

# ROLAP 环境下数据立方体的计算框架

李盛恩 张绍兰 巩建国

(山东建筑工程学院计算机系 济南250014)

**摘要** 数据立方体计算是联机分析中一项重要的技术。研究工作者提出了多种具有不同存储空间和查询相应时间的数据立方体,每种数据立方体有各自的生成算法。本文分析了使用关系系统作为存储结构的一般数据立方体、部分数据立方体和浓缩数据立方体的原理,提出用合作伙伴的概念统一这三类数据立方体的思想,并设计了一个算法 TCUBE 用于生成这些数据立方体。我们使用了一个实际数据集测试了 TCUBE 的性能,结果表明它生成浓缩数据立方体的速度要快于原有的算法。

**关键词** 数据仓库,联机分析,数据立方体

## A Framework of Data Cube Computation in ROLAP

LI Sheng-En ZHANG Shao-Lan GONG Jian-Guo

(Department of Computer, Shandong Institute of Architecture and Engineering, Jinan 250014)

**Abstract** The computation of data cube is one important technique in On-Line Analytical Processing. Researchers have proposed many kinds of data cubes that are of different query response time and occupy varying size of space. Any more, each data cube has its own constructing algorithm. This paper analyzes the principles of normal data cube, partial data cube and condensed data cube that put their tuples into relation system, proposes to use the idea of fellowship to unify these kinds of data cube, and designs an algorithm TCUBE to obtain them. We also conduct an experiment using a real data set to verify the performance of TCUBE. The results show that TCUBE outperforms the original algorithms used to produce condensed cube.

**Keywords** Data warehouse, OLAP, Data cube

## 1 前言

数据立方体(Data Cube)<sup>[1]</sup>是联机分析中一个重要的操作符。给定基本关系  $R(A_1, A_2, \dots, A_n, M)$ , 由  $R$  产生的数据立方体是一组由 SPJ-GROUP-BY 语句定义的实例化视图, 因为  $R$  有  $n$  个维属性 ( $A_i, 1 \leq i \leq n$ ), 共有  $2^n$  种组合, 所以数据立方体中有  $2^n$  个实例化视图(包括  $R$ )。

随着基本关系中维属性和元组数量的增加,数据立方体的体积和计算时间急剧增加,无法适应实际应用的需要。为了克服这个问题,研究工作者提出了一些其它种类的数据立方体和生成算法。如部分数据立方体:只实例化部分视图和这些视图中的所有元组<sup>[2,3]</sup>;浓缩数据立方体:实例化所有视图和视图中的部分元组<sup>[4-7]</sup>;近似数据立方体:采用某些近似计算技术,减少数据立方体的体积和查询时间,但是,查询结果有一定的误差<sup>[8]</sup>;冰山(IceBerg)数据立方体:施加某些限制条件以减少元组个数<sup>[9]</sup>。为了区别起见,我们把文[1]中提出的数据立方体叫做一般数据立方体,它实例化所有的视图和视图中的所有元组。每种数据立方体有独特的优点,一个理想的联机分析系统应该具有生成这些数据立方体的能力,以供用户根据自己的需要选用符合实际要求的数据立方体。

数据立方体可以选用不同的存储结构。例如,采用多维数组作为存储结构<sup>[10]</sup>,这类联机分析系统叫做 MOLAP;把元组存放到关系系统中,一般的是把一个视图中的元组存放到一个关系,不同的视图存放到不同的关系中,称这类联机分析系统为 ROLAP<sup>[1-6]</sup>;也可以存放到特殊的数据结构中<sup>[6,7]</sup>。由于关系系统技术成熟,伸缩性好,适合管理海量数据,因此,ROLAP 有很强的生命力。

目前,不同的数据立方体需要不同的生成算法,实现起来比较复杂。本文提出了合作伙伴的概念,在此基础上设计了一个算法,可以用来计算采用关系系统作为存储结构的一般数据立方体、部分数据立方体、浓缩数据立方体。

## 2 基本思想

为了能用一个算法计算三类数据立方体,首先需要分析这些数据立方体的生成算法和查询机制,找到共同规律。

**定义1** 假设  $u, v$  是数据立方体中任意实例化视图,如果存在元组  $t \in u, t$  需要从  $v$  的某个或某些元组中计算得出,则称  $v$  是  $u$  的合作伙伴。用  $P(u)$  表示  $u$  的所有合作伙伴。

