计算机科学 2005Vol. 32No. 11

一种有效的并行数据库数据分布方法 RCMD*)

艾春宇[1] 李建中[1,2] 高 宏[2]
(黑龙江大学计算机科学技术学院 哈尔滨 150080)[1]
(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)[2]

摘 要 在并行数据库中,数据的分布方法是影响系统查询处理性能的主要因素。目前已有的几种数据分布方法都只适用于某一类查询,而处理其它类型的查询则效率较低。本文提出了一种新的数据分布方法 RCMD,可以高效地支持多种查询类型。理论分析和试验结果表明本文提出的 RCMD 方法优于现有的数据分布方法,具有最好的查询处理性能。

关键词 并行数据库,数据分布方法,RCMD

An Efficient Data Declustering Method RCMD in Parallel Database

AI Chun-Yu[1] LI Jian-Zhong[1,2] GAO Hong[2]
(College of Computer Science and Tahnology, Heilongjiang University, Harbin 150080)[1]
(College of Computer Sciena and Technology, Harbin Institue of Technilg, Harbin 150001)[2]

Abstract Data declustering methods have great influence on the query processing performance in shared-nothing parallel database system. The existed data declustering methods have a same disadvantage that they are efficient only for some kinds of queries, and worse for the other kinds of queries. In this paper, we propose a new data declustering method—RCMD, which is effective for all kinds of queries. Theoretical analysis and experimental results show that RCMD outperforms the existed methods while processing queries in parallel database.

Keywords Parallel database, Data decluster, RCMD

1 引言

在基于机群系统的并行数据库研究中,并行数据库物理存储方法是一个重要的研究内容。并行数据库物理存储方法是指如何在多个处理机之间分布各种数据库对象,最小化查询处理时间。数据存储方法对并行数据库系统的查询性能有着极大的影响,在查询处理过程中,如果数据分布不合理,系统的并行性就得不到充分的发挥,降低了系统的查询处理能力[1]。

目前,在数据分布策略方面已开展了大量的研究工作,提出了很多有效的并行数据分布方法,如 Round-Robin[2]、Hash[3]、Range-Partition[4]、CMD[5]等数据分布方法。Round-Robin 方法以轮转的方式将关系的元组分布到各个处理机上,当关系上的操作需要存取大量元组时,一般采用这种分布方法,但是 Round-Robin 不能有效地支持具有低选择性谓词的查询,这样的查询仅存取较少的元组,而 Round-Robin 方法却要求所有的处理机都启动工作,降低了系统的效率。Hash 分布方法选择关系的一个属性进行 Hash,然后根据 Hash 值将关系分布到各个处理机上。这种分布方法可以有效地支持划分属性上精确匹配谓词的查询,但是 Hash 方法不能保证数据均匀地分布在多个处理机上。Range-Partition 的分布方法是将关系的一个属性的值域分成若干个区间,然后根据这些区间将关系分布到各个处理机上。这种分布方法可以有效地支持要求大数据量存取的查询和在划分属性上具有低选择性谓词的数据操作。但是 Range-Partition 分布方法可能引起两个问题。第一个问题是数据在处理机之间分布不均匀;另一个问题是工作负载的不均匀,在最坏的情况下,一个访问大量数据查询的执行可能集中在一个处理机上[1]。CMD 方法数据划分对称地在所有属性上进行,可以有效地支持各种选择谓词的查询,经过 CMD 方法划分的关系是部分有序的[5]。但是,CMD 方法在处理连接操作时,由于通讯开销较大,效率并不理想。Hash 和 Range 数据分布方法可以有效支持划分属性上的连接操作,但是 Hash 分布方法不能有效支持带选择性谓词的查询,而 Range 分布方法只支持划分属性上的选择性谓词查询。

在数据库的应用中,查询的类型是多样的。上述四种数据分布方法都是只适用于某些类型的查询,而在处理其它类型的查询则效率较低,甚至会导致严重负载倾斜或大量的网络通讯的出现,系统整体的查询处理效率不能达到最优状态。本文针对这个问题,提出了一种新的数据分布方法 Range-CMD(简称 RCMD)。RCMD 数据分布方法具有如下的特点:

1)能够有效地支持多种类型的查询,如多选择性谓词的查询、精确匹配的查询、连接查询等等;

2)RCMD 分布的关系在各个属性上都是部分排序的,因此基于 RCMD 的连接等操作的实现算法比现有的算法有效;

