

基于 Oracle Spatial 的空间数据库的索引与查询优化^①

钟伟清¹, 武 伟¹, 杜国伟²

¹(西南大学 计算机科学与信息学院, 重庆 400700)

²(西南大学 资源环境学院, 重庆 400700)

摘 要: 探讨了 oracle 10g spatial 空间索引的创建及如何利用空间索引优化空间分析. 了解空间查询主要是通过相关的空间算子和空间函数来进行的. 利用空间操作符和空间索引的结合, 实现空间查询的优化.

关键词: 空间索引; 空间查询; Oracle Spatial; 优化

Index and Query Optimization of Spatial Database Engine Based on Oracle

ZHONG Wei-Qing¹, WU Wei¹, DU Guo-Wei²

¹(School of Computer and Information Science, Southwest University, Chongqing 400700, China)

²(College of Resources and Environments, Southwest University, Chongqing 400700, China)

Abstract: This paper explores the creation of Oracle 10g spatial's spatial index and how to use spatial index to optimize spatial analysis. The spatial query is mainly carried out by relevant spatial operator and spatial function. Using spatial operator and spatial index can realize spatial query optimization.

Key words: spatial index; spatial query; oracle spatial; optimize

空间数据是以坐标和拓扑关系的形式存储的用于描述有关空间实体的位置、形状和相互关系的数据, 在地理信息系统中有着不可替代的位置^[1]. 随着地理信息系统技术的高速发展, 对空间数据的存储和查询这一技术也越来越受关注, Oracle Spatial、ArcSDE、Spatial Ware 等空间数据库引擎也相应地日益流行, 尤其是作为数据库领域巨头的甲骨文提供的空间数据管理模块 Oracle Spatial 更是倍受专业人士青睐^[2]. Oracle Spatial 是甲骨文公司针对空间数据管理的一组插件, 其对存储在基于 Oracle Spatial 的空间数据库中的空间数据提供了一种 SQL 模式和便于存储、检索、更新、查询的函数集. 它将空间数据和属性数据结合起来管理, 但是也带来了相应的问题, 最突出的就是空间数据的数据量往往是海量的. 因此, 为了能更加快速、准确地查询空间对象及其属性, 必须要对空间数据建立一种高效的索引和选择高效的空间操作符.

1 查询模型

Oracle Spatial 使用双层查询模型^[3,4]来解决空间查询问题, 即初级过滤操作和二级过滤操作. 经过两次过滤, 将返回精确的查询结果集, 其查询流程如下.

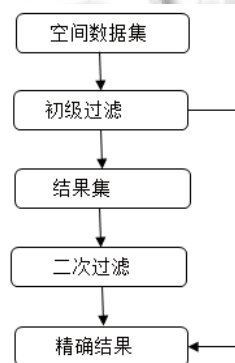


图 1 空间查询图

在初级操作过滤步骤中, 近似匹配满足条件的一组候选对象, 这些对象有可能满足给定的空间查询要

① 基金项目: 重庆市自然科学基金(CSTC, 2010BB1008)

收稿时间: 2013-06-07; 收到修改稿时间: 2013-07-26

求, 其结果集是精确查询的父集. 选择近似表示的条件为: 如果对象 A 与对象 B 的近似满足一种关系, 那么对象 A 与对象 B 就可能具有那种空间关系. 例如, 如果近似表示是分离的, 那么对象 A 和对象 B 就将是分离的, 但是如果近似表示非分离的, 对象 A 和对象 B 仍可能是分离的. 然后通过二次过滤, 对初次过滤结果再次求精, 就得到实体间的精确空间关系.

使用这样的二次过滤策略有几项优点: a. 空间对象一般都很大, 因此要占用大量主内存. 空间对象的近似表示在载入内存时占用的时间和空间要少的多. b. 对空间对象的计算一般都很复杂, 计算花费很大. 对象越复杂, 计算空间关系所需要的出来就越多. 使用近似对象的计算一般会很快, 需要的计算周期也要短的多.

