空间数据库（6）

陈斌
gischen@pku.edu.cn
2015.11.12

空间存储和索引

- 目标和基本思想
- 物理存储介质
- 缓冲区管理
- 存储组织
- 存取路径：索引结构

索引

- 索引：支持对于所要求的数据进行快速定位的附加的数据结构。
- 每个索引结构有一个特定的搜索码与之关联。
  - 索引按一定的方式存储搜索码的值，并将搜索码与包含该搜索码的记录关联起来。
- 搜索码：用于在文件中查找记录的属性或属性集。

基本索引结构

- 顺序索引
  - 索引基于对搜索码值的一种排序
- 散列索引
  - 索引基于将搜索码值平均分布到若干散列桶（hash buckets）中
- 内外存索引优化策略不同
  - 内存索引偏向减少空间需求，对速度不敏感
  - 外存索引偏向减少访外次数，对速度敏感

基本索引结构：顺序索引

- 顺序索引中按照一定的顺序存储搜索码的值
  - 主索引：若文件中的记录按照某个搜索码值的顺序来存储，则这个搜索码所对应的索引称作主索引，或者聚类索引（cluster index）
  - 辅助索引：索引对应的搜索码值的顺序与文件记录的存储顺序不一致，也称作非聚集索引

基本索引结构：散列索引

- 在外存中按照桶散列，通过散列函数将搜索码值对应到桶地址
  - 桶（bucket）是能存储一条或多条记录的一个存储单位，每个桶包括一个或多个磁盘块
- 散列牺牲存储效率
  - 可以通过可扩充散列，在数据库大小变化时对桶进行分裂或合并，保持一定的空间效率

对索引技术评价的考虑

- 访问类型
  - 能有效支持的数据库访问的类型；
- 访问时间
  - 访问一个或多个数据项所需的时间；
- 插入时间
  - 在索引中插入一个新数据项所需的时间；
- 删除时间
  - 从索引中删除一个数据项所需的时间；
- 空间开销
  - 索引结构所需的额外的存储空间。

聚类（cluster）

- 以某种搜索码值的顺序安排记录的物理存储
  - 搜索码值相近的记录在存储上也相近
    - 表现在磁道和扇区上的相邻
  - 降低对于常见的大查询的响应时间
    - 单搜索码值的查找，范围值的查找
    - 降低寻道时间和寻扇区时间
    - 提高磁盘缓存的命中率

聚类

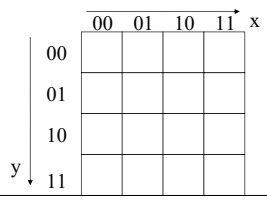
- 简单数据类型的聚类
  - 整数、定点数、浮点数
  - 字符串、日期
  - 具有完整的一维全序性质，其值可以排成线性单调序列，和存储器的线性性质相符
- 复杂数据类型的聚类
  - 两维以上的简单数据类型的组合向量
  - 如空间数据、多搜索码的结构

聚类

- 多维数据类型的聚类方法
  - 将高维地址空间映射到一维地址空间
    - 一一对应的映射，保证没有地址遗漏和重复
    - 保持距离的映射，保证高维中相邻的地址也在一维中相邻
  - 一一对应的映射容易构造
  - 保持距离只能近似的实现
    - Z序映射和Hilbert曲线映射

二维空间聚类

- 考虑有限二维整数平面
  - 以每次四分网格的形式递归划分平面
  - 递归划分的层次决定坐标的二进制位数
  - 每个网格具有唯一的二维坐标作为地址



两次递归划分的网格

Z序映射

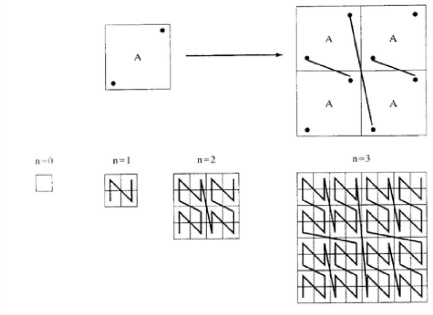


FIGURE 4.4. Generating a Z-curve [Asano et al., 1997].

