



(12) 发明专利申请

(10) 申请公布号 CN 103714145 A

(43) 申请公布日 2014. 04. 09

(21) 申请号 201310729418. 1

(22) 申请日 2013. 12. 25

(71) 申请人 中国地质大学(武汉)

地址 430074 湖北省武汉市洪山区鲁磨路  
388 号

(72) 发明人 王勇 张亮 薛思清 廖洪艳  
刘珍伶

(74) 专利代理机构 武汉华旭知识产权事务所  
42214

代理人 江钊芳

(51) Int. Cl.

G06F 17/30(2006. 01)

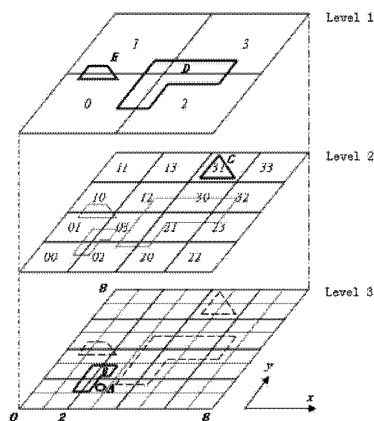
权利要求书2页 说明书7页 附图3页

(54) 发明名称

关系型和 Key-Value 型数据库空间数据索引方法

(57) 摘要

本发明涉及一种同时适用于关系型和 Key-Value 型数据库的空间矢量数据的索引方法。空间数据按分类组织成图层,根据图层数据的坐标范围划分出索引层和索引网格,每个网格赋予唯一的二维行列编码和一维编码;遍历图层的每个要素,根据要素的图形计算其与索引网格之间的覆盖关系,从而获得要素所属的索引层和覆盖的网格;将要素所属的网格的行列编码作为要素的空间索引编码存储在关系数据库的图层索引表中,网格的一维编码经复合设计后作为 Key-Value 型数据库中的图层索引表的行键。本发明统一了关系型和 Key-Value 型数据库中矢量空间数据的索引方式,当空间数据在它们之间迁移时,不需要重建空间索引。



1. 一种关系型和 Key-Value 型数据库空间数据索引方法,其特征在于:操作步骤如下:

步骤一:将空间矢量数据组织成图层,根据图层数据的坐标范围划分出索引层和索引网格,每个网格赋予唯一的二维行列编码和一维编码,具体为:

(a) 空间矢量数据按分类组织成图层,根据图层的坐标范围或图层所在的参考系的坐标范围划分索引层,索引层最多为 24 层,最上面的为 0 号索引层,只有 1 个网格,表示整个坐标范围,1 号索引层有 4 个网格,是在 0 号索引层的网格上四等分得到的,以此类推,N 号索引层的网格是在 N-1 号索引层的每个网格上四等分得到的,最底层为 23 号索引层,有  $2^{23} \times 2^{23}$  个网格;

(b) 根据索引层上的网格所在的行和列,赋予每个网格唯一的二维行列编码;

(c) 1 号索引层的 4 个网格的二维行列编码可以变换为一维编码 0、1、2、3,将 1 号索引层的每个网格继续四等分,得到 2 号索引层的网格,其一维编码由 1 号索引层的一维编码附加 2 号索引层网格 0, 1, 2, 3 编码组成,其余索引层的网格都按此规则赋予唯一的一维编码;

步骤二:遍历图层的每个要素,根据要素的图形计算其与索引网格之间的覆盖关系,从而获得要素所属的索引层和覆盖的网格,具体为:

①所述的要素记为  $s$ ,当  $s$  为点图形时,且  $s$  在某个网格内,则表示为  $s$  覆盖了该网格,当  $s$  为线或面图形时,且  $s$  与某网格相交,则表示为  $s$  覆盖了该网格, $s$  覆盖的全部网格坐标范围记为  $GBR(X1, Y1, X2, Y2)$ , $s$  覆盖的  $i$  号索引层的网格数记为  $s(i)$ ;

②对于线或多边形图层数据中的任一要素  $s$ ,根据  $s$  的外包矩形坐标范围,从 23 号索引层开始,或者根据  $s$  的尺度特征从小于 23 的某个索引层开始,逐层向上计算  $s(i)$ ,其中  $i > 0$ ;

③当  $s(N) = 4$  时,判定  $s$  在  $N$  号索引层上的必要条件是覆盖最多 4 个网格,如果:

$4 = s(N) > \dots > s(N-k) = s(N-k-1)$ , ( $k=1, 2, \dots, N-1$ ), 则  $s$  在  $N-k$  索引层上,否则  $s$  在  $N$  号索引层上, $N$  最大为 23,最小为 1;