3)RCMD 将关系划分为多个超长方体,每个超长方体存储在一个磁盘物理页面中,RCMD 为每个超长方体建立了索引,利用该索引在处理查询时能够有效地减少磁盘 I/O。

本文的贡献在于提出了一种新的数据分布方法 RCMD 及其物理存储结构,并将 RCMD 和已有的数据分布方法进行了对比分析。理论分析结果和试验结果都表明,采用 RCMD 数据分布方法分布数据时,并行数据库整体查询处理性能是最好的。本文第 2 节介绍 RCMD 数据分布方法;第 3 节给出基于 RCMD 分布的物理存储结构;第 4 节对 RCMD 方法和其他几种数据分布方法的性能进行对比分析;第 5 节给出试验结果及相应的分析,验证本文提出方法的有效性;最后

*)本文研究得到了国家 863 计划(2005AA4Z3080)基金支持。艾春宇 硕士,助教。李建中 博士导师,教授。高 宏 博士,副教授。

对本文的工作进行总结。

2 RCMD数据分布方法

给定一个具有 d 个属性的关系 $R(A_1,\cdots,A_d)\in D_1\times\cdots\times D_d$，其中是属性 A_i 的定义域，R 被视为 d 维数据空间 $S=D_1\times\cdots\times D_d$ 的子集合。RCMD数据分布方法首先在d个属性中选择一个属性作为划分属性，然后将数据空间 S 的各维划分成多个不相交区间，空间 S 被划分为多个子空间，并使得 R 在每个子空间中具有近似相等的元组数。于是，关系 R 被划分成大小近似相等的子空间。然后，按照一定的规则把 S 的每个子空间分配到一个处理机。为了叙述简单，令数据空间 $S=[0,1)^d$。设 $C=\{C_0,C_1,\cdots,C_{p-1}\}$ 为机群中的处理机集合，P 是处理机个数。P 个处理机的内存容量(元组数)分别为 M_1，M_2，…，M_p，数据划分属性为 A_T。

为了划分数据空间 S，RCMD方法把 S 的第 i 维的值域划分成长度为 $\frac{1}{n_iP}$ 的 n_iP 个子区间：

$$\left[0,\frac{1}{n_iP}\right),\left[\frac{1}{n_iP},\frac{2}{n_iP}\right),\cdots,\left[\frac{n_iP-1}{n_iP},1\right)$$

其中 n_i 是调整因子，它必须充分大，使得下列两个条件成立：

(1)划分后的每个子空间(超长方体)所包含的 R 的元组可装入一个物理磁盘页；

(2)R 的任一维 i 的任一区间 $\left[\frac{j}{n_iP},\frac{j+1}{n_iP}\right),j=(0,1,\cdots,n_iP-1)$ 满足：

$$\left|\left\{t|t\in R,t[A]\in\left[\frac{j}{n_iP},\frac{j+1}{n_iP}\right)\right\}\right|\leqslant\sum_{i=1}^{p}M_i$$

其中 $|X|$ 表示关系 X 的元组数，$t[A]$ 是元组 t 的 A 属性值，A 是 R 的第 i 维对应的属性。

条件(2)保证了关系 R 与每维的每个划分区间对应的子集合(即 A 属性值属于该划分区间的元组集合)可以存储在 P 个处理机的内存中，有效地支持并行数据操作算法的设计与实现。

设 $I_{ij}=\left[\frac{j}{n_iP},\frac{j+1}{n_iP}\right)$ 是第 i 维上的第 j 个子区间，j 是这个子区间的坐标。RCMD方法把数据空间 S 划分为 $(n_1P\times\cdots\times n_dP)$ 个超长方体，因此 S 可以看作是超长方体构成的集合。每个超长方体都是 d 个子区间的笛卡尔积 $I_{1x_1}\times\cdots\times I_{dx_d}$，$(x_1,\cdots,x_d)$ 定义为该超长方体的标识，其中 $0\leqslant x_i<n_iP,i=(1,2,\cdots,d)$。每个超长方体 $(x_1,\cdots,x_d)$ 由数据分布函数 df，根据其划分属性上的子区间来决定其分配到哪个处理机。df 函数如下定义：$df:S\to\{0,1,\cdots,P-1\}$，$\forall(x_1,\cdots x_d)\in S$，$df((x_1,\cdots,x_d))=x_i \bmod P$。其中，$x_i$ 为该超长方体在划分属性 A_T 上的子区间编号。若 $df((x_1,\cdots,x_d))=k$，则该超长方体被分配到处理机 C_k 上。

