SA:一种有利于多属性范围查询的多维聚簇方法

吴凌坤 汤 庸 王 鹏 舒 然
(中山大学计算机科学系 广州 510275)

摘 要 一般来说,外存访问的数据文件中针对多属性的区域查询有两个改进其效率的方向。一个是在其上建立索引,另一个是在物理层按照某种规律重新安排记录。探讨如何通过第二种方法来提高范围查询的效率,即通过多维聚簇的方式得到数据文件中更好的记录的存储顺序。首先,细致分析了该问题,并针对该问题构造了一个数学模型,然后通过引入光谱算法(SA)的思想为解决该NP难问题提供了一种多项式时间内的近似解。最后通过实验来验证了该方法在矩形区域查询和单维范围查询方面的有效性。

关键词 高维聚簇,数据重组,范围查询,光谱算法

中图法分类号 TP302 **文献标识码** A

SA: A New Multidimensional Clustering Method to Facilitate Range Queries on Multiple Attributes

WU Ling-kun TANG Yong WANG Peng SHU Ran
(Department of Computer Science, Sun Yat-Sen University, Guangzhou 510275, China)

Abstract Generally there are two directions to improve the query performance of range queries on multiple attributes in a static data file. One is to devise an index, and the other is to rearrange records in physical layer. In this paper, we took the second way to give a better data file organization, which we call multidimensional clustering. First we analyzed the problem, and constructed a mathematical model for this it, and then based on the idea of Spectrum Algorithm (SA), we devised a polynomial method to heuristically solve this NP-hard problem. And the experiment results show that the spectrum algorithm is an effective record reorganization method for range queries.

Keywords Multidimensional clustering, Data reorganization, Range query, Spectrum algorithm

1 引言

针对多维属性的区域查询在数据管理系统(如数据库、数据仓库、OLAP、数据挖掘等)中是一种十分重要且常用的记录检索类型。这类查询一般返回一个记录的集合而不是单个记录。为了提高这种查询的效率,首先需要知道这些数据文件的物理结构。外存访问系统的文件存储形式通常分为两个主要部分:数据文件和索引文件。因此一般有两个改进查询效率的方向:一个是设计更好的索引,一个是更好地组织数据文件。到目前为止,已有大量针对改进多维区域查询效率的研究,这些研究大多关注的是索引部分,但是对数据文件的研究也同样重要。

众所周知,主存与磁盘交换数据的单位是块(Block,或页,Page)而不是记录,一个块一般来说包含若干个记录。从外存中读取一个块的时间开销远大于在主存中取同样大小的数据。此外由于寻道的时间开销(对于绝大多数外存访问来说,就是硬盘的磁盘旋转和磁针移动),在磁盘上读取不连续的块将花费更多的时间,而主存则不存在这个问题(事实上,在主存上读取任何数据都只花费常数时间)。在外存上访问数据花费的时间为毫秒级(ms),而在主存上访问数据花费的时间为微秒级(ns),这意味着在外存上访问数据比主存慢1000倍以上。因此,为了提高外存访问的速度,应该尽可能地减少查询中必须访问的块并且使这些块尽量的近。故如果我们将关系紧密地记录尽可能地存储在同一个块或相邻的块里,那么就可以很好地减少在数据检索中占主要地位的I/O时间。其中关系紧密的记录是指具有相同或相近查询属性的记录。按照这样的方式将记录重新排序的方式可以看成是对记录的多维聚簇。

本文首先分析该问题,并对这个问题构造一个数学模型,而后通过引入光谱算法(Spectrum Algorithm, SA)的思想,给出一个多想是时间内的近似解。最后通过实验验证了该方法的有效性。第2节主要讨论相关工作,第3节用数学语言给出这个问题的定义,并指出该方法的动机,第4节设计并讨论解决该问题的算法,第5节将给出相关的实验结果,最后对论文进行总结。

2 相关工作

对物理层数据重组的研究相对较少,而对多维聚簇的研

到稿日期:2008-07-03 返修日期:2008-09-19 本文受国家自然科学基金(60673135,60373081)重点项目(60736020),教育部新世纪优秀人才支持计划(NCET-04-0805),广东省自然科学基金(7003721)资助。