下面我们分析文[1~6]提出的算法,从中可以发现,如果摒弃算法的细节问题,从更高的层次上看,这些算法的基本思想都可以使用合作伙伴的概念加以解释,因此可以给出生成这些数据立方体的单一程序。

一般数据立方体需要实例化所有的视图和所有的元组。当查找数据立方体中的一个元组时,首先要确定它所属的视图,然后对存储该视图的关系采用顺序扫描或使用索引找到这个元组。所以,对任何一个视图  $v$  有  $P(v) = \{v\}$ ,即要实例化每个视图中的所有元组,当查询  $v$  中的元组时,只需要到存储  $v$  的关系中查找。

部分数据立方体只实例化部分视图和这些视图中的所有元组。给定了存储空间上限  $S$ ,使用线性查询代价模型,利用贪心求解算法求出要实例化的视图集合  $M$ ,使得数据立方体占用的空间小于  $S$ ,同时,平均查找时间最少。当查询元组  $t$  时,首先确定  $t$  所属的视图  $v$ ,如果  $v \in M$ ,则在  $v$  中查找  $t$ ,否则,在  $M$  中找到视图  $u, v \subset u$  (即  $u$  包含  $v$  的所有属性),如果

存在多个视图满足条件,则选择元组数最少的视图作为  $u$ 。然后,用  $u$  中的某些元组计算出  $t$ 。因此,对任意的视图  $v$ ,有  $P(v) = \{w\}$ 。如果  $v \in M$ ,则  $w = v$ ,否则  $w = u$  并且  $v < u$ 。

可以使用关系系统作为存储结构的浓缩数据立方体有两类,即 condensed cube<sup>[4]</sup>和 quotient cube<sup>[5,6]</sup>,它们利用 Cover Partition<sup>[5]</sup>原理来减少数据立方体的体积。Cover Partition 在数据立方体中定义一个等价关系  $\equiv_{cov}$ ,对于任意的两个元组,  $t_1, t_2, t_1 \equiv_{cov} t_2$ ,当且仅当  $t_1$  和  $t_2$  是从基本关系  $R$  中的相同元组集聚集得到的。等价关系  $\equiv_{cov}$  将数据立方体中的元组划分成了若干个等价类,等价类中的所有元组对任何聚集函数都有相同的函数值,每个等价类用一个元组 upper bound<sup>[5]</sup>表示,在本文中我们把它叫做封闭元组,下面我们根据文[5]给出它的定义。

**定义2** 给定基本关系  $R(A_1, A_2, \dots, A_n, M), t = (a_1, a_2, \dots, a_n, m)$  是由  $R$  生成的数据立方体中的一个元组,  $a_i \in DOM(A_i) \cup \{All\}$  ( $All$  是文[1]中定义的一个特殊值),  $BTS(t)$  是基本关系  $R$  的子集,  $t$  是从  $BTS(t)$  集聚集得到的。如果,对于任何的  $a_i, a_i = All, BTS(t)$  在维  $A_i$  上的投影多于一个元组,则  $t$  是一个封闭元组。

quotient cube 只存储封闭元组。当查找一个元组  $t$  时,如果  $t$  是一个封闭元组,则可以在相应的视图  $v$  中找到它,否则,要找到  $t$  所在等价类的封闭元组  $t'$ ,因为  $t'$  与  $t$  有相同的聚集函数值。根据封闭元组的性质,  $t'$  一定存在于视图  $u$  中,  $v < u$ 。因此,  $P(v) = \{v, u_1, u_2, \dots, u_k\}, v < u_i$ ,即只实例化  $v$  中的封闭元组,  $v$  中的非封闭元组被其它视图中的封闭元组所封闭。

condensed cube 提出了基本单元组的概念:如果基本关系  $R$  中的元组  $t$  在  $SD$  上的投影具有唯一性,则称  $t$  是  $SD$  上的基本单元组( $SD$  是某些维属性的集合)。如果  $t$  是  $SD_1$  上的基本单元组,并且  $SD_1 \subseteq SD_2$ ,则  $t$  也是  $SD_2$  上的基本单元组。显然,基本单元组就是基本关系  $R$  中的封闭元组。分析 condensed cube 的生成算法 MinCube<sup>[4]</sup>可以发现,在计算任何一个视图  $v$  中的元组时,首先从  $R$  中选择候选元组,这些元组不是视图  $u$  包含的维属性集合上的基本单元组,  $u < v$ ,当一个候选元组是视图  $v$  包含的维属性集合上的基本单元组时,则不在  $v$  中实例化该元组。用封闭元组的术语来说,就是如果  $v$  的元组  $t$  被基本关系  $R$  中某个封闭元组所封闭,则不实例化  $t$ ,所以,  $P(v) = \{v, R\}$ 。当查找  $v$  中某个元组时,首先到  $v$  中查找,如果没有找到,再到  $R$  中查找。