下面以一个例子来说明RCMD数据分布过程。给定一个数据空间 $S=[0,1)^2$，即 S 有两维，每个维度的值域空间为 $[0,1)$。在本例中，数据划分的调整因子 $n_1=n_2=2$，处理机个数 $P=4$，划分属性选择的是第一维属性，则数据空间 S 被划分成64个超长方体，$S=\{(0,0),(0,1),\cdots,(7,7)\}$。每个超长方体根据数据分布函数 df 来决定其所分配的处理机。图1给出了本例中每个超长方体分配的处理机编号，其中超长方体 (i,j) 分配的处理机编号为图中第 i 行、第 j 列对应的值。例如，超长方体(6，3)被分配的处理机编号为2。从图1中可以看出，在划分属性上属于同一子区间的超长方体被分配到相同的处理机上。

7	0	1	2	3	0	1	2	3
6	0	1	2	3	0	1	2	3
5	0	1	2	3	0	1	2	3
4	0	1	2	3	0	1	2	3
3	0	1	2	3	0	1	2	3
2	0	1	2	3	0	1	2	3
1	0	1	2	3	0	1	2	3
0	0	1	2	3	0	1	2	3
	0	1	2	3	4	5	6	7

图1 $S=[0,1)^2$ 在4个处理机间划分的实例

RCMD数据分布方法同时具备CMD和Range分布的优点。除了整个关系均匀地分布在多个处理机上，同时还具有以下优点：

1)在任一维 k 上都部分有序。设 I_{ki} 和 I_{kj} 是第 k 维(对应属性为 A)上的两个划分区间，若 $i<j$，则

$\forall u\in\{t|t\in R,t[A]\in I_{ki}\}$，

$\forall v\in\{t|t\in R,t[A]\in I_{kj}\}$

$\Rightarrow u[A]<v[A]$；

2)对于任意维 k 上的任意一个划分区间 I_{kj}，$\{t|t\in R,t[A]\in I_{kj}\}$ 的数据量不超过全部处理机的内存空间总容量，并且均匀地分布在 P 个处理机上(划分属性维例外)；

3)如果 I_{ki} 和 I_{kj} 是第 k 维(对应属性为 A)上两个不同的划分区间，则 $\{t|t\in R,t[A]\in I_{ki}\}$ 和 $\{t|t\in R,t[A]\in I_{ki}\}$ 具有近似相等数量的元组。

RCMD关系在每个处理结点上的超长方体个数相等，每个超长方体的大小也近似相等，所以每个处理结点上的数据子集合的大小都近似相等。RCMD分布的关系基本是均衡的，能够充分发挥系统的并行性。由于数据划分对称地在所有属性上进行，RCMD方法可以有效地支持具有各种选择谓词的查询。经过RCMD方法划分的关系是部分排序的，该性质使基于RCMD的SORT和JOIN等数据操作的实现算法远比现有的算法有效。RCMD方法可以通过为频繁连接的两个关系指定相同的划分属性和划分区间而避免并行处理连接时产生的网络通讯开销。

3 基于RCMD划分的物理存储结构

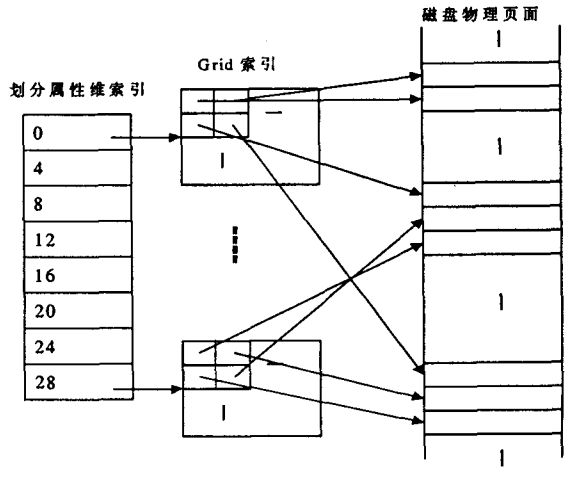

图2 基于RCMD分布的物理存储结构

在每个处理机上，基于RCMD分布的数据采用如图2所

示的结构进行存储。每个超长方体单独存储在一个磁盘物理页面上，同时 RCMD 为这些超长方体建立两级索引。第一级索引建立在划分属性维上，划分属性上的每个划分子区间作为一个索引项，索引项的值为指向第二级索引地址的指针。第二级索引为一个具有($d-1$)维的 Grid 索引，每个索引项的值为对应的超长方体所在的磁盘页物理地址。当 RCMD 的两级索引占用空间较小时，可以保存在主存中；如果两级索引很大，无法装入主存时，则只在主存保存第一级索引。