吴凌坤 博士生,研究方向为数据库、数据挖掘,E-mail:wulingkun@gmail.com;汤 庸 博士生导师,研究方向为协同软件、数据库;王 鹏 硕士生,研究方向为数据库;舒 然 硕士生,研究方向为数据库。

究则更少。Chung-Shu Yang 提出了一种在磁盘设备中记录地址分配的方法[1]，这种方法保证了对多值查询拥有最少的读写次数。但是应用这种方法的前提是必须事先知道需要优化的多值查询及其相应使用频率。Edward Omiecinski 和 Perter Scheuermann[2]提出了一种减少每次查询访问页面数的记录聚簇和文件重组方法，但是这种方法也只适用于各种记录查询频率已知的情况。在文献[3,4]中，Jong-Hak Lee 等针对数据库底层设计提出了一种用于高维数据重组的区域分裂策略，同样，该方法也要求查询的模式事先知道。Sam S. Lightstone[5]和 S. Padmanabhan[6]等对 DB2 提出一种高维聚簇技术(MDC)，但该方法也需要查询的工作量、执行频率和其他一些细节作为输入参数。C. T. Yu[7]等提出一种减少磁盘寻道时间的适应性方法，该方法将频繁访问的块移动到磁盘的中间，使其寻道时间减少。在这种方法中相关的频率(记录访问频率)无需预先知道，但是需要比较长时间才能起效果(用于慢慢的收集统计数据)，因此在系统刚刚建立的时候该方法并不能提供任何帮助。

可以看到大部分提出的记录重组方法都依赖于某些假设，如已知某些统计数据：查询模式、记录或块的访问频率等。但得到统计数据通常不是一件立刻能做到的事，而是一项耗时耗力的工作。因此在数据文件刚刚建立的情况下以上方法都不适用。

虽然也存在一些不需要预先统计数据的记录组织方法[8,9]，但是它们也有自己的局限性，比如这些方法只能应用于某些确定的索引类型(如 k-d 树)，其他多维索引结构就不能使用它的数据文件。

本文针对静态数据文件提出一种高维聚簇方法，按照该方法组织的文件对于范围查询有着良好的性能。该方法不需要预先的统计数据，且不受制于高层索引文件，这意味着它可能应用于各种不同类型的高维索引结构。

3 问题描述

本文讨论二维的区域查询，但不失一般性，其结果可以简单地扩展至高维。

首先将问题总结如下：在一个数据文件中有 n 个记录，每个记录包含一些属性，假设记录的属性 x 和 y 将要被查询(称之为查询属性 querying attributes 或组织属性 organizing attributes)，目的是重组这些记录，使得对 x 和 y 的任何范围查询所需的 I/O 访问时间最少。I/O 访问时间主要由访问的物理块数决定(这些被访问的块包含着满足查询条件的记录)。我们知道物理块线性存储于磁盘的磁道上，因此对记录的重组就是给所有记录排序。

从另外一个角度考虑这个问题，所有记录可根据 x 和 y 的值映射到二维坐标平面上的点。这种映射是可行的因为所有数据类型均可经过变换映射到实数上。于是对 x 和 y 的范围查询就变成了坐标平面上的矩形框查询。范围查询的结果对应着查询矩形框内包含点的集合(本文的以下部分中的点即指记录)。

这有一个重要的潜在规律：越靠近的点越可能被一个查询矩形框同时包含，或同时不包含。根据这个规律，要做的就是给出如下线性序列：平面上靠得越近的点在线性顺序中也离得越近。

对问题的数学定义如下：

问题定义 坐标平面内给定点集 P，其包含 n 个点(p_1，p_2，…，p_n)，找出一个单射函数 $\pi: P \to \{1,\cdots,n\}$使得

$$W(\pi)=\sum_{p_i,p_j\in P} d(p_i,p_j)(\pi(p_i)-\pi(p_j))^2$$

取最大值。

其中 $d(p_i,p_j)$是点 p_i 和 p_j 的欧式距离。而此时讨论的二维空间中对任意两点 $p_i(x_i,y_i)$ 和 $p_j(x_j,y_j)$有：$d(p_i,p_j)=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$。

