DOI:10.13203/j.whugis20130110　　文章编号:1671-8860(2014)10-1248-04

一种双映射变换的空间索引及空间连接算法研究

付仲良[1]　刘思远[1]　俞志强[2]

1 武汉大学遥感信息工程学院,湖北 武汉,430079
2 浙江省地理信息中心,浙江 杭州,310012

摘　要:空间索引会极大地影响空间连接操作的效率。提出了一种基于双映射变换的分布式空间索引,通过结合平面角变换和空间填充曲线的优点,对二维空间进行两次维度变换,使空间数据分片建立在一维的顺序存储队列基础上。在此基础上提出了一种空间拓扑连接算法,并进行了算法的四叉树优化和处理效率实验,对比了本文存储方法和传统 R-tree 存储在时效性和冗余度方面的效率。实验结果表明,本文方法能支持高效的空间连接。

关键词:双映射变换;空间索引;空间连接;四叉树优化
中图法分类号:P208　　**文献标志码**:A

海量空间数据管理与查询的主要问题在于处理与存储资源的高效利用[1],而分布式空间数据库技术的发展为海量空间数据的管理提供了高效的存储平台。空间连接[2]是查询多个数据集的空间对象间的关系,如叠加、包含、相交等。空间连接操作主要分为筛选和精化两个过程。其中筛选过程是通过最小外接矩形(minimum bounding rectangle,MBR)近似表达空间对象,简化大数据量的查询操作,并获取候选结果集。空间连接操作由于需要反复地读取空间对象而占用大量的 CPU,因而空间索引结构会极大地影响设备的利用效率。

现有的针对空间连接的索引改进方法大多是基于 R-tree 和空间填充曲线的改进。Huang 等[3]通过广度遍历优化提高 R-tree 的查询效率;Dai 等[4]通过角变换改进 R*-tree 提高数据访问效率;Lee 等[5]采用空间填充曲线降低维度的方法解决了空间数据随机分布的问题。但这些索引方法都是针对多空间数据库集成[6]模式设计的,即存在跨边界问题[7]。

本文结合空间填充曲线和维度变换方法,针对"精化"过程提出了一种基于 MBR 信息独立存储的分布式双映射变换索引,将二维空间通过平面角变换[4]映射到高维空间,再通过空间填充曲线映射到一维空间,对空间数据集进行一维顺序存储,有效地提高了分布式环境下空间连接操作[8]的处理效率。

1　双映射变换索引

维度变换是指将数据从原始的空间映射到不同维度的空间。Bohm 按维度变换将空间索引分为三类[9]:非变换索引、高维映射索引和低维映射索引。非变换索引有 R-trees、R*-trees、R+-trees、Cell-trees 等[10];高维变换索引有 Grid 文件索引[11]、KDB-trees[12]、LSD-trees[13] 等;低维映射索引有 Peano 曲线[14]和 Hilbert 曲线[10]等。双映射变换(double transform index, DT-index)是一种结合高维映射和低维映射的空间数据索引方法。第一次变换即角变换[4],是一种基于 MBR 近似表达的高维映射方法,可以有效地将 N 维空间的 MBR 信息映射到 $N\times 2$ 维空间表达,同时提升运算效率。第二次映射变换是一个低维映射过程,使用 Peano 曲线穿过所有的空间对象将 $N\times 2$ 维空间的上三角点阵按照 Peano 曲线的穿越顺序进行排序,并映射到一个一维空间中,称该一维空间为最终空间。最终空间中的每个点表示一个空间对象的 MBR,从而有效地避免了两个维度上

收稿日期:2013-03-25
项目来源:国家科技支撑计划资助项目(2011BAK07B02).
第一作者:付仲良,教授,博士生导师。主要从事地理信息系统、空间数据库及空间分析方法研究。E-mail:fuzhl@263.net

区域分片的问题。

2　空间连接算法

2.1　范围查询转换

空间拓扑连接查询是查询空间对象的拓扑关系的过程。由于空间对象采取 MBR 近似表达[2],因而任何一组空间对象间的拓扑关系都是通过比较 MBR 在 X、Y 方向的最大值和最小值进行的。在双映射变换索引下,可以将每一组空间对象间的拓扑关系连接转换为范围查询来执行。

