

基于共享排序实现 CUBE 操作的一个优化算法

陈志伟 励晓健 李伟生

TP3 A

摘要 CUBE 操作在数据仓库/联机分析处理中是一个关键性的操作, 它需要计算一个属性列表上的所有可能的组合, 并等价于标准 SQL 中一组 group by 子句。作者分析了 CUBE 操作的几种实现方法, 并扩展了一种以 group by 子句共享排序为基础实现 CUBE 操作的优化算法, 在应用中有较好的效果。

关键词 数据仓库 联机分析处理 数据立方体 CUBE 操作

Optimized Share-Sorts Based CUBE Algorithm

Chen Zhiwei Li Xiaojian Li Weisheng

(Northern Jiaotong University, Beijing, 100044)

Abstract: The CUBE operator is one of most important operators in the data warehouse and OLAP. It requires computing group-bys on all possible combinations of a list of attributes, and is equivalent to the union of a number of standard group-by operations. This paper analyses several methods which can compute the CUBE operator and extends share-sorts based algorithms with optimizations which have a good effect in the application.

Keywords: Data warehouse, OLAP, Data Cube, Cube operator

1 引言

在数据仓库 (Data Warehouse) 和联机分析处理 (OLAP) 中, 数据立方体 (Data Cube) 是核心的概念。从逻辑上讲, 数据仓库是一个多维数据库, 但目前的产品实现多以关系数据库为基础, 将星型模式或雪花模式来表示多维。OLAP 以多维分析为基础, 重在对数据进行多层次、多角度的分析, 以满足管理决策的需要。

2 Group by 子句与 CUBE 操作

对于数据仓库和 OLAP, 数据在逻辑上是用多维的

方式来组织和处理的, 在多维模式中的一些属性可以作为对数据对象性质观察的角度, 称为维 (dimension), 反映数据对象特性的属性称为指标 (measure), 这样的结构称为数据立方体 (Data Cube)^[1]。

标准 SQL 中的 group by 子句 (称为分组子句) 常用于在一个属性集合上的聚集 (如: 求和)。OLAP 常在不同维上进行聚集计算, 为使交互式分析成为可能 (反映时间在几秒内), 我们可以预先计算一些维上的聚集, 这样查询时, 就能够直接使用聚集数据而不需要从原始数据计算。大量的数据与计算使得聚集计算很耗时。为了满足对数据立方体的这种普通且必须的计算, 提出了 CUBE 操作符。

陈志伟 北方交通大学计算机学院 在读硕士研究生 100044 北京市
励晓健 北方交通大学计算机学院 在读硕士研究生 100044 北京市
李伟生 北方交通大学计算机学院 教授 100044 北京市

内可能要做几次, 利用语声应答系统记录接线表代替这一单调乏味的工作将显得很有必要。它的优点: a. 通常接线指令都是些简单的命令, 前后没有关系, 如线的颜色, 线长, 起始端和终了端, 因此不需要柔和连贯的语音; b. 指令可以由一个相对来说较小的词汇表构成——对于设备的某些特定部件来说, 50 个左右的单字就够了; c. 接线表一般都用计算机设计——因此它们所用的形式通常适用于数字语声应答系统; d. 接线指令经常要做修改, 因此, 使用语声应答系统可以简化更换接线表引起的繁杂任务。

4.2 语声应答在信息检索系统中的应用

在为通信设备接线的语声应答系统的应用中, 可能的使用者与构成输出消息的系统之间基本上没有相互作用。这是因为语声应答系统通常的按键电话输入被计算机的卡片预消机所代替, 在卡片上记有所需的输出消息。若把语声应答系统应用到查号辅助、信贷查询以及编目控制等问题时, 基本前提是语声应答系统能够对信息的数据库进行存取。这样, 它才能找出所需的信息并组成适当的消息送到用户。语声应答系统的研究与应用必能给铁路的信息事业带来勃勃生机。

(收稿日期 2001-03-20)

CUBE操作符是一个group by子句的n维一般化^[1]。它对所有可能的维组合进行group by计算。显然，n维的数据立方体上所有的聚集有^[2]种。一个聚集也称为数据立方体的一个节点(cuboid)。在所有维上的聚集称为基节点(base cuboid)。对于一个4维的数据立方体，若维表示为A, B, C, D，则基节点的group by子句为group by A, B, C, D。为表示方便，可以用聚集所用的维来表示节点，如节点(A, B)指group by A, B。