需说明的是，以上公式 $W(\pi)$是经验公式。其思想是：如果一个顺序 π 满足我们的要求，令 p_1，p_2 为两个高度关联的记录，p_3，p_4 为关联很小的记录(坐标平面上距离较远的两个记录)，于是一般情况下有 $d(p_1,p_2)<d(p_3,p_4)$ 和$|\pi(p_1)-\pi(p_2)|<|\pi(p_3)-\pi(p_4)|$，故有$(\pi(p_1)-\pi(p_2))^2<(\pi(p_3)-\pi(p_4))^2$。

此外，在该问题的大多数情况中$(\pi(p_i)-\pi(p_j))^2>>d(p_i,p_j)$是成立的。因为我们可以很容易地将所有的记录映射到坐标平面相关很小的区域，在这种情况下，大部分$(\pi(p_i)-\pi(p_j))^2$ 是大于 $d(p_i,p_j)$的。所以，如果有

$(\pi(p_3)-\pi(p_4))^2>(\pi(p_1)-\pi(p_2))^2>d(p_3,p_4)>d(p_1,p_2)$，则：

$d(p_1,p_2)(\pi(p_1)-\pi(p_2))^2+d(p_3,p_4)(\pi(p_3)-\pi(p_4))^2>d(p_1,p_2)(\pi(p_3)-\pi(p_4))2+d(p_3,p_4)(\pi(p_1)-\pi(p_2))^2$ 成立，因为有下面这个简单的定理。

定理 a,b,c,d 是实数，如果 $a<b<c<d$ 则 $a\times d+b\times c<a\times c+b\times d$。

证明：$a\times d+b\times c-(a\times c+b\times d)=(a-b)(d-c)<0$。

事实上 $W(\pi)$可被定义为

$$W'(\pi)=\sum_{p_i,p_j\in P} d(p_i,p_j)^m|\pi(p_i)-\pi(p_j)|^n$$

其中 m,n 为任意值，$W'(\pi)$仍然拥有我们所需要的性质(当 $m=1,n=2$ 时，$W'(\pi)$变成了 $W(\pi)$)。并不是因为 $m=1,n=2$ 为最好的参数，只是这样的参数适合应用光谱算法。

现在我们讨论解的一致性问题。我们说 π 对该问题是一致的，当对任意 $\pi(p_i)<\pi(p_j)<\pi(p_k)$，有 $d(p_i,p_k)\geq d(p_j,p_k)$和 $d(p_i,p_k)\geq d(p_i,p_j)$。这就是说在一个一致性解里，对任意的点 p_i,p_j,p_k，若在线性序列上相对于 p_i,p_j 更靠近 p_k，则坐标平面上 p_j 与 p_k 间的距离小于 p_i 与 p_k 间的距离。

很明显一致解并不一定存在(其实经常不存在)。这是因为将一个更高维的点集映射到较低维的空间上必将丢失一些信息，不可能在低维空间中仍保存所有的高维的信息。因此，需要做的就是找到一个相对的最好解。

不幸的是找到使 $W(\pi)$最优的 π 是一个 NP 难问题[10]。为找到最好的 π 需要遍历所有可能的排列，这在现实应用中是不可行的(因为有 $n!$ 种可能)。因此需要设计一种多项式时间的算法来求得近似最优解。

我们发现这个问题与最小线性排列问题(MinLA)[11-13]十分相似，所以可以在解决 MinLA 问题[14]中采用十分有效的算法——光谱算法(Spectrum Algorithm)作为解决方案。

4 解决该问题的光谱算法

图论中的光谱理论应用有不短的历史。在最近几年，光谱算法已被广泛应用于组合优化问题。这些问题涉及范围很

广，如：VLSI 电路设计、行程安排和考古学、并行计算的网络优化等[15-18]。

在此处利用光谱算法的基本思想是：利用 $f(x)=\sum_{i,j\in\{1,\cdots,n\}} d(p_i,p_j)(x_i-x_j)^2$ 去逼近 $W(\pi)=\sum_{i,j\in\{1,\cdots,n\}} d(p_i,p_j)(\pi(p_i)-\pi(p_j))^2$，其中用连续变量 x_i 去逼近离散变量 $\pi(p_i)$，但 $f(x)$ 仍拥有 $W(\pi)$ 的基本结构。注意当 x_i，$i\in\{1,\cdots,n\}$ 没有约束条件时 $f(x)$ 并没有最大值。因此我们添加约束条件 $x_1^2+x_2^2+\cdots+x_n^2=1$。但是此时当 $f(x)$ 取最大值时 x 没有唯一的取值(因为当对所有的 x_i 加或减一个常数时 $f(x)$ 的值并不改变)。为了避免这种不确定性，我们添加另外一个限制条件 $x_1+x_2+\cdots+x_n=0$