2 空间查询

Oracle 10g Spatial 中对空间数据的查询主要是通过相关的空间算子以及空间函数来进行的. 下面我们来分析常见的一个空间函数和一个空间算子.

2.1 空间函数 SDO_GEOM.RELATE

SDO_GEOM.RELATE 主要是用于检查两个几何对象, 以确定它们之间的关系. 此函数不使用空间索引, 只是利用参数列传递过来的相应参数进行判定, 可以满足特定拓扑关系的查询.

语法格式: SDO_GEOM.RELATE(geometry1 in MDSYS.SDO_GEOMETRY, mask in VARCHAR2, geometry2 in MDSYS.SDO_GEOMETRY, tolerance in NUMBER) RETURN VARCHAR2

功能描述: 不使用空间索引, 来检查两个几何对象之间的空间关系.

参数描述:

geometry1: 空间数据对应的几何对象.

geometry2: 空间数据对应的几何对象.

Tolerance: 允许的精度范围

MASK 可以有以下参数:

anyinteract: geometry2 落在 geometry1 面上(包括在边上).

contains: geometry2 完全包含在 geometry1 几何对象中, 并且两个几何对象的边没有交叉.

coveredby: geometry1 完全包含在 geometry2 中, 并且这两个几何对象的边有一个或多个点相互重叠.

covers: geometry2 完全包含在 geometry1 中, 并且这两个几何对象的边有一个或多个点相互重叠.

disjoint: 两个几何没有重叠交叉点, 也没有共同的边.

equal: 两个几何是相等的.

inside: geometry1 完全包含在 geometry2 几何对象中, 并且两个几何对象的边没有交叉.

on: geometry1 的边和内部的线完全在 geometry2 上.

overlapbdisjoint: 两个几何对象相交, 但是边没有交叉.

overlapbintersect: 两个几何对象相交, 并且边有部分交叉.

touch: 两个几何对象有共同的边, 但没有交叉.

函数返回值: SDO_GEOM.RELATE(geometry1 in MDSYS.SDO_GEOMETRY, mask in VARCHAR2, geometry2 in MDSYS.SDO_GEOMETRY, tolerance in NUMBER)='some_mask_value', 其中 some_mask_value 的值是上述 MASK 参数中的值.

SDO_GEOM.RELATE 空间函数可以对少量数据直接进行判定求出精确结果, 但是数据量大时效率低下, 且可以在 where 子句及 select 语句中使用.

2.2 空间算子 SDO_RELATE

SDO_RELATE 用来判断哪些几何对象与给定的几何对象具有特定的空间关系, 主要用于二次过滤, 与 SDO_FILTER 空间算子不同, 获得的是精确结果, 这个操作相当于同时执行了第一步过滤(SDO_FILTER 的功能)和第二步过滤操作.

语法格式: SDO_RELATE(GEOMETRY1, GEOMETRY2, PARAMS)

功能描述: 通过空间索引, 根据给定的几何要素(如一个点)检索出与其有特殊空间关系的几何要素. 这个空间关系包括十一种: touch、overlapbdisjoint、overlapbintersect、equal、inside、coveredby、contains、covers、anyinteract、on、disjoint.

参数和关键字描述: GEOMETRY1 是指需要查询表中的几何列, 该列务必已经建立空间索引, 且该列的数据类型为 MDSYS.SDO_GEOMETRY. GEOMETRY2 可以是表中的几何对象, 也可以是具体的几何对象(包括关联的几何对象变量、通过 SDO_GEOMETRY 构造函数的几何对象), 该列的数据类型也为 MDSYS.SDO_GEOMETRY. PARAMS 决定了操作算子

的行为, 它的数据类型为 VARCHAR2, 它具有以下几种参数: MASK 指定了空间关系, 这是个必须的参数, 有如上所述的十一种关系, 多种空间关系可以组合; QUERYTYPE 指查询类型, 主要有 WINDOW 和 JOIN 两种。