RCMD 的物理存储结构能够减少数据访问时的磁盘 I/O。对于各个属性上的区域查询条件和精确查询条件，都可以通过划分属性维索引和 Grid 索引直接定位到要访问的数据所在的磁盘物理页进行存取，并且不会存取任何不包括符合条件数据的页面。RCMD 的物理存储结构并不会影响在任何属性上建立其它索引结构。

4 RCMD 与其他数据分布方法的性能对比分析

在机群并行环境中，磁盘 I/O 和网络通讯带宽都是影响查询处理性能的关键因素，所以比较各种分布方法的性能也要从这两个方面进行分析。查询的通讯开销主要是处理连接和聚集操作产生的，其它操作如选择、投影的处理不会产生通讯开销，则网络通讯的开销为 0。下面给出会产生网络通讯开销的连接和聚集操作的网络通讯代价的计算公式：

$$Cost\_COMM(O,R)=|R|\times|t_r|\times\rho\times S(O\_Set(A))$$

其中，$Cost_COMM(O,R)$表示查询 Q 上的连接或聚集操作 O 在关系 R 上产生的通讯代价，$|R|$表示关系 R 的元组数，t_r 表示 R 中的一个元组包含的字节数，ρ 表示查询 Q 操作的 R 中元组数占总元组数的百分比。关系 R 上的操作相关的属性或属性集用 $O_Set(A)$表示[6]：

$$S(O\_Set(A))=\begin{cases}0, & \text{if}(O\_Set(A)\text{包括划分属性})\\ 1, & \text{if}(O\_Set(A)\text{不包括划分属性})\end{cases}$$

对于聚集操作，如果聚集属性包括划分属性，则聚集操作算法不需要重新分布数据，不会带来任何网络通讯开销，$S(O_Set(A))$取 0 值，则通讯代价为 0。如果聚集属性不包括划分属性，则操作算法首先要重新分布数据，所以通讯代价为$|R|\times|t_r|\times\rho$。对于连接操作，如果连接的两个关系的划分属性均为连接属性，并且划分子区间相同，则采用本地连接策略，不需要任何网络通讯，$S(O_Set(A))$取 0 值，否则关系 R 上的通讯代价为$|R|\times|t_r|\times\rho$。各种分布方法的网络通讯代价如表 1 所示。由于 CMD 和 Round-Robin 方法没有划分属性，因此连接和聚集操作需要重新分布数据，通讯开销为$|R|\times|t_r|\times\rho$。由前面的分析可知，当一种数据分布方法没有划分属性时，如 CMD、Round-Robin 分布方法，在处理连接和聚集操作时必然会有通讯开销；当一种数据分布方法具有划分属性时，如 RCMD、Hash 和 Range 分布，在处理划分属性上的连接和聚集操作时没有通讯开销。

下面分析各种数据分布方法处理操作时的磁盘 I/O 代价分析。设关系 R 存储在 $P(R)$个磁盘页面上，查询 Q 需要访问的关系 R 上数据存储在 $P(R_Q)$个磁盘页面上。因为操作系统对磁盘到内存的调度单位为页，所以查询 Q 在关系 R 上读取数据的最小磁盘 I/O 代价即为 $P(R_Q)$。CMD 分布方法的磁盘 I/O 代价能够达到这个最小值 $P(R_Q)$[5]，而 RCMD 方法如上一节物理存储结构所述，不会存取任何不包括查询所需数据的超长方体，也就是物理页面，所以磁盘 I/O 代价也是最小值 $P(R_Q)$。

表 1 各种分布方法的代价比较

数据分布方法	连接和聚集的网络通讯代价	磁盘 I/O 代价
RCMD	$\begin{cases}0 & \text{连接或聚集属性包括划分属性}\\ |R|\times|t_r|\times\rho & \text{其它情况}\end{cases}$	$P(R_Q)$
CMD	$|R|\times|t_r|\times\rho$	$P(R_Q)$
Range 或 Hash	$\begin{cases}0 & \text{连接或聚集属性包括划分属性}\\ |R|\times|t_r|\times\rho & \text{其它情况}\end{cases}$	$Cost\_IO(Q,R)=\begin{cases}P(R_Q) & (1)\\ P(R)\times\alpha,\ (\frac{P(R_Q)}{P(R)}<\alpha<1) & (2)\\ P(R) & (3)\end{cases}$
Round-Robin	$|R|\times|t_r|\times\rho$	同上