现在这个问题变成了：

求 $f(x)=\sum_{i,j\in\{1,\cdots,n\}} d(p_i,p_j)(x_i-x_j)^2$ 的最大值，其中 $\sum_{i=1}^{n} x_i=0$ 与 $\sum_{i=1}^{n} x_i^2=1$，$p_i,p_j\in P$，$d(p_i,p_j)$ 指 p_i 与 p_j 间的欧式距离。

更进一步我们将 $f(x)$ 重写成 $f(x)=x^TL_Ax$，其中

$$L_A=\begin{bmatrix}\sum_{i\neq1}d(p_1,p_i) & -d(p_1,p_2) & \cdots & -d(p_1,p_n)\\ -d(p_2,p_1) & \sum_{i\neq2}d(p_2,p_i) & \cdots & -d(p_2,p_n)\\ \vdots & \vdots & \ddots & \vdots\\ -d(p_n,p_1) & -d(p_n,p_2) & \cdots & \sum_{i\neq n}d(p_n,p_i)\end{bmatrix}$$

为对称矩阵

$$A=\begin{bmatrix}d(p_1,p_1) & d(p_1,p_2) & \cdots & d(p_1,p_n)\\ d(p_2,p_1) & d(p_2,p_2) & \cdots & d(p_2,p_n)\\ \vdots & \vdots & \ddots & \vdots\\ d(p_n,p_1) & d(p_n,p_2) & \cdots & d(p_n,p_n)\end{bmatrix}$$

的拉普拉斯矩阵。并且对于 L_A 的任意特征值 λ 和给定的特征向量 x 有

$$L_Ax=\lambda x$$

$$f(x)=x^TL_Ax=\lambda x^Tx$$

$$f(x)=x^TL_Ax=\lambda(\sum_{i=1}^{n}x_i^2)$$

$$f(x)=x^TL_Ax=\lambda$$

显然 L_A 是一个对称矩阵(因为对任意 $p_i,p_j\in P$ 均有 $d(p_i,p_j)=d(p_j,p_i)$)，因此对 L_A 的任意特征向量 x，$\sum_{i=1}^{n}x_i=0$ 成立(容易用线性代数证明)，这满足前面的约束条件。对于任意特征值 λ，如果 x 是 λ 的一个特征向量，则 $kx(k\in R)$ 也是 λ 的特征向量。这意味着我们可以对任意特征值 λ 找到满足第二个限制条件 $\sum_{i=1}^{n}x_i^2=1$ 的特征向量。因此对于 L_A 任意的特征值均可找到满足以上两个限制条件的特征向量。

上面的讨论意味着使 $f(x)$ 最大的解就是对应于 L_A 的最大特征值 λ，而所求的 π 即是最大特征值对应的特征向量 x，按照递增的顺序将其所有分量 x_i 排序后各分量的位置和排序前位置的映射。

最大特征值是否是重根并不重要，因为当最大特征值不是重根时，所有线性无关的特征向量 x 都是解，它们都使 $f(x)$ 取得最大值。

同样如果特征向量 x 中存在 $x_{i1}=x_{i2}\cdots=x_{ik}$，$i_1,\cdots,i_k\in\{1,\cdots,n\}$ 也没关系。因为这样意味着 $p_{i1},\cdots,p_{ik}$ 的内部顺序并不影响 $f(x)$ 的值，所以 $p_{i1},\cdots,p_{ik}$ 的顺序可以是任意的。