④对于空间点要素图层,其全部点的集合构成点云,以点云的平均密度为 4 的半径  $R$  作为网格参考尺度,可以确定索引层号,然后再逐个确定点在该索引层上所覆盖的网格;

④记录  $s$  的所属索引层号、所覆盖网格和覆盖的全部网格坐标范围  $GBR(X1, Y1, X2, Y2)$ ;

⑤当图层的全部要素遍历结束,记下最大索引层号和最小索引层号,作为图层的有效索引层范围;

步骤三:将要素所属的网格的行列编码作为要素的空间索引编码存储在关系型数据库的图层索引表中,网格的一维编码经复合设计后作为 Key-Value 型数据库中的图层索引表的行键编码,其具体为:

1) 图层要素存储在图层数据表中,包括要素 ID、要素的空间和属性信息;

2) 每个图层数据表对应一个图层索引表,关系型数据库中的图层索引表存储要素索引网格的二维行列编码,Key-Value 型数据库中的图层索引表将索引网格的行键编码作为行键存储;

3) 所有图层的坐标范围和有效索引层范围都存储在一张图层元数据表中。

2. 根据权利要求 1 所述的方法,其特征在于:步骤三中所述的关系型数据库和 Key-Value 型数据库图层索引表,其结构是:

关系数据库图层索引表的字段有要素 ID、索引网格的行和列编码、网格坐标范围 GBR (X1, Y1, X2, Y2),并对它们建立数据库联合索引;

Key-Value 数据库图层索引表的每一行代表唯一的索引网格,包括行键编码、网格的行和列编码、网格的空间坐标范围、覆盖本网格的要素 ID 集合。

3. 根据权利要求 1 所述的方法,其特征在于:步骤三中所述的图层索引表中二维行列编码和行键编码,其编码方式为:

关系型数据库中的图层索引表存储的要素索引网格的行和列编码分别为 4 字节整数,首字节为网格的索引层号,其余 3 个字节表示网格的行和列号;

Key-Value 型数据库中的图层索引表的行键为 8 字节编码,首字节为索引层号,6 个字节存储索引网格的一维编码,1 个字节保留未用。

## 关系型和 Key-Value 型数据库空间数据索引方法

### 技术领域

[0001] 本发明涉及空间数据管理和空间信息处理领域,尤其涉及一种对存储于关系型数据库和 Key-Value 型数据库中的空间矢量数据建立一致的空间索引的方法。

### 背景技术

[0002] 空间数据关注空间地物要素的几何形状,利用点、线和多边形的形式来表达现实世界。随着空间数据采集能力的提高和地理信息系统的发展,空间数据急剧增长,基于 GIS 的大规模数据、高流量负载的网络应用也越来越多。当采用关系型数据库存储这些海量空间数据时,单一节点的数据存储能力很难为用户提供高并发和低延迟的服务。为了提升访问效率,可以通过集群来解决这类问题,但关系型数据库数据模型的复杂性影响其扩展规模。

[0003] Key-Value 型数据库以其在大规模并发访问与海量数据存储方面的优势得到广泛的应用。Key-Value 型数据库没有严格的范式,通常面向列存储,可以在一个服务集群上运行,当有更多的节点加入到集群时,可以方便地进行扩展。集群中的各节点提供了存储、缓存和计算的能力。Key-Value 型数据库比较优秀的代表有 HBase、MongoDB 等,它们提供了易于扩展的超大规模的数据存储能力,同时提供低延迟访问,为海量空间数据的存储提供了极佳的选择。

[0004] 空间数据存储于数据库中,通过空间索引可以快速地检索出满足空间查询条件的空间要素。空间索引的实质是通过某种编码的快速查找达到快速定位空间要素的目的,常用的空间索引方法有网格索引、R 树索引和四叉树索引等。其中网格索引和四叉树索引方法简单、易于实现,被大多数空间数据库采用。在实际应用中,当需要在关系型和 Key-Value 型两种不同类型的空间数据库之间迁移数据时,如果它们的空间索引编码机制不同,就需要重新编码。对于海量空间数据的情形,重建索引编码的代价是相当大的。因此,不同类型的空间数据库采用统一的空间索引方法具有重要实际意义。

### 发明内容

[0005] 本发明的目的是解决上述问题,提出一种对存储在关系型和 Key-Value 型数据库中的海量空间数据创建高效、一致的空间索引方法,为跨类型数据库的空间数据迁移提供便利。

[0006] 为了实现上述目的,本发明所采取的技术方案是:提供一种统一的关系型和 Key-Value 型数据库空间数据索引方法,操作步骤如下:

[0007] 步骤一:将空间矢量数据组织成图层,根据图层数据的坐标范围划分出索引层和索引网格,每个网格赋予唯一的二维行列编码和一维编码,具体为:

[0008] (a) 空间矢量数据按分类组织成图层,根据图层的坐标范围或图层所在的参考系的坐标范围划分索引层,索引层最多为 24 层,最上面的为 0 号索引层,只有 1 个网格,表示整个坐标范围,1 号索引层有 4 个网格,是在 0 号索引层的网格上四等分得到的,以此类推,

N 号索引层的网格是在 N-1 号索引层的每个网格上四等分得到的,最底层为 23 号索引层,有  $2^{23} \times 2^{23}$  个网格;

[0009] (b) 根据索引层上的网格所在的行和列,赋予每个网格唯一的二维行列编码;

[0010] (c) 1 号索引层的 4 个网格的二维行列编码可以变换为一维编码 0、1、2、3,将 1 号索引层的每个网格继续四等分,得到 2 号索引层的网格,其一维编码由 1 号索引层的一维编码附加 2 号索引层网格 0, 1, 2, 3 编码组成,其余索引层的网格都按此规则赋予唯一的一维编码;

[0011] 步骤二:遍历图层的每个要素,根据要素的图形计算其与索引网格之间的覆盖关系,从而获得要素所属的索引层和覆盖的网格,具体为:

[0012] ①所述的要素记为  $s$ ,当  $s$  为点图形时,且  $s$  在某个网格内,则表示为  $s$  覆盖了该网格,当  $s$  为线或面图形时,且  $s$  与某网格相交,则表示为  $s$  覆盖了该网格, $s$  覆盖的全部网格坐标范围记为  $GBR(X1, Y1, X2, Y2)$ , $s$  覆盖的  $i$  号索引层的网格数记为  $s(i)$ ;

[0013] ②对于线或多边形图层数据中的任一要素  $s$ ,根据  $s$  的外包矩形坐标范围,从 23 号索引层开始,或者根据  $s$  的尺度特征从小于 23 的某个索引层开始,逐层向上计算  $s(i)$ ,其中  $i > 0$ ;

[0014] ③当  $s(N) = 4$  时,判定  $s$  在  $N$  号索引层上的必要条件是覆盖最多 4 个网格,如果:

[0015]  $4 = s(N) > \dots > s(N-k) = s(N-k-1)$ , ( $k=1, 2, \dots, N-1$ ), 则  $s$  在  $N-k$  索引层上,否则  $s$  在  $N$  号索引层上, $N$  最大为 23,最小为 1;

[0016] ④对于空间点要素图层,其全部点的集合构成点云,以点云的平均密度为 4 的半径  $R$  作为网格参考尺度,可以确定索引层号,然后再逐个确定点在该索引层上所覆盖的网格;

[0017] ④记录  $s$  的所属索引层号、所覆盖网格和覆盖的全部网格坐标范围  $GBR(X1, Y1, X2, Y2)$ ;

[0018] ⑤当图层的全部要素遍历结束,记下最大索引层号和最小索引层号,作为图层的有效索引层范围;

[0019] 步骤三:将要素所属的网格的行列编码作为要素的空间索引编码存储在关系型数据库的图层索引表中,网格的一维编码经复合设计后作为 Key-Value 型数据库中的图层索引表的行键编码,其具体为:

[0020] 1) 图层要素存储在图层数据表中,包括要素 ID、要素的空间和属性信息;

[0021] 2) 每个图层数据表对应一个图层索引表,关系型数据库中的图层索引表存储要素索引网格的二维行列编码,Key-Value 型数据库中的图层索引表将索引网格的行键编码作为行键存储;

[0022] 3) 所有图层的坐标范围和有效索引层范围都存储在一张图层元数据表中。

[0023] 本发明步骤三中所所述的关系型数据库和 Key-Value 型数据库图层索引表,其结构是:

[0024] 关系数据库图层索引表的字段有要素 ID、索引网格的行和列编码、网格坐标范围  $GBR(X1, Y1, X2, Y2)$ ,并对它们建立数据库联合索引;