3 CUBE操作的实现与优化

3.1 独立计算法

这是一种直接计算的方法。每个节点都要读基节点，从基节点计算本节点。这样，对于一个n维的数据立方体，CUBE操作要对基节点读(2n-1)次并计算，所以，本算法简单但性能较差。

3.2 双亲法

本方法在文献[2]提出，假设在4维{A, B, C, D}数据立方体上进行CUBE操作，节点(A, C)可以从节点(A, B, C)或是节点(A, C, D)中生成。一般来说，在属性集X上的节点(称为节点X)可从节点Y上生成，当X属于Y。这时，节点Y称为节点X的双亲，一种显然的优化方法是选择最小的Y，称为最小双亲法。因为属性集中的属性越多，节点就越大，所以从k个属性的节点生成k-1个属性的节点，可使节点的双亲变小。

整个CUBE生成可用一个DAG图来表示。图中的顶点(node)就是聚集的节点，每条从k个属性的节点i到k-1个属性的节点j的边，表示节点j可从节点i生成，且节点j的属性是节点i的属性的子集。

双亲法也是一种比较成熟的算法，实现也比较简单，其性能比独立计算法要好的多。目前的数据仓库与OLAP产品中的CUBE计算通常采用以上两种算法。

3.3 共享排序法

本算法在文献^[3]中提出，仅对于以排序为基础生成group by的算法，目的在于多个group by共享排序的结果，既通过以特定次序排序的数据来计算以这个次序为前缀的所有group by。例如：将原始数据以ABCD排序，则可计算group by ABCD, group by ABC, group by AB, group by A，而不用另外的排序。这种算法的特点是对计算单个group by子句的算法进行了修改，使得一些group by子句的计算在底层上得到了优化。

3.4 缓存优化

这种优化方法通过缓存group by的结果，使其导出的group by子句减少磁盘i/o操作。如：若节点(A, B, C)已计算，如节点(A, B, C)在内存，可直接计算节点(A, B)、节点(A, C)、节点(B, C)。这种方法，侧重于通过减少磁盘i/o这个费时操作来提高整个算法的性能。

4 CUBE算法

4.1 搜索格^[4]算法输入，定义如下：表示CUBE生成的DAG图扩展可形成搜索格，K层表示K个属性的节点，关键字all表示空，记为0层。每条边eij有两个权值，S(eij)指当节点i需要排序时，由节点i生成节点j的权值，A(eij)指节点i无需排序时由节点i生成节点j的权值。根据最小双亲法，可以将S(eij)和A(eij)定义为节点i大小的增函数。当然，对任一条边eij，都有A(eij) < S(eij)。

4.2 O是搜索格的一个子图算法输出，每个group by只有一个双亲，并有一个属性次序(以这个次序排序)相连。若节点j的属性次序是它双亲节点i的前缀次序，则节点j能从节点i计算，无需排序。节点i图中eij边标志为A，权值A(eij)，否则，节点i需要排序，边权值S(eij)很清楚，对任一节点，至多有一个出边标志为A，因为相邻层只有一个i的前缀，标为S的边可有多条。算法的目的是发现O有最小的权值和。

4.3 本算法逐层执行处理过程，从第k=0层到k=N-1层。N是属性总数，对第k层，要发现一个最好路线从k+1层去计算k层，可以用二部图^[5]的问题来解决，方法如下：

首先，将原始搜索格的第k+1层变形，每个节点作k个附加拷贝。每个复制节点仍连到原节点连到的节点上。从原始节点i到节点j的权值为A(eij)，而所有复制节点到节点j的权值都为S(eij)。然后我们可以找到最小匹配的二部图，每个k层的节点h将匹配到k+1层的顶点g。

4.4 演示

图1(a)三个属性的搜索格，先对层2每个节点作一个拷贝。实边表示A()边，虚边S()边，每个节点下边的数字为出度权值。图1(b)为最小权值匹配所得到的解。节点(A)连到节点(A, B)，边S()，节点(B)到节点(A, B)，边A()，因此，层2的节点(A, B)在计算时，必须以属性BA为次序来排序，以保证计算节点(B)时，无需排序。同样，节点(C)到节点(A, C)，边A()，计算节点(A, C)时，必须以属性CA为次序来排序。节点(B, C)没有匹配，则节点(B, C)可用任意次序排序计算。我们用^[5]中的算法来发现最小匹配的二部图。算法的时间复杂性O(((k+1)M[k+1])3)，M[k+1]是k+1层节点的数目。