下面给出相关的光谱算法。

输入：P，n 个点的集合。

输出：order，一个包含 1 到 n 的整形数组。

第一步：计算 P 中 n 个点的距离矩阵 A

第二步：计算的 A 拉普拉斯矩阵 L_A

第三步：计算 L_A 最大特征值的特征向量 x

第四步：对特征向量 x 排序并生成索引数组 order，其中对 $\forall$ i，j，若 i>j 则 $x_{order[i]}\geq x_{order[j]}$。

第五步：返回 order。

分析一下该算法的时间复杂度与空间复杂度。第一步的时间复杂度和空间复杂度均为 $O(n^2)$，第二步的时间复杂度与空间复杂度也均为 $O(n^2)$，第三步中计算特征值与特征向量花费的时间最多，我们将在后面单独讨论，第四步的时间复杂度为 $O(n\log n)$，空间复杂度为 $O(n)$，第五步的时间复杂度为 $O(n)$，空间复杂度为 $O(l)$。

现在讨论步骤三。计算特征值和特征向量的算法可以分成两种主要类型，一个是稳定的直接的方法，它的时间复杂度为 $O(n^3)$，另一种是迭代的方法，该类方法每次迭代的时间复杂度为 $O(n^2)$，但迭代次数是不确定的(但是一般都小于 n)，计算特征值与特征向量的空间复杂度均为 $O(n^2)$。

因此可得到光谱算法总的时间复杂度为大于 $O(n^2)$ 且小于等于 $O(n^3)$，空间复杂度为 $O(n^2)$。

值得注意的是该算法容易扩展到高维的情况。只需简单地将二维欧式距离公式 $d(p_i,p_j)=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$ 替换为 n 维欧氏距离公式 $d(p_i,p_j)=\sqrt{\sum_{k=1}^{n}(x_i^k-x_j^k)^2}$ 即可，其中 $(x_i^1,x_i^2,\cdots,x_i^n)$ 和 $(x_j^1,x_j^2,\cdots,x_j^n)$ 分别为 p_i 和 p_j 在 n 维空间中的坐标。

5 实验结果

本节用具体的实验验证该算法的有效性。该节的实验运行环境为：Windows XP，PentiumD 2.8GHz，512MB ram。

实验中所有测试点都是随机生成的。不失一般性，将 x 和 y 的取值范围限制在 0 到 100 之间方便查询框的产生。

在不同的实验中块因子(Block Factor)有所不同，其中块因子是指一个块包含的记录数。这个参数用来模拟不同的应用，有些应用中记录较大，每块能存放的记录数就较小，相应的有些应用记录较小，每块存放的记录数就较大。在本节实验中块因子取 5 和 50 两个值。

对于每个实验产生的数据文件，执行几个查询集合，每个查询集合包含 1000 个随机产生的查询矩形框。且每个查询集合中查询矩形框的面积是常数，作为实验的一个参数。在实验中选择总面积(大小为 100 * 100)的 1/10，1/20，1/100，1/1000，1/10000 五个不同大小作为这个参数的值(即大小分别为 1000，500，100，10，1)。另外每组实验数据上执行的查询集合是一样的。

为了评估实验结果，选择了以下参数作为参考：平均访问块数(B_v)、平均访问块距和访问块间最大距离($\langle A,M\rangle$)、查询效率(QE)。平均访问块数是指平均每次查询所必须访问块的数目，即这些块中都含有满足查询的记录。访问块间最大距离是指访问的所有的块在线性序列中相距最远的两个块

的距离，它能在一定程度上代表一次查询中磁头最多需要移动的距离。平均访问块距是指一次查询中所有需要访问的块之间的平均距离。以上两个参数都不能精确地说明问题，但可以从整体上反映记录的聚簇情况。另外查询效率 QE 的定义如下：

$$QE=\frac{R_q/R_{total}\times B_{total}}{B_v}$$

其中 R_q 是一次查询得到的记录数，R_{total} 是数据文件中的记录总数，B_v 是访问块数。对于一个测试用例 QE 值越大，该算法性能也就越优。注意以下表中所给出的参数值都是 1000 次查询的平均值。$W(\pi)$ 的值根据算法运行之后生成的线性序列的 π 求得。

