept
WHERE Temp. Dept _ No = Dept. Dept _ No AND
Dept. Type= '外' 语句 2
GROUP BY Year
如果按照本文提出的汇总型多表连接查询优化方法
将汇总操作提前,则生成同样的查询使用的 SQL 语句
是:
```

```
Insert Into Temp(Year,Amount,Dept _ No)
Select Year,Sum(Amount),Dept _ No
From Consume,Medince
Where Consume. year >= '1997'and Medince. Cate-
gory= '抗菌素' AND 语句 1
Consumes. Medince _ No= Medince. Medince _ No
Group By Year,Dept _ No
SELECT Year,Sum(Amount)
From Temp,Dept
Where Temp. Dept _ No= Dept. Dept _ No AND Dept.
Type= '外'
语句 2
Group By Year
```

在依照本文方法生成的查询 SQL 中,第一条 SQL 语句已将抗菌素用量按科室和年份进行了分组,因此用于第二条 SQL 连接的临时表 Temp 中的记录较临时关系法已大为减少。这样第一步的汇总就可能大幅度减小后期连接的输入规模,而中间结果的规模往往是衡量执行时间的重要标志。可以看出本文方法的目的就是通过将汇总尽量提前,以求改善查询性能。

4. 执行效率分析

为了比较 2 中同一查询的三种实现方法的运行效果,按实际数据的规模,用仿真数据对分别对三种实现方法进行了测试。测试在客户机/服务器环境下进行,客户机是 Compaq MMX166(内存 32M),服务器是 HP NET-SERVER LDPRO 奔腾 180MHZ(内存 64M), DBMS 采用微软 SQLSERVER6.5,网络为 10M 以太网,结果如下表:

Medince 记录数	Comsume 记录数	Dept 记录数	直接实现 执行时间(秒)
10,000	1000	100	17
20,000	1,0000	100	33
500,000	2,0000	100	95

临时关系法(语句 1+语句 2) 执行时间(秒)	本文方法(语句 1+语句 2) 执行时间(秒)
16 + 2	17 + 1
23 + 3	24 + 2
66 + 17	68 + 7

注意,Medince 表中的记录数是指类别为抗菌素的药品数量,Comsume 表中的记录数是指类别为抗菌素的药品消耗统计记录数。

从测试结果可以看出,当参与连接的数据量比较小时,三种方法实现的速度基本同,其中直接实现由于 SQL 语句少速度还略快一点,而随着参与连接的表规模的增大,临时关系法和本文方法相对于直接实现的优势也逐渐趋于明显。

比较临时关系法和本文方法的数据还可以看出,两种方法的第一步实现耗时虽然大致相同,本文方法的第二步实现,由于连接规模的缩小而获得的速度上的优势却在随着数据量的增加逐步加大。考虑到一张报表中往往有多个统计项目来自于同一个关系,这些项目可以通过临时关系法和本文方法得到的中间关系与其他关系进行连接和投影操作及进一步的汇总来实现,即中间关系具有一次生成,多次使用的特点,所以在报表生成中充分利用本方法,可以改善系统性能。

5. 结语

通过分析 RDBMS 中多表连接的实现方法,指出影响多表连接查询效率的两个主要因素是参与连接表的数目和参与连接表的规模。然后介绍了提高多表连接查询效率的成效显著的临时关系法。并针对汇总型查询特点,在临时关系法的基础上提出了汇总型多表连接查询的一种优化方法,阐述了基本原理和实现方法,并通过实验说明了它可提高汇总型查询的执行效率。

参考文献

[1] Rick Snodgrass. Querying valid - time state tables . Database Programming & Design, 1998, Vol 11(7): 6062

[2] Stephen Brobst and Bob Vechinone. Starburst grows bright. Database Programming & Design, 1998, Vol 11(2): 4449

[3] WWW.Sybase.com.cn/database.html

(来稿时间:1999 年 7 月)