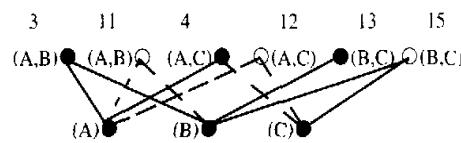


图1(a) 三个属性搜索格

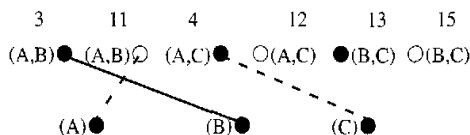


图1(b) 最小权值匹配解

4.5 输入含边权值 $A()$, $S()$ 的搜索格混合 CUBE 算法:

```

For level k=0 to N-1
  { // 从层 k+1 生成层 k
    对层 k+1 的每个节点，做 k 个拷贝，每个拷贝与原
    节点连接同样的节点；

    将 A (e[ij]) 赋给从原节点发出的边，S (e[ij]) 赋
    给从拷贝节点发出的边；
```

在变形层发现最小权值匹配的二部图:
 For each group by g in level k+1
 { 根据与 g 相连的 A () 边, 确定 g 的排序次序;
 例子: 我们用本算法来解一个四维的搜索格. 图2中, 每个 group by 下的数字是 A () 与 S () 权值. 实边是 A (), 虚边 S (). 图2是用本算法得到的结果.

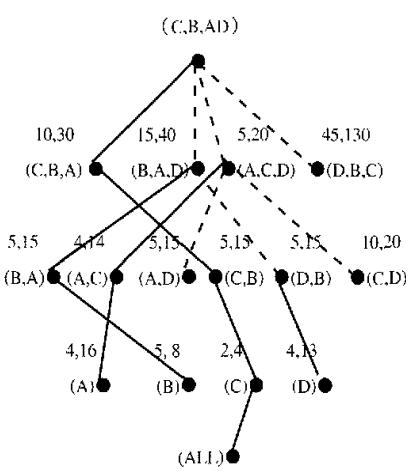


图2 四维搜索格

4.6 在执行结果子图时，加入缓存优化。例如对图2的执行，节点(C,B,A,D)排序后，在内存中可依次计算节点(C,B,A)，节点(C,B)，节点(C)，然后，以次序BADC将基节点排序，在内存中可依次计算节点(B,A,D)，节点(B,A)，节点(B)缓存优化。

4.7 对于数据立方体来说,一般情况下,维数很小,但数据量极大。CUBE操作的绝大部分时间用于group by子句,本算法能动态选择一个最优的双亲节点,使group by的效率大大提高。相对group by子句的计算,产生子

图的时间在整个CUBE操作中基本可以忽略不计,故不再进行时间与空间复杂性分析算法评价。

5 结束语

本文的混合算法结合了几种现有操作的优点,Gray J提出的CUBE操作是数据立方体的基本操作,对于拥有海量数据的数据仓库与OLAP来说,CUBE操作是一个极费时的操作,所以,研究更好的算法有很大意义,有较好的性能。

6 参考文献

- 1 裴健, 柴玮等, 联机分析处理数据立方体代数.软件学报, 1999(6)

2 Gray J, Chaudhuri S et. *Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub_totals*. *Data Mining and Knowledge Discovery*. 1997, 1(1):29-53

3 Sunila sarawagi, Rakshs Agrawal et. *On computing the data cube*. Research Report RJ 100026, IBM Almaden Research Center, 1996
<http://www.almaden.ibm.com/cs/quest>

4 Venky harinarayan, Anand Rajaraman and Jeff Ullman. *Implementing Data Cubes Efficiently*. In Proc. of the 1996 ACM-SIGMOD Conference, 1996

5 刘尉敏, 数据结构. 北京: 清华大学出版社, 1999

(收稿日期2000-11-15)

信息 · · · · ·

以以太网交换机为基准的测试工具

W&G公司宣布已开发出被称为STBench™的用于以太网交换机测试的应用软件。该软件使W&G的DA-30互连网分析仪能在交换机高业务量实时环境下进行测试。它的面世，使交换机生产商拥有了可靠的测试工具。

产品部经理凯文·葛里克说：“W&G公司由于最早开发出名为RTVench的应用软件而被广大路由器生产商所熟知，该软件为路由器测试提供了工业标准。现在，随着STBench的开发成功，W&G更确立了其在网间连接设备测试的领先地位。

用了STBench软件的DA-30平台为研制开发者和生产厂商提供了更先进的互连网分析仪。W&G提供全套互连网分析解决方案,包括用于现场测试的Domino系列互连网分析仪和RDA-3000远程检测系统。

徐榮字