通过上面的分析,可以用合作伙伴来描述一般数据立方体、部分数据立方体和浓缩数据立方体,在具体计算时还需要借助于封闭元组的概念。

### 3 算法

我们在 BUC<sup>[9]</sup>算法的基础上设计了算法 TCUBE,因为 BUC 是生成一般数据立方体的一个比较好的算法,特别适用于稀疏数据集(而实际应用中遇到的数据多是稀疏的),同时也非常适合于应用合作伙伴和封闭元组的概念。

算法 TCUBE 如图1所示。程序有两个输入参数:基本关系  $R$  和合作伙伴表  $P$ 。为了方便起见,我们用一维数组表示  $P$ 。对于一般数据立方体,  $P[i] = 1, 0 \leq i \leq 2^N - 1$ ,即任何一个视图的合作伙伴是其自身。对于部分数据立方体,如果视图  $v \in M$ ,则  $P[v] = 1$ ,表示要实例化  $v$  中的所有元组,否则,  $P[v] = 0$ ,即不实例化  $v$  中的任何一个元组。对于 condensed cube,

除基本关系  $R$  外,每个视图都以自身和  $R$  作为合作伙伴,  $P$  的每个元素的值为2。最后,对 quotient cube,  $P[i] = 3, 0 \leq i \leq 2^N - 1$ 。

Algorithm TCUBE

Input: base relation  $R$ , partner list  $P$

Output: data cube

1:  $gtuple = (All, All, \dots, All), gmask = (0, 0, \dots, 0)$

2:  $MBUC(R, 0)$

$MBUC(bts, dim) \{$

1  $ID = cID = 0$

2 for  $(d=0; d < N; d++) ID = 2 * ID + gmask[d]$

3 switch  $(P[ID]) \{$

4 case 0: break

5 case 1:  $m = AF(bts)$ , write  $gtuple, m$  to view  $[ID]$

6 break

7 case 2:

8 case 3: for  $(d=0; d < N; d++) \{$

9 if  $(bts[d]$  has unique value)

10  $ctuple[d] = bts[d], cmask[d] = 1$

11 else

12  $ctuple[d] = All, cmask[d] = 0$

13 for  $(d=0; d < N; d++) cID = 2 * cID + cmask[d]$

14 if  $((P[ID] = 2 \&\& cID) \neq 2^N - 1)$

15  $\|(P[ID] = 3 \&\& cID = ID)$

16  $m = AF(bts)$ , write out  $gtuple, m$  to view  $[ID]$

17 break  $\}$

17 for  $(d=dim; d < N; d++) \{$

18  $c = Partition(bts, d, PT)$

19 for  $(i=0; i < c; i++) \{$

20  $gtuple[d] = PT[i], gmask[d] = 1$

21  $MBUC(PT, d+1)$

22  $gtuple[d] = All, gmask[d] = 0 \}$

图1 TCUBE 算法

对数据立方体中的每个视图,我们赋予它一个标识,既可以用二进制表示也可以用十进制表示。例如,假设基本关系  $R$  有两个维属性  $A$  和  $B$ ,共有四个视图:  $AB, A, B$  和  $\Phi$ ,它们的标识为11(3)、10(2)、01(1)和00(0),其中括号内为十进制数。按照这种表示方法,如果基本关系  $R$  有  $N$  个维属性,则  $R$  的标识为  $2^N - 1$ 。

在程序中,全局变量  $gtuple$  表示要计算的元组,  $gmask$  是  $gtuple$  所属视图的二进制表示,它们的初始值代表视图  $\Phi$  中的唯一元组。全局变量  $N$  代表基本关系  $R$  中维属性的个数。设置了  $gtuple$  和  $gmask$  的初始值后,调用递归函数  $MBUC$ 。