Z序映射编码

- 读入x和y坐标的二进制表示；
- 隔行扫描二进制位到一个字符串；
- 计算出结果二进制串的十进制值。

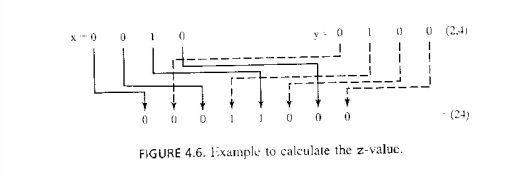
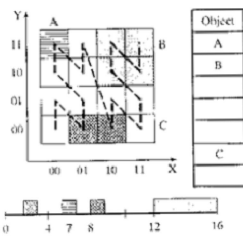


FIGURE 4.6. Example to calculate the z-value.

Z序映射编码例子



Object	Points	x	y	interleave	z-value
A	1	00	11	0101	5
B	1				
	2				
	3				
	4				
C	1	01	00	0010	2
	2	10	00	1000	8

FIGURE 4.7. A trace for finding z-values.

Hilbert曲线映射

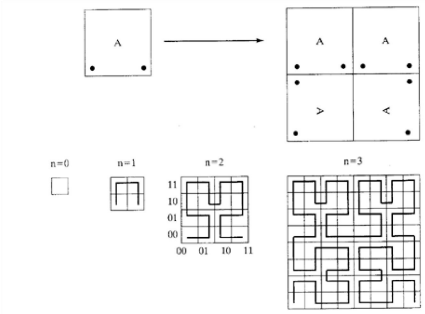


FIGURE 4.5. Generating a Hilbert curve [Asano et al., 1997].

Hilbert曲线映射编码

- 读入x和y坐标的二进制表示；
- 隔行扫描二进制位到一个字符串；
- 将字符串从左到右分成若干2位长的串s_i(i=1..n)，并将其换成规定的十进制数
  - 00->0, 01->1, 10->3, 11->2
- 对十进制数进行替换
  - 01->03, 03->01, 30->32, 32->30
- 再将十进制表示换成二进制表示，连接后计算十进制值，得到一维的地址

Hilbert曲线映射编码例子

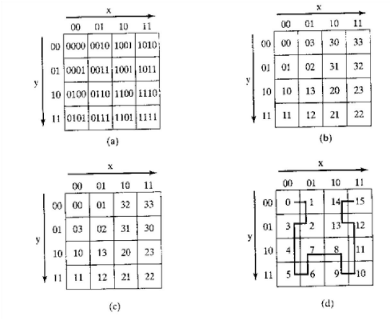


FIGURE 4.8. Example showing Hilbert curve translation.

聚类的磁盘访问性能

- 基本假定
  - 有限范围的多维空间，有限个网格单元
  - 映射将多维空间的单元指定一个整数地址
  - 每个网格单元对应一个磁盘页面的存储
  - 连续地址的单元存储在相邻磁盘页面
- 性能衡量指标
  - 对一片连续空间范围网格的访问涉及尽量少间断的磁盘页面

聚类的磁盘访问性能

- Hilbert曲线映射的聚类性能比Z序映射稍好，但算法更为复杂

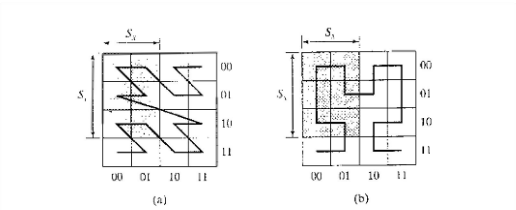
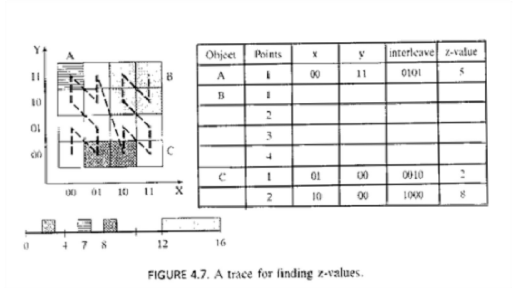


FIGURE 4.9. Illustration of clusters: (a) two clusters for the Z-curve; (b) one cluster for the Hilbert curve.

连续区域的聚类表示

- 具有多维坐标的单个网格单元可以通过映射编码直接得到聚类存储单元的地址，对应点查询的情形
- 一片连续区域的网格，对应范围查询的情形
  - 某些连续区域包含的网格单元具有共同的编码前缀
  - 任意的连续区域需要拆分成几片上述性质的区域
  - 通常采用近似的方法减少拆分片数提高效率

连续区域的聚类表示



Object	Points	x	y	interleave	z-value
A	1	00	11	0101	5
B	1				
	2				
	3				
	4				
C	1	01	00	0010	2
	2	10	00	1000	8

FIGURE 4.7. A trace for finding z-values.