[0025] Key-Value 数据库图层索引表的每一行代表唯一的索引网格,包括行键编码、网格的行和列编码、网格的空间坐标范围、覆盖本网格的要素 ID 集合。

- [0026] 本发明步骤三中所述的图层索引表中二维行列编码和行键编码,其编码方式为:
- [0027] 关系型数据库中的图层索引表存储的要素索引网格的行和列编码分别为 4 字节整数,首字节为网格的索引层号,其余 3 个字节表示网格的行和列号;
- [0028] Key-Value 型数据库中的图层索引表的行键为 8 字节编码,首字节为索引层号,6 个字节存储索引网格的一维编码,1 个字节保留未用。
- [0029] 本发明的方法具有的有益效果是:
- [0030] (A) 本发明提出的 24 个索引层划分方法可满足海量空间数据范围的编码需要,这是因为假设待处理的空间数据为整个地球范围(约 40000km×40000km),用  $2^{23} \times 2^{23}$  个网格去划分它,最底下的 23 号索引层的网格尺寸不到 5 米,可为足够小的地物建立索引。
- [0031] (B) 本发明中关系型数据库索引网格的二维行列编码和 Key-Value 型数据库索引网格的一维行键编码具有一一对应的空间映射关系,可以相互变换,当空间数据在它们之间迁移时,只需要转换索引编码,不需要重新计算要素与索引网格的关系。
- [0032] (C) 本发明中二维行列编码采用 4 字节编码,一维行键编码采用 8 字节编码,考虑了 64 位的操作系统中内存 8 字节对齐的特点,可利用操作系统的最佳特性。

#### 附图说明

- [0033] 图 1 为本发明索引分层与网格划分示意图。
- [0034] 图 2 为本发明中网格二维行列编码示意图。
- [0035] 图 3 为本发明中网格一维编码示意图。
- [0036] 图 4 为本发明中网格一维编码原理图。
- [0037] 图 5 为本发明中要素建立空间索引的过程示意图。

#### 具体实施方式

- [0038] 下面结合附图和具体实施例对本发明作进一步的详细说明。
- [0039] 实施例 1:本发明提供一种统一的关系型和 Key-Value 型数据库空间数据索引方法,操作步骤如下:
- [0040] 步骤一:将空间矢量数据组织成图层,根据图层数据的坐标范围划分出索引层和索引网格,下一索引层的网格是在上一索引层的每个网格上四等分得到的,每个网格具有唯一的一维编码和二维行列编码,具体为:
- [0041] 1) 空间矢量数据按分类组织成图层,根据图层的坐标范围或图层所在的参考系的坐标范围划分索引层,索引层为 24 层,最上面的为 0 号索引层,只有 1 个网格,表示整个坐标范围 MBR (Xmin, Ymin, Xmax, Ymax),1 号索引层有 4 个网格,是在 0 号索引层的网格上四等分得到的,以此类推, N 号索引层的网格是在 N-1 号索引层的每个网格上四等分得到的,最底层为 23 号索引层,有  $2^{23} \times 2^{23}$  个网格,如图 1 说明了索引分层与网格划分原理。
- [0042] 2) 记 N 号索引层为 Level N。Level0 只有一个网格,表示整个 MBR。将 Level0 的网格四等分得到的 Level1 的 4 个网格,其行列坐标即二维网格编码分别为:(0,0)、(0,1)、(1,0)、(1,1),参见图 2 的二维网格编码示意图。
- [0043] 3) Level1 的 4 个网格一维编码为:0,1,2,3。将 Level1 的每个网格继续四等分,得到 Level2 的 16 个网格,其一维编码是将 Level1 的网格编码作为前缀附加上其自身网格

编码 0, 1, 2, 3 组成;同样可以得到 Level3 的网格一维编码。图 3,图 4 分别为 Level3 网格一维编码示意图与原理图。其余索引层的网格都按此规则赋予唯一的一维编码。

[0044] 步骤二:遍历图层的每个要素,根据要素的图形计算其与索引网格之间的覆盖关系,从而获得要素所属的索引层和覆盖的网格:

[0045] 1)所述的要素记为  $s$ ,当  $s$  为点图形时,且  $s$  在某个网格  $G$  内,则表示为  $s$  覆盖了  $G$ ,当  $s$  为线或面图形时,且  $s$  与  $G$  相交,则表示为  $s$  覆盖了  $G$ , $s$  覆盖的全部网格坐标范围记为  $GBR(X1, Y1, X2, Y2)$ , $s$  覆盖的  $i$  号索引层 Level  $i$  号的网格数记为  $s(i)$ 。

[0046] 2)对于图层数据表中的任一要素  $s$ ,根据  $s$  的坐标范围  $MBR$ ,从 Level23 开始,或者根据  $s$  的尺度特征从小于 23 的某个索引层开始,逐层向上计算  $s(i)$ ,其中  $i>0$ 。