在光谱算法中我们用幂乘法(power method)来计算矩阵特征值和特征向量，这是一个得到广泛应用的迭代算法(需注意的是幂乘法所求的是绝对值最大的特征值，而在我们的应用中 $\lambda=f(x)=\sum_{i,j\in\{1,\cdots,n\}}d(p_i,p_j)(x_i-x_j)^2\geqslant 0$，这意味着特征值都是非负的，因此幂乘法求得的最大特征值就是我们所需的)。

在第一组测试中我选取的块因子为 5，记录总数为 10000。

表 1　记录总数：10000，块因子：5

区域大小	文件类型	初始文件	经 SA 聚簇
	W(π)	$8.69*10^{16}$	$1.21*10^{17}$
	B_v	1.609	1.592
1	⟨A,M⟩	⟨137.6,334.3⟩	⟨2.2,5.8⟩
	QE	0.200	0.202
	B_v	10.108	9.488
10	⟨A,M⟩	⟨172.0,1609.1⟩	⟨7.5,73.7⟩
	QE	0.200	0.214
	B_v	98.330	82.212
100	⟨A,M⟩	⟨19.2,1959.2⟩	⟨2.4,274.6⟩
	QE	0.204	0.244
	B_v	446.708	319.005
500	⟨A,M⟩	⟨3.6,1991.5⟩	⟨0.8,597.6⟩
	QE	0.221	0.309
	B_v	798.464	534.579
1000	⟨A,M⟩	⟨1.6,1995.9⟩	⟨0.5,821.3⟩
	QE	0.243	0.364

由表 1 的实验结果可知，无论是哪种情况下，经 SA 算法聚簇后的文件在范围查询时物理块访问数量较小，QE 值较大，这个领先优势随着区域查询的大小而增加。值得注意的是经过聚簇的文件的 $\langle A, M\rangle$ 值均小于未经聚簇的文件。也就是说经过聚簇的文件中相关的记录是很靠近的。当查询区域较小时，这种差异就变得更为明显。以查询区域大小为 1 为例，经过聚簇的文件的 $\langle A,M\rangle$ 值分别为 ⟨2.2, 5.8⟩，而未聚簇文件的 $\langle A,M\rangle$ 值则为 ⟨137.6, 334.3⟩，这说明经聚簇文件在一次查询中磁头移动的距离要远小于未聚簇文件。

在第二组测试中我选取的块因子为 50，记录总数仍为 10000。

由表 2 的实验结果可知，在块因子较大时，经 SA 算法聚簇后的文件在所有查询区域大小下的表现仍然优于原始文件。同样经过聚簇的文件的 $\langle A, M\rangle$ 值均优于未经聚簇的文件。且通过这两个测试可以看到 SA 的 $W(\pi)$ 值要远大于未经聚簇的文件的 $W(\pi)$ 值，与假设吻合，说明 $W(\pi)$ 值能很好地反映文件范围查询的优劣。

表 2　记录总数：10000，块因子：50

区域大小	文件类型	初始文件	经 SA 聚簇
	W(π)	$8.71*10^{16}$	$1.18*10^{17}$
	B_v	1.593	1.373
1	⟨A,M⟩	⟨13.8,33.3⟩	⟨0.1,0.7⟩
	QE	0.020	0.023
	B_v	9.677	6.017
10	⟨A,M⟩	⟨16.6,158.5⟩	⟨0.6,8.7⟩
	QE	0.020	0.033
	B_v	77.927	29.367
100	⟨A,M⟩	⟨1.6,196.1⟩	⟨0.1,33.0⟩
	QE	0.025	0.067
	B_v	182.489	73.902
500	⟨A,M⟩	⟨0.1,198.9⟩	⟨0.0,76.4⟩
	QE	0.053	0.132
	B_v	197.782	102.565
1000	⟨A,M⟩	⟨0.0,199.0⟩	⟨0.0,104.1⟩
	QE	0.097	0.188

下一个测试被用来比较在一维的查询中各个算法的表现。表 3 给出了仅对 x 轴进行范围查询的结果，由于各个记录的值是随机生成的，所以对 x 轴上的范围查询应该和对 y 轴上的范围查询结果相似。故在此仅仅比较对 x 的范围查询。