假设给定的欧氏空间中存在两个数据集 R 和 S,$R=\{a, b, c\}$,$S=\{q\}$,对 R 和 S 进行空间相交连接操作。可将二维空间中的对象 q 视作查询范围,在对象 a、b、c 中查询与 q 有拓扑相交关系的对象,则需要同时满足以下几个条件:

$$Y_e \geqslant QY_s \wedge Y_s \leqslant QY_e$$

$$X_e \geqslant QX_s \wedge X_s \leqslant QX_e$$

其中,X_s、X_e、Y_s、Y_e 分别表示空间对象在 X、Y 方向上的最小值和最大值。当进行第一次映射变换时,二维空间中的对象被映射到 2×2 维空间的上三角矩阵中,查询范围表现为多个相同形态的区域。而通过空间填充曲线映射到最终空间后,该区域表现为一维空间中的多个片段,而查询对象则表现为一维空间中的点,这样,判断点是否在某个区间的执行效率显然高于计算两个对象 MBR 的空间关系处理。

根据此方法将空间连接转换为范围查询处理,对于空间数据集 $R=\{R_1,R_2,\cdots,R_n\}$ 和 $S=\{S_1,S_2,\cdots,S_n\}$ 的空间拓扑连接算法(以空间包含连接为例)的步骤可以描述为:

1)建立一个候选结果集 RS,初始化 $RS=$ Null;

2)对于每一个存储节点的点阵队列,顺序读取 R 和 S 中的每一个空间对象的 MBR;

3)对 S 中的每一个对象 S_i,分别读取 S_i 的 MBR,并将其作为查询范围。对于该范围内的每一个片段 Segment 进行顺序遍历,如果遍历的片段对应于数据集 R 中的空间对象 R_i,则将 S_i 写入到候选结果集 RS。

此时获取的候选结果集即为"筛选"阶段的结果,再对 RS 中的每一个空间对象进行精确的图形计算,即可获得空间连接的最终结果集。空间连接需要对两个数据集中的每组对象进行拓扑关系计算,因而对硬件设备的要求较高。通过双映射变换方法,一个空间数据集 R 转换为了查询范围的多个片段,每个片段的范围查询过程可以在多个计算节点同步并发执行,这种并行处理机制可以有效地提高计算效率。

2.2　空间填充曲线的四叉树优化

二维空间的双映射变换索引可以有效地提高范围查询的效率,但也存在复杂度较高的问题。假设范围 q 处在过度空间$(0, n-1, 0, n-1)$的点阵范围内,那么当使用空间填充曲线对三角点阵填充时,范围 q 在三角点阵的左上区域几乎占据了大部分的点,而在右上区域和左下区域则覆盖范围较小。在这种情况下,位于左上方区域的范围查询复杂度为 n^2(n 代表点阵的宽度),而处于左上和右下点阵的范围查询的复杂度的最大值可以达到$((n^2+n)/2)^2$。也就是说,对于整个点阵存储的双映射变换,其复杂度 N 在$(n^2, ((n^2+n)/2)^2)$范围内,即最大复杂度为 $O(n^4)$。从复杂度的最大值可以看出,范围 q 在左上区域与右上、左下区域的分布是影响计算效率的主要因素。

四叉树遍历方法可以有效地解决这个问题。过度空间的点阵符合四叉树分割处理的基本要求,即对 q 覆盖范围较大的区域进行整个区域的范围查询处理,而对如左下和右上这样的区域则进行四叉树分割处理。处理步骤如下。

1) 对于每个分布式存储的数据片段,建立一个候选结果集 RS;

2) 如果该片段处理过度空间的左上区域,则直接进行范围查询处理,判断对象是否处于范围 q 内,并将结果写入到 RS 中;

3) 如果该片段未处于过度空间的左上区域,则对该片段进行四叉树分割处理;对四叉树分割后的三个子片段进行遍历;

4) 判断遍历是否结束,若是,则进入步骤 5);否则返回步骤 2)进行处理,直到片段中无空间对象为止;

5) 合并所有的 RS,输出处理结果。

在该优化算法过度空间中,四分后的点阵片段在$(n, 2n-1)$范围内,因而其复杂度在$(n^2, 4n^2-4n+1)$范围内,即最大复杂度为 $O(n^2)$。

3　实验与讨论

本文实验数据主要采用城市基础数据中提取的数据集,总共分为 4 类:高程点、居民地面、水系线、道路线。实验数据集的基本信息如表 1 所示。

表1 实验数据

Tab. 1 Experimental Data

数据集	类别	大小/MB	MBR数目	缩写
高程点	点	567	739 744	TER
居民地	面	123	470 720	RES
水系	线	234	68 026	HYD
道路	线	123	147 518	TRA