[0047] 3)对于图层的任一要素  $s$ , $s$  覆盖的 Level  $N$  上的网格数记为  $s(N)$ 。当  $s(N)=4$  时,可以判定  $s$  可能在 Level  $N$  上,即  $s$  在 Level  $N$  上的必要条件是覆盖最多 4 个网格,如果: $4=s(N)>\dots>s(N-k)=s(N-k-1)$ , $(k=1,2,\dots)$ ,则  $s$  所在的 Level 为  $N-k$ , $(k<N)$ ,否则为  $N$ 。规定要素的 Level 最大为 23,最小为 1。

[0048] 为了更好地说明要素建立空间索引的全过程,参见图 5。假定最大 Level 为 3,要求为要素 A~E 建立索引表。要素 A 在 Level3 只覆盖了 1 个网格(2,1),因此 A 的 Level=3;B 在 Level3 和 Level2 上都覆盖了 4 个网格,根据本方法的规定,B 的 Level=3。C 在 Level2 上覆盖的网格数 1,少于其在 Level3 上的网格数 4,因此 C 的 Level=2。D 和 E 都是随着 Level 的提升,覆盖的网格数也随之减少,因此最终 D 和 E 的 Level=1。

[0049] 4)对于空间点要素图层,其全部点的集合构成点云,以点云的平均密度为 4 的半径  $R$  作为网格参考尺度,可确定索引层号,然后再逐个确定点在该索引层上所覆盖的网格。

[0050] 5)记录  $s$  的所属索引层号、所覆盖网格和覆盖的全部网格坐标范围  $GBR(X1, Y1, X2, Y2)$ 。

[0051] 6)当图层的全部要素遍历结束,记下最大索引层号(MaxLevel)和最小索引层号(MinLevel),作为图层的有效索引层范围。

[0052] 步骤三:将要素所属的网格的行列编码作为要素的空间索引编码存储在关系型数据库的图层索引表中,网格的一维行键编码作为 Key-Value 数据库中的图层索引表的行键;

[0053] 1)图层要素存储在图层数据表中,包括要素的空间和属性信息。

[0054] 表 1 是关系型数据库中的图层数据表,其包括要素 ID (主键)、要素的空间列和属性列。其中空间列包括要素的坐标范围(MBR)和要素图形的二进制数据(WKB 格式)。

[0055] 表 1 :关系型数据库中的图层数据表

[0056]

主键 要素 ID	空间列				属性列			
	MBR				Shape	属性字段列表		
	Xmin	Ymin	Xmax	Ymax	WKB	A1	A2	...
<i>A</i>	2.6	1.2	2.9	1.6				
<i>B</i>	1.7	1.1	2.6	2.9				
<i>C</i>	4.3	6.3	5.8	7.9				
<i>D</i>	3.3	1.6	6.7	4.8				
<i>E</i>	0.6	3.6	2.3	4.4				

[0057] 表 2 是 Key-Value 型数据库中的图层数据表, 其以要素 ID 作为行键 (Rowkey), 包括空间列族 S 和属性列族 A, 空间列族 S 包括的列有要素的 MBR、GBR 所属的 Level 和要素图形数据 (WKB 格式)。

[0058] 表 2 :Key-Value 型数据库中的图层数据表

[0059]

Rowkey *	空间列族 S									属性列族 A	
	MBR				GBR				Level	图形	
SID	Xmin	Ymin	Xmax	Ymax	X1	Y1	X2	Y2	Lvl	WKB	...
<i>A</i>	2.6	1.2	2.9	1.6	2	1	3	2	3	...	...
<i>B</i>	1.7	1.1	2.6	2.9	1	1	3	3	3	...	...
<i>C</i>	4.3	6.3	5.8	7.9	2	3	3	4	2	...	...
<i>D</i>	3.3	1.6	6.7	4.8	0	0	2	2	1	...	...
<i>E</i>	0.6	3.6	2.3	4.4	0	0	1	2	1	...	...

[0060] 2) 每个图层数据表对应一个图层索引表, 关系型数据库中的图层索引表存储要素索引网格的二维行列编码。

[0061] 表 3 是关系型数据库中的图层索引表, 其包括要素 ID、要素的二维网格编码 (X1, Y1)、要素覆盖的网格范围 GBR (X1, Y1, X2, Y2), 其中 :X1=Level<<24|Xi, Y1=Level<<24|Yi, 并对这 7 个字段 (Level 除外) 建立数据库联合索引。

[0062] 表 3 :关系型数据库中的图层索引表



[0063]