$MBUC$  首先计算  $gtuple$  所在视图的十进制标识,用变量  $ID$  表示[行1,2]。然后根据数组元素  $P[ID]$  的值做相应处理。  $P[ID] = 0$ ,意味着不需要实例化视图中的任何元组;  $P[ID] = 1$ 时,在第5行计算出元组  $gtuple$  的聚集函数值  $m$  ( $AF$  可以是任何一个聚集函数)并将元组  $gtuple$  和  $m$  写到  $gtuple$  所属的视图中,达到实例化  $gtuple$  的目的。  $P[ID] = 2$  或  $P[ID] = 3$  表示要生成 condensed cube 或 quotient cube。行8~12计算  $gtuple$  所属等价类的封闭元组  $ctuple$  和  $ctuple$  所属视图的二进制标识  $cmask$ ,然后在行13把  $cmask$  转换成十进制形式  $cID$ 。对 condensed cube,如果  $ctuple$  不属于基本关系  $R$  ( $cID \neq 2^N - 1$ ),则需要实例化  $gtuple$ 。对 quotient cube,如果  $ctuple$  和  $gtuple$  属于同一个视图( $cID = ID$ ),也意味着  $ctuple$  与  $gtuple$  相同,则实例化  $gtuple$ 。因此,在15行计算出  $gtuple$  的聚集函数值并输出。行17~22同 BUC 算法,用于递归地产生数据立方体中的其它元组。

### 4 实验结果

为了验证算法 TCUBE 的性能,我们使用实际数据集  $weather$ <sup>[11]</sup>进行了实验。实验是在一台赛扬1.9GHz,256M内存,运行 Windows XP 的笔记本电脑上执行的,我们用 Microsoft Visual C++ 6.0 实现了所有的算法,聚集函数为 SUM,

计算结果存放在文本文件中。根据文[4,5]的思想,尽我们最大的能力实现了算法 MinCube<sup>[4]</sup>和 DFS<sup>[5]</sup>。

*weather* 数据集被多个算法所采用<sup>[4~7,9]</sup>,它有1,015,367个元组(大约 27.1MB),9个维: *station-id*(7037)、*longitude*(352)、*solar-altitude*(179)、*latitude*(152)、*present-weather*(101)、*day*(30)、*weather-change-code*(10)、*hour*(8)和 *brightness*(2),括号中的数字是基数。我们通过投影的方法从 *weather* 数据集中产生了8个数据集,它们的维数是2、3、...、9,即前2个、3个、...、9个维的投影。去掉相同的元组后,8个数据集中的元组个数分别是7037、298215、298215、540769、1004515、1005318、1015367和1015367。

实验比较了算法 TCUBE 和 MinCube, TCUBE 和 DFS 在生成 condensed cube 和 quotient cube 所需要的时间(秒),结果如表1和表2所示。实验结果表明,TCUBE 计算速度要快于 MinCube 和 DFS。主要原因在于 MinCube 需要输出大量的 Bitmap 索引,并且随着维数和基本关系中元组个数的增多,索引的个数和索引占用的空间也增多;而 DFS 会产生大量相同的 upper bound,需要通过排序来去除重复。

表1 TCUBE 与 MinCube 的执行时间

Dims	TCUBE	MinCube
2	9	0
3	7	17
4	4	36
5	15	197
6	63	917
7	141	1570
8	256	4271
9	600	15428

表2 TCUBE 与 DFS 的执行时间

Dims	TCUBE	DFS
2	6	8
3	9	14
4	5	16
5	9	44
6	33	199
7	52	595
8	94	1014
9	161	1420

**结束语** 研究工作者提出了许多类型的数据立方体,需要不同的存储空间,查询响应时间也各不相同,给用户提供了

丰富的选择,不同的数据立方体需要不同的生成算法。本文分析了可以使用关系系统作为存储结构的一般数据立方体、部分数据立方体和浓缩立方体的特点,发现可以用合作伙伴的概念来统一描述这些立方体,并设计了算法 TCUBE 用于生成这三类数据立方体。我们还通过实验验证了 TCUBE 的性能,结果表明 TCUBE 生成浓缩数据立方体的速度要快于原有的算法。