将这几种数据按照一个矩形范围进行提取，分别提取2、4、…、1 024 kB大小的10种不同的数据块，并将10种数据块转换为符合索引近似表达的MBR。实验环境为16台在局域网环境下的i7处理器PC机，时效性实验采用独立的PC机处理对比，并行处理实验则采用1台PC机与多个节点的加速比统计对比。

3.1 时效性分析

实验对4类空间数据集的10种测试数据块构建了R-tree和DT-index索引。通过对两种不同的索引方法存储的相同数据执行空间拓扑连接查询操作，对比两者在运行时间和数据容量的关系。

从图1可以看出，DT-index的执行效率总体上要明显优于R-tree索引的执行效率。高程点类与居民地类要素因其重叠要素较少，在DT-index索引算法上的执行时间要比R-tree索引方法的低。

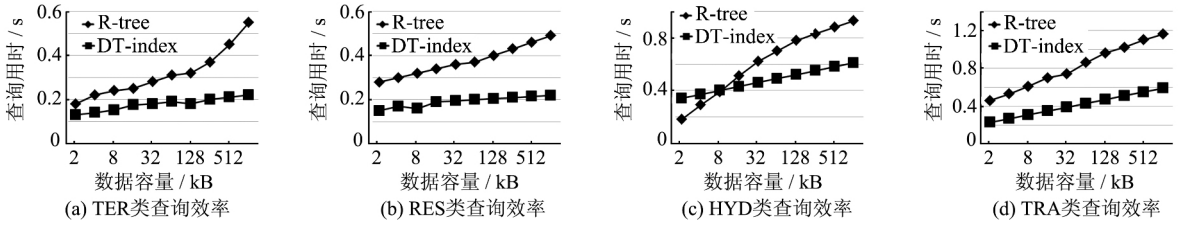

图1 空间查询的时效性对比

Fig. 1 Efficiency Comparision Between R-tree and DT-index

3.2 存储冗余度分析

由于空间数据的分片存储存在冗余，较低的索引存储冗余度有利于降低网络传输开销。对象重复率和结果重复率[15]可以有效地反映空间索引的存储冗余度，当数据量变化时，重复率越稳定，则冗余度越低。本文提取交通要素类作为测试数据，分别计算了数据集在不同容量上空间索引的对象重复率和结果重复率。

图2表示的是这两种冗余度验证参数的统计图。从图2中可以看出，DT-index在对象重复率和结果重复率方面都优于传统的R-tree存储方式，其对象重复率稳定在0.3～0.4之间，结果重复率则稳定在0.1～0.3之间，并随数据量增大的影响较缓上升。而R-tree由于区域分片的影响，数据子集的分割具有随机性，从而使得对象重复率和结果重复率具有较大的波动性。

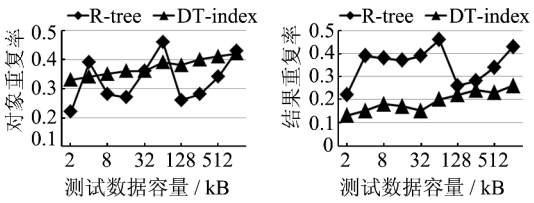

图2 分布式存储的冗余度对比

Fig. 2 Redundancy Comparision Between R-tree and DT-index

4 结 语

本文提出了一种双映射变换索引方法，分别通过高维和低维变换使原始空间映射到一维空间，空间对象形成顺序存储，有效解决了因区域分片造成的海量数据空间连接操作效率降低的问题。实验表明，本文提出的基于双映射变换索引的空间拓扑连接算法在分布式环境下具有较高的执行效率，同时还存在进一步研究的空间。首先，空间连接查询和分析操作可以进一步扩展到其他的查询类别，如空间相关性查询以及空间统计分析；其次，本文探讨的双映射变换及实验主要是基于二维空间，而双映射变换索引在三维空间和动态空间数据也存在可用性。未来的研究将着重基于双映射变换索引的空间分析方法及其应用。

参 考 文 献

[1] Goodchild M F. Geographic Information Systems and Science: Today and Tomorrow [J]. *Annals of GIS*, 2009, 15(1): 3-9

[2] Jacox E H, Samet H. Spatial Join Techniques [J]. *ACM Transactions on Database Systems* (TODS), 2007, 32(1): 7