要素 ID	二维网格编码 (XI, YI)			要素覆盖的网格范围 GBR			
	<i>Level</i>	<i>Xi</i>	<i>Yi</i>	<i>X1</i>	<i>Y1</i>	<i>X2</i>	<i>Y2</i>
<i>A</i>	<i>3</i>	<i>2</i>	<i>1</i>	<i>2</i>	<i>1</i>	<i>3</i>	<i>2</i>
<i>B</i>	<i>3</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>3</i>	<i>3</i>
<i>B</i>	<i>3</i>	<i>1</i>	<i>2</i>	<i>1</i>	<i>1</i>	<i>3</i>	<i>3</i>
<i>B</i>	<i>3</i>	<i>2</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>3</i>	<i>3</i>
<i>B</i>	<i>3</i>	<i>2</i>	<i>2</i>	<i>1</i>	<i>1</i>	<i>3</i>	<i>3</i>
<i>C</i>	<i>2</i>	<i>2</i>	<i>3</i>	<i>2</i>	<i>3</i>	<i>3</i>	<i>4</i>
<i>D</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>2</i>	<i>2</i>
<i>D</i>	<i>1</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>2</i>	<i>2</i>
<i>D</i>	<i>1</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>2</i>	<i>2</i>
<i>D</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>2</i>	<i>2</i>
<i>E</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>2</i>
<i>E</i>	<i>1</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>2</i>

[0064] Key-Value 型数据库中的图层索引表将索引网格的一维行键编码作为行键存储。表 4 是 Key-Value 数据库中的图层索引表, 每一行表示唯一的索引网格 (Xi, Yi), 网格的行列坐标对应的一维编码作为行键, 即 Rowkey=Level<<56|Code, 索引列族 I 包括的列有网格的行列坐标 (Xi, Yi)、网格的坐标范围 MBR 和 SID 列。SID 列存放覆盖了本网格的所有要素 ID, 它们指向了 Key-Value 数据库的图层数据表的 Rowkey。

[0065] 表 4 :Key-Value 型数据库中的图层索引表

[0066]

Rowkey *		索引列族 I						SID
<i>Level</i>	<i>Code</i>	网格坐标		MBR				
<i>Level</i>	<i>Code</i>	<i>Xi</i>	<i>Yi</i>	<i>Xmin</i>	<i>Ymin</i>	<i>Xmax</i>	<i>Ymax</i>	
<i>1</i>	<b>0</b>	0	0	0.0	0.0	4.0	4.0	<i>D, E</i>
<i>1</i>	<b>1</b>	0	1	0.0	4.0	4.0	8.0	<i>D, E</i>
<i>1</i>	<b>2</b>	1	0	4.0	0.0	8.0	4.0	<i>D</i>
<i>1</i>	<b>3</b>	1	1	4.0	4.0	8.0	8.0	<i>D</i>
<i>2</i>	<b>31</b>	2	3	4.0	6.0	6.0	8.0	<i>C</i>
<i>3</i>	<b>003</b>	1	1	1.0	1.0	2.0	2.0	<i>B</i>
<i>3</i>	<b>012</b>	1	2	1.0	2.0	2.0	3.0	<i>B</i>
<i>3</i>	<b>021</b>	2	1	2.0	1.0	3.0	2.0	<i>A, B</i>
<i>3</i>	<b>030</b>	2	2	2.0	2.0	3.0	3.0	<i>B</i>

[0067] 3) 在图层元数据表中记录所有图层的坐标范围和有效索引层范围。图层的坐标范围 (MBR) 对应 0 号索引层上的唯一网格的数据坐标范围, 存储图层的有效索引层范围

(MinLevel, MaxLevel) 是方便图层的空间查询, 当对图层空间查询时, 只需要遍历该图层的有效索引层, 而不用对 1 到 23 号全部索引层进行遍历。

[0068] 实施例 2: 对采用本发明方法所建立的统一的关系型和 Key-Value 型数据库空间数据进行查询的方法, 具体查询的步骤是:

[0069] (一) 根据本发明所述的基于关系型数据库的数据存储和空间索引建立方法, 对一个图层进行空间矩形范围 MBR (Xmin, Ymin, Xmax, Ymax) 查询的步骤是:

[0070] a) 从图层元数据表得到图层有效索引层号范围:(MinLevel, MaxLevel)。

[0071] b) 遍历(MinLevel, MaxLevel) 范围内的每个索引层 i, 计算 MBR 在 Level i 上的网格坐标范围 GBR (X1, Y1, X2, Y2), 然后将索引层号 i 存储在首字节中(左移 24 位), 网格坐标(X1, Y1, X2, Y2) 分别存储在余下的 3 个字节中, 就得到 GBR 的二维索引编码表示:(MinXI, MinYI, MaxXI, MaxYI), 其中: MinXI=i<<24|X1, MinYI=i<<24|Y1, MaxXI=i<<24|X2, MaxYI=i<<24|Y2。