在一维的查询中，原来的查询区域大小被 x 轴上不同长度代替。

表 3　记录总数：10000 块因子：5

X 长度	文件类型	初始文件	经 SA 聚簇
	B_v	97.912	94.989
1	⟨A,M⟩	⟨19.2,1956.0⟩	⟨12.7,1293.2⟩
	QE	0.204	0.210
	B_v	453.375	414.288
5	⟨A,M⟩	⟨3.4,1992.7⟩	⟨2.3,1387.3⟩
	QE	0.221	0.242
	B_v	818.9	722.712
10	⟨A,M⟩	⟨1.4,1996.4⟩	⟨1.1,1517.3⟩
	QE	0.244	0.277
	B_v	1345.082	1173.169
20	⟨A,M⟩	⟨0.5,1998.2⟩	⟨0.4,1695.2⟩
	QE	0.298	0.341
	B_v	1934.468	1784.057
50	⟨A,M⟩	⟨0.0,1999.0⟩	⟨0.1,1923.9⟩
	QE	0.515	0.559

通过表 3 同样可以看到在单维的范围查询中，经过 SA 算法聚簇后的文件仍然是比原有文件高效的。

结束语　在本文中，通过对文件中的记录安排一个较好的顺序以提高多属性区域查询的效率。首先为这个问题构造一个数学模型，然后引入光谱算法(SA)尝试近似地解决这个 NP 难问题。通过实验，展示了 SA 算法无论是在二维范围查询还是单维范围查询上的有效性。

将来的工作是进一步利用实际应用中的数据进行测试，并对算法进行修改以便能应用于动态数据库。

参考文献

[1] Nianlong Y. Reducing application load time by rearranging disk data[D]. Brigham Young University, Provo, UT, 1998

[2] Omiecinski E, Scheuermann P. A global approach to record clustering and file reorganization[R]. Northwestern University, Evanston, IL, 1983

[3] Jong-Hak L, Young-Koo L, Kyu-Young W, et al. A region

splitting strategy for physical database design of multidimensional file organizations[C]// Proceedings of the 23rd International Conference on Very Large Data Bases. August 1997:416-425

[4] Lee J H, et al. A physcal database design method for multidimensional file organizations [R]. CS/TR-96-104. Korea Advanced Institute of Science and Technology, 1996

[5] Sam S L, Bishwaranjan B. Automated design of multidimensional clustering tables for relational databases[C] // Proceedings of the 30th VLDB Conference. Toronto, Canada, 2004

[6] Padmanabhan S, Bhattacharjee B, Malkemus T, et al. Multi-dimensional clustering: A new data layout scheme in DB2[C]// SIGMOD2003. San Diego, USA, 2003

[7] Yu C T, Suen Cheing-mei, Lam K, et al. Adaptive record clustering[J]. ACM Transactions on Database Systems (TODS), 1993, 10(2): 180-204

[8] Chang J, Fu K. Extended K-d tree database organization: A dynamic multiattribute clustering method[J]. IEEE Softw. Eng., 1981: 284-290

[9] Robinson J T. The K-D-B-tree: a search structure for large multidimensional dynamic indexes[C] // Proceedings of the 1981 ACM SIGMOD International Conference on Management of Data. Ann Arbor, Michigan, April 29-May 01, 1981

[10] George A, Pothen A. An analysis of spectral envelope-reduction via quadratic assignment problems[J]. SIAM J. Matrix Anal. Appl., 1997, 18: 706-732

[11] Adolphson D, Hu T C. Optimal linear ordering[J]. SIAM journal on Applied Mathematics, 1973, 25(3): 403-423

[12] Diaz J, et al. Linear orderings of random geometric graphs[C]// Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science. Ascona, Switzerland, June 1999

[13] Diaz J, Petit J, Serna M. A survey on graph layout problems [J]. ACM Comput. Surveys, 2002, 34(3): 313-356

[14] Petit J. Experiments on the minimum linear arrangement problem[J]. ACM J. Experimental Algorithmics, 2003(8)

[15] Atkins J E, Bowman E G, Hendrickson B. A spectral algorithm for seriation and the consecutive ones problem, SIAM J. Computing, 1998, 28(1): 297-310