[3] Huang Y W, Jing N, Rundensteiner E A. Spatial Joins Using R-trees: Breadth-first Traversal with

Global Optimizations[C]. International Conference on Very Large Data Bases, Athens, Greece, 1997

[4] Dai H, Whang K Y, Su H. Locality of Corner Transformation for Multi-dimensional Spatial Access Methods [J]. *Electronic Notes in Theoretical Computer Science*, 2008, 212:133-148

[5] Lee M J, Whang K Y, Han W S, et al. Transform-space View: Performing Spatial Join in the Transform Space Using Original-Space Indexes [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2006, 18(2): 245-60

[6] Wu Lun, Zhang Yi. The Integrated Framework on Distributed Multispatial Database System[J]. *Geography and Territorial Research*, 2002, 18(1): 6-10(**邬伦, 张毅. 分布式多空间数据库系统的集成技术**[J]. **地理学与国土研究**, 2002, 18(1): 6-10)

[7] Chen Di, Zhu Xinyan, Zhou Chunhui. Distributed Spatial Query Processing and Parallel Schedule Based on Zonal Fragmentation[J]. *Geomatics and Information Science of Wuhan University*, 2012, 37(8): 892-896(**陈迪, 朱欣焰, 周春辉. 区域分片下的分布式空间查询处理与并行调度方法**[J]. **武汉大学学报·信息科学版**, 2012, 37(8): 892-896)

[8] Chen Zhanlong, Wu Xincai, Xie Zhong. Study of Distributed Index Mechanism of Geospatial Data [J]. *Microelectronics*, 2007, 24(10): 54-57(**陈占龙, 吴信才, 谢忠. 分布式空间数据索引机制研究**[J]. **微电子学与计算机**, 2007, 24(10): 54-57)

[9] Bohm C, Berchtold S, Keim D. Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases [J]. *ACM Computing Surveys*, 2001, 33(3): 322-373

[10] Guo Wei, Guo Jing, Hu Zhiyong. Spatial Database Index [M]. Shanghai: Shanghai Jiaotong University Press, 2006 (**郭薇, 郭菁, 胡志勇. 空间数据库索引技术** [M]. **上海:上海交通大学出版社**, 2006)

[11] Orlandic R, Yu B. Implementing KDB-trees to Support High-Dimensional Data[C]. 2001 International Database Engineering & Applications Symposium, Grenoble, France, 2001

[12] Tao Y, Papadias D. The mv3r-tree: A Spatiotemporal Access Method for Timestamp and Interval Queries [C]. The 27th International Conference on Very Large Data Bases, Roma, Italy, 2001

[13] Lee K C K, Zheng B, Li H, et al. Approaching the Skyline in Z Order[C]. The 33rd International Conference on Very Large Data Bases, University of Vienna, Austrain, 2007

[14] Meng L, Huang C, Zhao C, et al. An Improved Hilbert Curve for Parallel Spatial Data Partitioning [J]. *Geo-spatial Information Science*, 2007, 10(4): 282-286

[15] Zhou Chunhui. A Study of Multi-Database Integration Model and Methods[D]. Wuhan: Wuhan University, 2010(**周春辉. 多空间数据库系统模式集成与空间查询关键技术研究**[D]. **武汉:武汉大学**, 2010)

A Novel Spatial Index with a High-performance Spatial Join

FU Zhongliang[1]　*LIU Siyuan*[1]　*YU Zhiqiang*[2]

1　School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China

2　Geographic Information Center of Zhejiang Province, Hangzhou 310012, China

Abstract: Spatial indexing seriously impacts the efficiency of spatial processing. In this paper, a new spatial index based on double transformation (DT-index) is proposed. As a dimensional transformation the DT-index benefits from both high and low dimensional mapping. The spatial objects are partitioned in sequential queue;more efficient than area partitioning. A spatial join algorithm based on the DT-index is introduced and optimized with a quad-tree. The experimental results reveal that the proposed method improves the performance of spatial join processing in terms of redundancy and speedup ratio through a comparison with the widely-used R-tree method.

Key words: DT-index; spatial index; spatial join; quad-tree optimization

First author: FU Zhongliang, professor, PhD supervisor, specializes in GIS and distributed spatial database. E-mail:fuzhl@263.net

Foundation support: The National Science and Technology Support Project, No. 2011BAK07B02.