[0072] c) 对于每个 GBR, 对图层索引表和数据表进行组合查询: 首先查询索引表中的要素 ID, 要求其对应的索引(XI, YI) 在(MinXI, MinYI, MaxXI, MaxYI) 范围内。然后用查询得到的要素 ID 到图层数据表中查询出要素的空间和属性数据。整个过程写成 SQL 语句如下:

[0073] SELECT DISTINCT ID, WKB, A1, A2, ...

[0074] FROM 图层数据表 d

[0075] INNER JOIN(

[0076] SELECT ID

[0077] FROM 图层索引表

[0078] WHERE (XI>=MinXI AND XI<MaxXI) AND (YI>=MinYI and YI<MaxYI)

[0079] )t ON t. ID=d. ID。

[0080] (二) 根据本发明所述的基于 Key-Value 型数据库的数据存储和空间索引建立方法, 对一个图层进行空间矩形范围 MBR (Xmin, Ymin, Xmax, Ymax) 查询的步骤是:

[0081] (1) 从图层元数据表得到图层有效索引层号范围:(MinLevel, MaxLevel)。

[0082] (2) 遍历(MinLevel, MaxLevel) 范围内的每个索引层 i, 计算 MBR 在 Level i 上的网格坐标范围 GBR (X1, Y1, X2, Y2), 对于 GBR 中的每个网格坐标(XI, YI),  $X1 \leq XI < X2$ ,  $Y1 \leq YI < Y2$ , 可以得到其对应的 Rowkey。

[0083] (3) 根据(2)生成的 Rowkey, 在图层索引表中查询 SID 列, 得到要素 ID 集合, 由要素 ID 集合到图层数据表中查询得到集合中每个要素的实际数据。

[0084] Key-Value 型数据库查询的关键是利用“按 Rowkey 检索的效率是最高的”这个特性, 通过把二维空间范围查询问题转化为一维 Rowkey 查询, 达到快速空间范围检索的目的。

[0085] 根据本发明所述的空间数据库存储表结构和索引方法, 当空间数据在关系型数据库和 Key-Value 型数据库之间迁移时, 首先将图层要素按对应数据库的图层数据表存储, 然后对把图层索引表中的索引编码进行二维编码和一维行键编码之间的变换, 因此不需要重新计算要素与索引网格的关系, 即不需要重建空间索引, 为空间数据迁移带来极大的方便。