Range 和 Hash 分布方法磁盘 I/O 代价的计算比较复杂。Range 和 Hash 分布方法能够根据划分属性维的值域确定数据分布在哪些处理结点上。如果查询的选择性谓词包括划分属性，则能够消除不包括所需存取数据的处理结点上的磁盘 I/O，但是对于需要存取数据的处理结点的磁盘 I/O 的减少则没有任何贡献。如果关系 R 在查询属性上建有索引，则能够有效地减少磁盘 I/O。Range 和 Hash 分布方法查询 Q 在关系 R 上的磁盘 I/O 代价 $Cost_IO(Q,R)$如表 1 所示，公式中的三种情况如下：

(1)查询的所有选择性谓词包含的属性上都建有索引。

(2)查询的某一个或几个选择性谓词包含的属性上有索引，或者某个选择性谓词包含的属性为划分属性。

(3)选择性谓词包含的属性都没有任何索引，也不包括划分属性。

第一种情况达到了磁盘 I/O 值代价的最小值，但是必须满足的条件比较苛刻，能够满足条件的一般都是简单查询。在其它两种情况下的磁盘 I/O 代价比 CMD 和 RCMD 方法要大得多。Round-Robin 分布方法的磁盘 I/O 代价的计算方法与 Range 和 Hash 方法基本相同，但是不存在查询的选择性谓词包括划分属性能够较少磁盘 I/O 的优势，因为 Round-Robin 与 CMD 方法一样没有划分属性。

表 1 给出了各种数据分布方法在处理连接和聚集操作的网络通讯代价和磁盘 I/O 代价对比[1]。可以看出，RCMD、Range 和 Hash 具有最小的网络通讯代价，而 RCMD 和 CMD 方法具有最小的磁盘 I/O 代价。由分析可知，RCMD 方法的通讯和磁盘 I/O 代价都是最小的，因此 RCMD 要优于其它的分布方法。一个查询的执行代价既包括磁盘 I/O 代价也包括网络通讯代价，所以从总体衡量，关系采用 RCMD 方法分布数据能够使得其上的查询获得更高的整体执行性能。

5 试验结果及分析

试验的硬件环境为 8 个结点组成的机群并行机，每个节点的 CPU 为 3G Hz，硬盘大小为 80G，I/O 随机访问速度为 5M/s，结点机之间采用千兆以太网互联。实验的数据是

[1] 对于任何一种数据分布方法，在处理选择和投影操作时都没有网络通讯的开销。

TPC-H 中的关系数据。试验中，我们使用了 100 个查询作为查询负载，其中包括多种类型的查询，如区域查询，连接查询、聚集查询等。试验中，对采用 Range、Hash 和 Round-Robin 方法分布的每个关系都在选择性谓词出现频率最高的两个属性上建有索引。

试验 1 中，我们考察了在查询负载选择性(查询访问的元组数占关系总元组数的百分比)固定、试验数据量变化的情况下，当数据采用不同的分布方法进行存储时，查询负载整体的执行时间情况。试验数据的数据量分别为 100MB、300MB、1GB 和 3GB。图 3 给出了选择性为 0.6 时查询负载的执行时间，图 4 给出了选择性为 0.1 时查询负载的执行时间。从这两个图中可以看出，查询负载的执行时间随着数据量的增加而增加。RCMD 方法的执行时间在各种数据量的情况下都是最小的，并且 RCMD 方法的执行时间随数据量的增加而增加的趋势较缓，而其它方法则增长较快。这是因为 Round-Robin、Hash 和 Range 方法随着数据量的增加，磁盘 I/O 也大量地增加，而 CMD 方法随着数据量增加通讯代价显著增加的结果。从图 4 可以看出，当查询负载的选择性小时，RCMD、CMD 方法和其它三种方法的执行时间差距更大，这是因为 RCMD 和 CMD 方法在查询的选择性小的情况下，查询处理所需的磁盘 I/O 也随之变小，一维划分的关系上虽然有索引，但不能满足所有的选择性谓词，因此磁盘 I/O 不会随查询负载的选择性成正比下降。