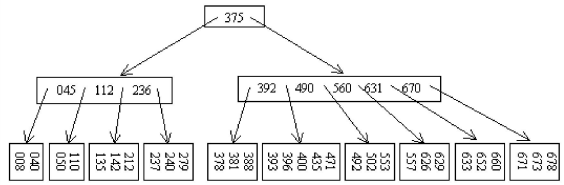
一维搜索码的索引

- B树与B+树
  - 多叉树，分支数量受到上下限的限制
  - 平衡树，子树的层次差受到限制
- 区别
  - 内部节点是否存储实际的搜索码值
  - 是否允许顺序索引

一维搜索码的索引：B树

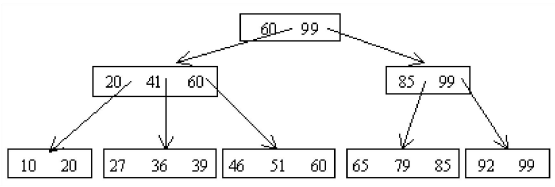
B树的示例

下图所示的是一棵6阶的B树。



一维搜索码的索引：B+树

- B+树的分支结点上关键码与指向子女的指针总是成对出现，仅记录子节点最大关键码，称为分界值关键码



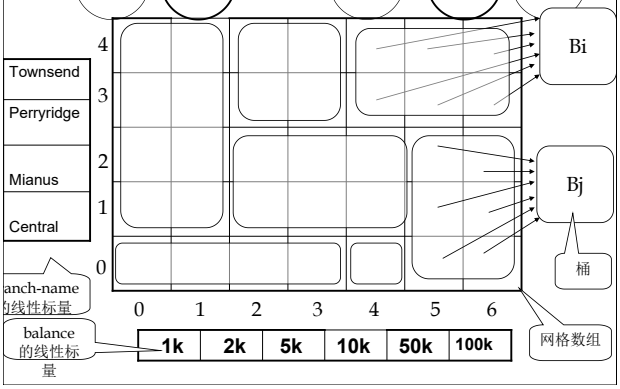
多维索引

- 类似散列表的结构
  - 网格文件
  - 分段散列
- 基于树形的结构
  - 四分树
  - R树

网格文件

- 通过对每个维的值进行排序，将搜索码值“散列”到桶中
- 结构
  - 一个网格数组，其每个单元包含一个指向桶的指针，可以有多个单元指向同一个桶。
  - 每个维一个线性标量，对该维的值进行划分。由线性标量确定一个多维搜索码值应该落到网格数组的哪一个单元中。

网格文件



网格文件

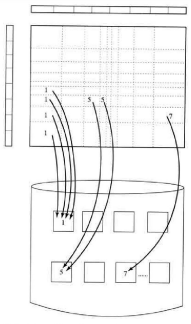


FIGURE 4.13. A two-dimensional grid directory and data pages.

网格文件：特性

- 适合于多码查询，也可以回答包含一个搜索码的查询。
- 支持范围查询
- 线性标量的选择必须使记录在单元格中的分布是均匀的。
- 如果桶满了还要插入新的记录，则要分配一个新桶，并进行指针的调整和记录的重新分布，或把新桶作为溢出桶。

网格文件

- 优点：大大减少了多码查询的处理时间。
- 缺点：增加了空间开销，和记录插入和删除的开销。此外，很难选择适当的、保证记录均匀分布的分段范围。

分段散列

- 对散列的扩充，以对多个属性进行散列。
- 散列函数产生k个二进制位，这k位在n个属性中进行划分，设为第i个属性产生ki位散列值，于是k1 + k2 + ... + kn=k。
- 更精确地说，散列函数h实际上是一组散列函数（h1，h2，...，hn），其中每个hi运用到第i个属性上且产生ki位二进制位序列。
- 进行散列时，在这n个属性上值为（v1，v2,...,vn）的元组所属的桶通过拼接二进制序列h1(v1)h2(v2)...hn(vn)计算得到。