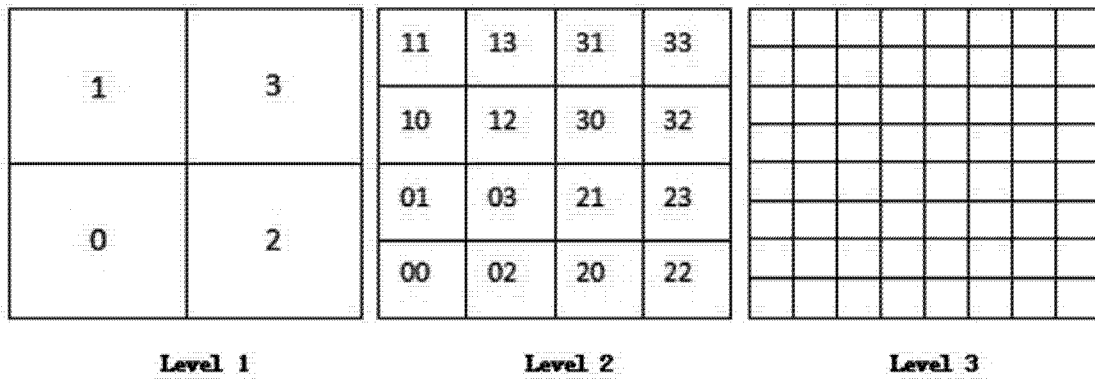


图 3

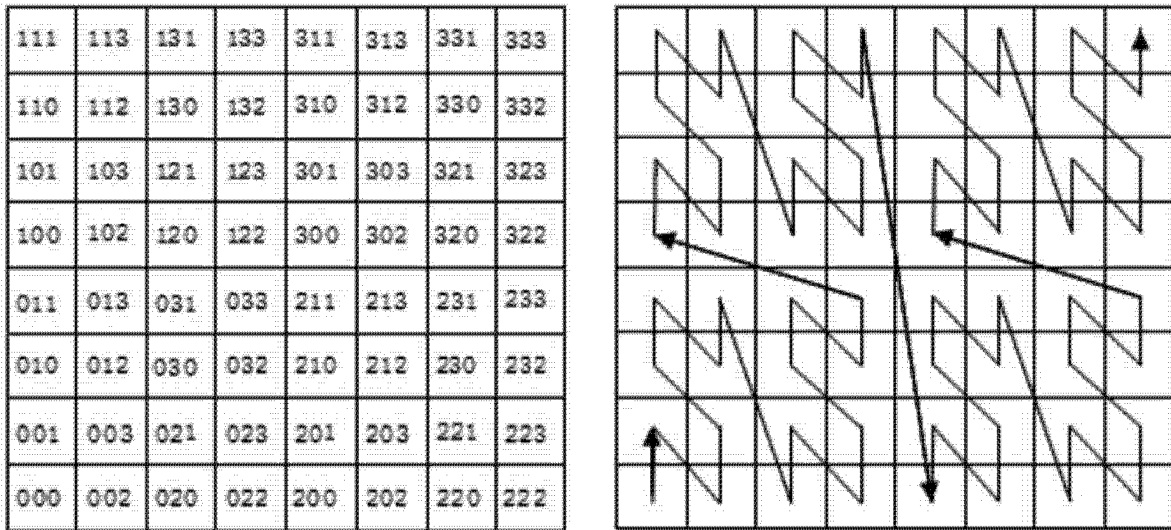


图 4

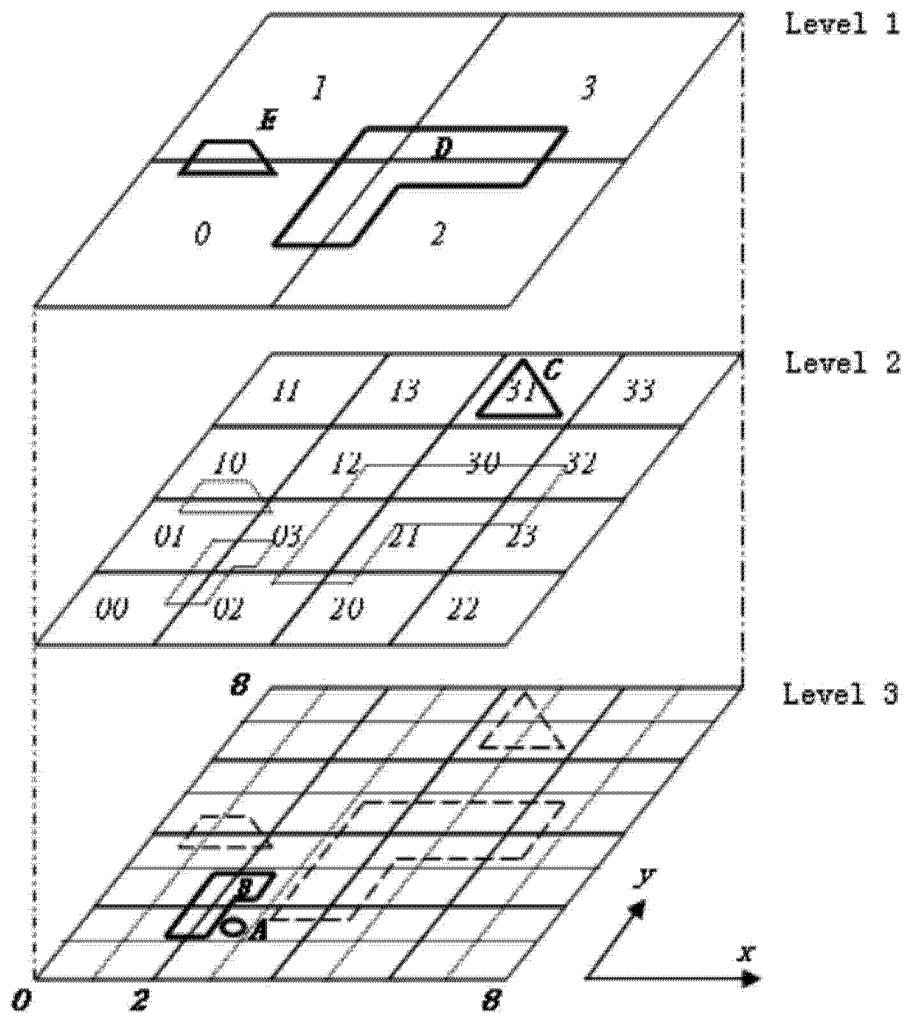


图 5