[16] Juvan M, Mohar B. Optimal linear labelings and eigenvalues of graphs[J]. Disc. Appl. Math., 1992, 36: 153-168

[17] Barnard S T, Pothen A, Simon H. A spectral algorithm for envelope reduction of sparse matrices[J]. Numerical Linear Algebra with Applications, 1993, 2(4): 317-334

[18] Helmberg C, Rendl F, Mohar B, et al. A spectral approach to bandwidth and separator problems in graphs[J]. Linear and Multilinear Algebra, 1995, 39(1/2): 73-90

(上接第 132 页)

参 考 文 献

[1] Childs S. Information Classifition: The Cornerstone to Information Management [EB/OL]. 2006. http://www.snia.org/education/tutorials/2007/fall/data-management/SheilaChilds_Information_Classification.pdf

[2] Li Yin, Uttamchandani S, Palmer J, et al. AutoLoop: Automated Action Selection in the Observe-Analyze-Act Loop of Storage Systems[C]//Proceeding of The 6th International Workshop on Policies for Distributed Systems and Networks (POLICY'05). Washington, DC: IEEE Computer Society, April 2005: 129-138

[3] Mesnier M, Thereska E, Ganger G R, et al. File classification in self-* storage systems[C]// Proceeding of the First International Conference on Autonomic Computing (ICAC-04). Washington, DC: IEEE Computer Society, May 2004: 44-51

[4] Ganger G R, John D S, Klosterman A J. Self-* Storage: Brick-based Storage with Automated Administration [R]. Carnegie Mellon University: CMU-CS-03-178, August 2003

[5] Ellard D, Mesnier M, Thereska E, et al. Attribute-Based Prediction of File Properties[R]. Harvard Computer Science Group: TR-14-03. December 2003

[6] Lin Qiao, Iyer B R, Agrawal D, et al. PulStore: Automated Storage Management with QoS Guarantee in Large-scale Virtualized Storage Systems[C] // Proceeding of IEEE International Conference on Autonomic Computing (ICAC). Washington, DC: IEEE Computer Society, March 2005: 302-303

[7] Wang Mengzhi, Au K, Ailamaki A, et al. Storage Device Performance Prediction with CART Models[C]//Proceeding of the 12th MASCOTS Volendam. October 2004: 588-595

[8] Sundaram V, Shenoy P. Efficient Data Migration for Load Balancing Large-scale Storage Systems[R]. TR05-51. July 2005

[9] Sundaram V, Wood T, Shenoy P. Efficient Data Migration in Self-managing Storage Systems[C]// Proceeding of Autonomic Computing. June 2006: 297-300

[10] Joseph H, Hartline J D, Karlin A R, et al. On algorithms for Efficient Data Migration [C] // Symposium on Discrete Algorithms. 2001: 620-629

[11] Khuller S, Kim Y, Wan Y. Algorithms for data migration with cloning[C]// ACM Symp. on Principles of Database Systems. 2003

[12] Lu Chenyang, Alvarez G A, John W. Aqueduct: Online Data Migration with Performance Guarantees[C]// Proceedings of File and Storage Technologies'02, USENIX Association. January 2002: 219-230

[13] Li Huaiyang, Xie Changsheng, Cai Bin, et al. A New Technique for Eliminating Data Migration in Logistic Evolution Storage System[C]// Proceeding of the Fifth International Conference on Computer and Information Technology (CIT'05). 2005: 327-333

[14] Chen Ying. Information Valuation for Information Lifecycle management[C]// Proceeding of the 2nd International Conference on Autonomic Computing. Washington, DC: IEEE Computer Society, June 2005: 135-146

[15] Xiong Muzhou, Jin Hai, Wu Song. Information Lifecycle Management in Multi-Gird Environment[C]//第十四届全国信息存储技术学术会议论文集. 北京:电子工业出版社, 2006: 175-184

[16] Maria E G, Santonja V. A New Approach in the Modeling and Generation of Synthetic Disk Workload[C]// Proceeding of 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOT'00). Washington, DC: IEEE Computer Society, August 2000: 199-206

[17] 冯泳, 张延园. 数据迁移在 SAN 中性能优化的研究和应用[J]. 计算机工程, 2005, 31(7): 43-45