## 参考文献

- Gray J, Bosworth A, Layman A, et al. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. In: S. Y. W. Su, ed. Proc. of the Twelfth Intl. Conf. on Data Engineering. New Orleans: IEEE Computer Society, 1996. 152~159
- Harinarayan V, Rajaraman A, Ullman J D. Implementing Data Cubes Efficiently. In: H. V. Jagadish, Inderpal Singh Mumick, eds. Proc. ACM SIGMOD Intl. Conf. on Management of Data. Montreal: ACM Press, 1996. 205~216
- Shukla A, Deshpande P M, Naughton J F. Materialized View Selection for Multidimensional Datasets. In: A. Gupta, O. Shmueli, J. Widom, eds. Proc. of 24rd Intl. Conf. on Very Large Data Base. New York City: Morgan Kaufmann, 1998. 488~499
- Wang W, Lu H, Feng J, Yu J X. Condensed Cube: An Effective Approach to Reducing Data Cube Size. In: Proc. of the 18th Intl. Conf. on Data Engineering. San Jose: IEEE Computer Society, 2002. 155~165
- Lakshmanan L V S, Pei Jian, Han Jiawei. Quotient Cube: How to Summarize the Semantics of a Data Cube. In: Proc. of 23rd Intl. Conf. on Very Large Data Bases. Hong Kong: Morgan Kaufmann, 2002. 778~789
- Lakshmanan L V S, Pei Jian, Zhao Yan. QC-Trees: An Efficient Summary Structure for Semantic OLAP. In: Alon Y. Halevy, Zachary G. Ives, AnHai Doan, eds. Proc. ACM SIGMOD Intl. Conf. on Management of Data, San Diego, California, USA: ACM Press, 2003. 64~75
- Sismanis Y, Deligiannakis A, Roussopoulos N, Kotidis Y. Dwarf: Shrinking the PetaCube. In: M. J. Franklin, B. Moon, A. Ailamaki, eds. Proc. ACM SIGMOD Intl. Conf. on Management of Data. Madison: ACM Press, 2002. 464~475
- Barbara D, Sullivan M. Quasi-cubes: Exploiting approximation in Multidimensional Databases. SIGMOD Record, 1997, 26: 12~17
- Beyer K, Ramakrishnan R. Bottom-Up Computation of Sparse and Iceberg CUBEs. In: A. Delis, C. Faloutsos, S. Ghandeharizadeh, eds. Proc. ACM SIGMOD Intl. Conf. on Management of Data. Philadelphia: ACM Press, 1999. 359~370
- Zhao Y, Deshpande P M, Naughton J F. An Array-Based Algorithm for Simultaneous Multidimensional. In: Joan Peckham, ed. Proc. ACM SIGMOD Intl. Conf. on Management of Data. Tucson: ACM Press, 1997. 159~170
- Hahn C, Warren S, London J. Edited synoptic cloud reports from ships and land stations over the globe. <http://cdiac.esd.ornl.gov/cdiac/ndps/ndp026b.html>

(上接第68页)

- Anomaly Based Network Intrusion Detection. In: Proc. of the First IEEE Intl. Workshop on Information Assurance(IWIA'2003)
- Warrender C, Forrest S, Pearlmutter B. Detecting Intrusions Using System Calls: Alternative Data Models. In: IEEE Symposium on Security and Privacy, 1999
- Jha S, et al. Classifiers, and Intrusion Detection. In: 14th IEEE Computer Security Foundations Workshop(CSFW'01), June 2001
- Gao B, et al (HMMS (HIDDEN MARKOV CHAIN MODELS) BASED ON ANOMALY INTRUSION DETECTION METHOD. In: Proc. of the First Conf. on Machine Learning and Cybernetics, Beijing, Nov. 2002. 381~385
- Das K. Protocol Anomaly Detection for Network-based Intrusion Detection. <http://www.sans.org/rr/papers/30/349.pdf>

- Bykova M, Ostermann S, Tjaden B. Detecting Network Intrusions via Statistical Analysis of Network Packet Characteristics. In: Proc. of the 33rd southeastern Symposium on System Theory, 2001
- <http://www.snort.org/>
- Paxson V. Bro: A System for Detecting Network Intruders in Real-Time. In: Proc. of the 7th USENIX Security Symposium San Antonio, Texas, Jan. 1998
- [http://www.ll.mit.edu/IST/ideval/data/data\\_index.html](http://www.ll.mit.edu/IST/ideval/data/data_index.html)
- Postel J. Transmission Control Protocol. RFC 793, Sep. 1981
- <http://www.tcpdump.org/>
- 谭小彬,王卫平,奚宏生,殷保. 系统调用序列的 Markov 模型及其在异常检测中的应用. 计算机工程, 2002, 12: 189~191
- 罗鹏飞,张文明,刘福声. 随机信号分析. 国防科技大学出版社, 2000. 176~183