分段散列

- 例 搜索码(customer-street，customer-city)的分段散列函数

search-key value	hash value
(Main，Harison)	101 111
(North，Rye)	110 101
(Main，Brooklyn)	101 001
(North，Princeton)	110 000
(Park，Palo Alto)	010 010
(Putnam，Stamford)	011 001
(Nassau，Princeton)	011 000
(Spring，Brooklyn)	001 001
(Alma，Palo Alto)	110 010

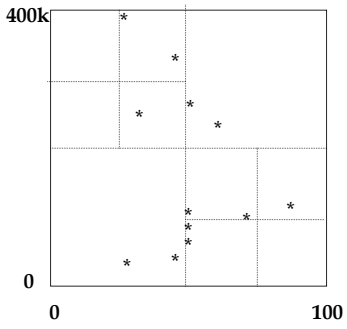
分段散列

- 适合于多码查询
- 也可以方便地回答包含一个搜索码的查询
- 但不支持范围查询

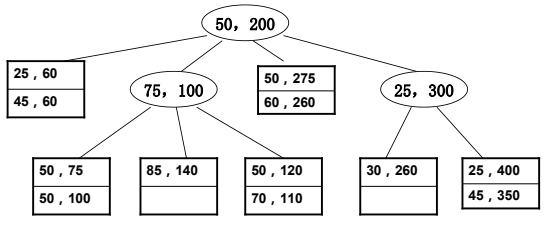
四分树

- 四分树的每个内部节点对应于二维空间中的一个正方形区域，或是K维空间的K维立方体，外部结点对应于存放空间中点的块
- 以二维的情形为例，
  - 如果一个正方形中的点数不比一个块中能存放的数多，那么我们就把这个正方形看作树的叶结点，该结点就表示成存放它的点的块；
  - 如果矩形中还有太多的点以至于一个块存放不下，那么我们把这个正方形看作内部结点，它的子结点对应于它的四个象限

四分树



四分树



R树

- R-树是平衡树的结构，非常象B-树。主要用于对矩形和其它多边形的索引。
- R-树的每个树结点对应一个矩形边界框，多边形只存在叶结点上。
- 叶结点的边界框是包含叶结点中所有对象的最小矩形MBR
  - 类似地内部结点的边界框是包含其子结点的边界框的最小矩形。

R树

- MBR空间范围可以相交，但所有对象仅属于一个MBR

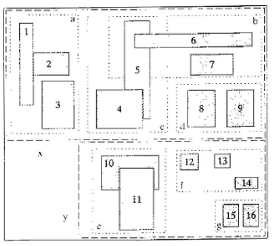


FIGURE 4.15. A collection of spatial objects.

R树

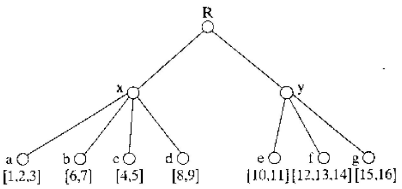


FIGURE 4.16. R-tree hierarchy.

R+树

- 所有MBR不相交，但对象可以属于多个MBR

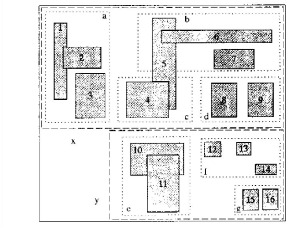


FIGURE 4.18. Rectangles for interior nodes of a R+ tree.

R+树

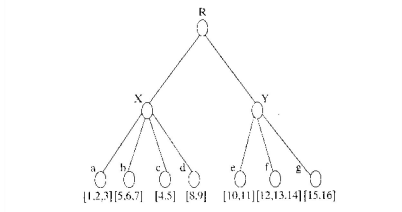


FIGURE 4.17. R+ tree hierarchy.

讨论：NoSQL中的空间数据存储组织

- NoSQL：泛指非关系型的数据库
- Key-Value pair：Redis
- 列存储：HBase，Cassandra
- 文档型：MongoDB，CouchDB
- 图Graph数据库：Neo4J
- 11月26日

参考文献

- [TP311.13/261]空间数据库 = Spatial databases a tour (美) Shashi Shekhar, Sanjay Chawla著 谢昆青 ... 等译 北京 机械工业出版社 2004
- 北京大学计算机系数据库教研室
  - 数据库原理与技术讲义（杨冬青）
- 数据库系统概论
  - 岳丽华，丁卫群 编著
  - 科学出版社，2000年