空间算子的返回值: 表达式 SDO_RELATE(GEOMETRY1, GEOMETRY2, 'MASK= SOME_MASK_VALUE, QUERYTYPE = SOME_QUERYTYPE_VALUE')= 'TRUE', 其中的 TRUE 是指两个几何对象满足 SOME_MASK_VALUE 条件, 否则为 FALSE。

SDO_RELATE 空间算子可以对海量数据进行查询分析, 效率高但精度低, 且利用空间索引进行判定, 数据必须包含空间索引. 空间算子 SDO_RELATE 和空间函数 SDO_GEOM.RELATE 的功能是相对应的, 所以我们在使用时, 应该优先考虑使用空间算子 SDO_RELATE。

3 空间查询优化

3.1 实验数据

本文以奉节县和云阳县的行政区划(村组级)为例, 探讨空间数据索引和查询优化. 在 ARCMAP10.0 中, 利用 FeatureToPoint 方法将行政区划面数据转换为点数据, 共获得大约 26 万条记录, 实验区及样点分布图如图 2 所示。

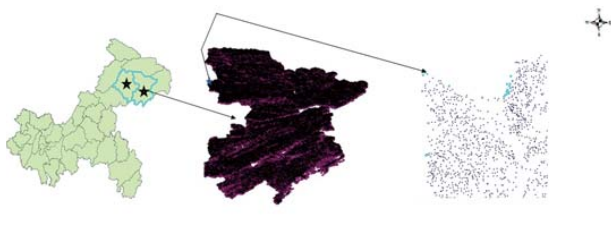


图2 实验区及样点分布图

3.2 建立指明对象类型的空间索引

在对数据表的几何列建立空间索引时, 可以使用 parameters(layer_gtype='spa_obj_type') 来指明空间索引中的对象类型, 其中 spa_obj_type 必须是 Oracle 10g Spatial 支持的数据类型(如 point、polyline、polygon 等), 本文采用 point 类型^[5]。

下面以奉节县的采样点空间数据表为例, 说明指定空间索引中的对象类型的实现方法。

```
create index dlbtb_wp_ftp_idx_y
```

```
on dlbtb_wp_ftp(geom)
indextype is mdsys.spatial_index
parameters('layer_gtype=point')
```

在该步骤中, dlbtb_wp_ftp 为存放大约 26 万条记录的样点空间数据表, 其中空间列名为 geom, 空间索引名为 dlbtb_wp_ftp_idx_y, 缺省类型为 R-tree 索引, parameters('layer_gtype=point') 用于指定空间对象类型为点。

3.3 优化效率评价

为了说明空间函数 SDO_GEOM.RELATE、空间算子 SDO_RELATE(使用未指明对象类型的空间索引)和空间算子 SDO_RELATE(使用指明对象类型的空间索引)对空间分析效率的影响, 本文对奉节县和云阳县的采样点数据表中的空间数据进行空间分析实验. 对比实验使用的 SQL 语句分别为:

A、使用空间函数 SDO_GEOM.RELATE

```
SELECT count(a.orig_fid)
FROM dlbtb_wp_ftp a, dlbtb_wp_d_x b
WHERE b.shape_id=1 and
SDO_GEOM.RELATE(a.geom,'INSIDE',b.geom,
0.5)=INSIDE';
```

B、使用空间算子 SDO_RELATE(使用未指明对象类型的空间索引)

```
SELECT COUNT(a.orig_fid)
FROM dlbtb_wp_ftp a, dlbtb_wp_d_x b
WHERE b.shape_id=1 AND
SDO_RELATE(a.geom,b.geom,'mask=INSIDE
querytype=WINDOW')='TRUE'
```

C、使用空间算子 SDO_RELATE(使用指明对象类型的空间索引)

```
SELECT COUNT(a.orig_fid)
FROM dlbtb_wp_ftp a, dlbtb_wp_d_x b
WHERE b.shape_id=1 AND
SDO_RELATE(a.geom,b.geom,'mask=INSIDE
querytype=WINDOW')='TRUE'
```