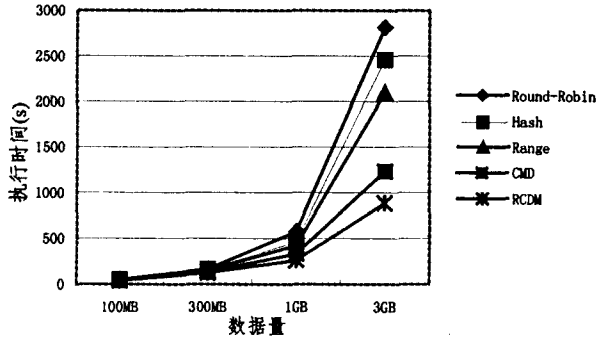

图 3　选择性为 0.6 的查询负载处理性能

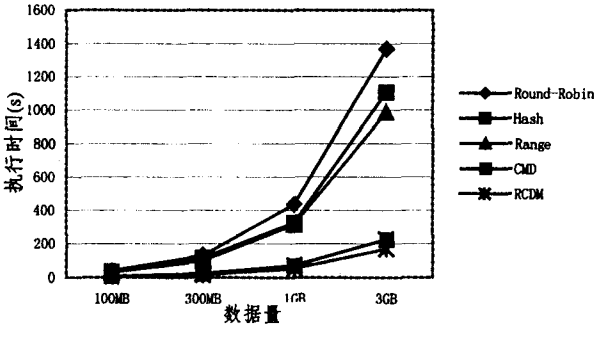

图 4　选择性为 0.1 的查询负载处理性能

试验 2 中，我们考察了在数据量大小固定、查询选择性变化的情况下，当数据采用不同的分布方法进行存储时，查询负载整体的执行时间情况。试验结果如图 5 所示。可以看出，随着查询负载选择性的增加，各个划分策略的执行时间都呈现增长趋势。其中 CMD 和 RCMD 方法增长较快，特别是 CMD 方法，因为 CMD 方法在选择性增加的时候磁盘 I/O 和网络通讯开销都随之增加。在选择性为 0.9 时，CMD 方法的执行时间已经超过了 Hash 和 Range 方法，这是因为在选择性较大情况下 CMD 已经没有磁盘 I/O 少的优势，同时通讯代价却显著增加。由图 5 可以看出，在查询选择性不同的情况下，RCMD 一直具有最好的性能。

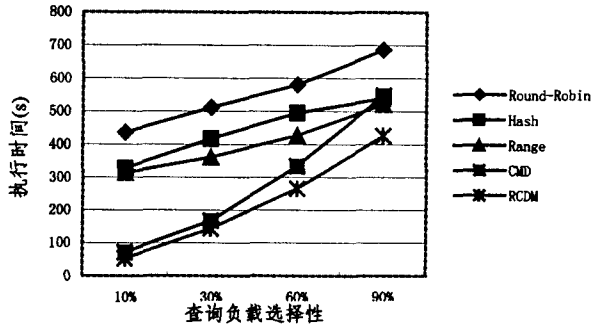

图 5　不同选择性的查询负载执行时间

由上述试验结果可以看出，RCMD 方法的查询处理性能要优于目前已有的方法，特别是在查询负载选择性较低的情况下。

结论　数据的分布方法是影响并行数据库系统查询处理性能的主要因素。目前已有的几种数据分布方法都只适用于某一类查询，而在处理其它类型的查询则效率较低。本文提出的数据分布方法 RCMD，可以高效地支持多种查询类型。理论分析和实验结果表明，本文提出的 RCMD 方法优于现有的数据分布方法，具有最好的查询处理性能。

参 考 文 献

1　李建中，孙文隽. 并行关系数据库管理系统引论. 北京：科学出版社，1998. 57～71

2　Teradata Corporation. DBC/1012 Data Base Computer Concepts and Facilities. Teradata Document C02-001-05, Los Angeles, Calif, 1998

3　Kitsuregawa M, Tanaks H, Moto-Oka T. Architecture and Performance of Relational Algebra Machine GRACE. In: Proc. of the Intl. Conf. on Parallel Processing, Chicago, 1984

4　DeWitt D J, et al. GAMMA: A High Performance Dataflow Database Machine. In: Proc. of Inter. Conf. on VLDB, 1986. 228～237

5　Li Jianzhong, Srivastava J, Rotem D. CMD: A Multidimensional Declustering Method for Parallel Database Systems. In: Proc. of the 18th VLDB Conf. Vancouver, British Columbia, Canada, 1992

6　艾春宇，李建中，高宏，等. 自适应的并行关系存储方式选择算法及在线转换技术. NDBC, 长沙，2003