其中 dlbtb_wp_ftp 为点(采样点)对象空间表、dlbtb_wp_d_x 为面(县)对象空间表, 空间分析的含义是: 查询落在 shape_id 号为 1 的多边形范围内的样点个数。

然后运用 pl/sql 工具统计了这三个查询语句所消耗的时间, 实验的对比结果如下表所示。

表 1 SDO_GEOM.RELATE、SDO_RELATE(使用未指明对象类型的空间索引)和 SDO_RELATE(使用指明对象类型的空间索引)的空间分析耗时对比表

操作类型	时间 (以秒为单位)			平均值
A	3262.059	2900.215	3965.124	3375.799
B	109.747	128.077	121.322	119.715
C	77.985	79.717	91.572	83.091
最小值	77.985	79.717	91.572	83.091

注: A: SDO_GEOM.RELATE; B: SDO_RELATE(使用未指明对象类型的空间索引); C: SDO_RELATE(使用指明对象类型的空间索引).

从表 1 可以看出, 使用函数的平均耗时是 3375.799 秒, 使用 SDO_RELATE(使用未指明对象类型的空间索引)的平均耗时是 119.715 秒, 而使用 SDO_RELATE(使用指明对象类型的空间索引)的平均耗时是 83.091 秒, 故可以得出以下结论: a. 完成相对应空间分析操作时, 使用空间算子的耗时比使用空间函数的耗时要少. 本例的空间分析耗时, 从使用函数的平均耗时 3375.799 秒降到使用算子(使用指明对象类型的空间索引)的平均耗时 83.091 秒, 节约了 3292.708 秒, 空间分析效率提高了 97.538%. b. 进行空间分析时, 使用指明对象类型的空间索引可以比使用未指明对象类型的空间索引的操作要耗时少. 本例的空间分析耗时从未指明的 119.715 秒降到指明后的 83.091 秒, 节约了 36.624 秒, 空间分析效率提高了 30.59%.

4 结语

随着地理信息系统的不断发展, 对空间数据库要

求也在不断的提高, 随之而来, Oracle Spatial 也面临新的挑战. 不过随着 Oracle 版本的不断更新, Spatial 的功能也越来越完善. 从注重海量空间数据的存储与管理到对更强大的空间查询分析支持双管齐下的过渡, 这将使得 Oracle Spatial 的功能更加强大, 从而受到越来越多人的青睐.

参考文献

- 1 龚健雅, 杜道生, 李清泉. 当代地理信息技术. 北京: 科学出版社, 2004. 11-41.
- 2 熊雷. 基于 Oracle Spatial 的空间查询技术及其优化. 软件导刊, 2008, 7(8): 7-9.
- 3 魏金占, 邓青. Oracle Spatial 的空间查询分析功能浅析. 信息技术与信息化, 2006, (2): 99-101.
- 4 李天琦, 韦春桃, 李全杰. 基于 Oracle Spatial 空间数据库的索引与查询. 桂林工学院学报, 2008, 28(1): 131-136.
- 5 姚顺彬. 基于 Oracle Spatial 的空间分析性能优化研究. 林业资源管理, 2007, (1): 87-90.
- 6 苏炳均, 诸昌铃, 李林. 基于 Oracle Spatial 的 GIS 数据组织及查询. 计算机应用研究, 2006, (12): 278-281.
- 7 肖飞, 唐媛. Oracle Spatial 空间索引解析. 信息技术与信息化, 2009, (1): 66-67.
- 8 Oracle Spatial User's Guide and Reference Release 9.2. 2002.4.
- 9 王亚文, 容晓峰, 李建元. 基于 Oracle Spatial 10g 的 GIS 数据存储研究. 陕西理工学院学报, 2010, 26(1): 48-53.

(上接第 170 页)

Applications, 2005, 41(1): 1464-1471.

- 5 Dev Shukla R, Tripathi RK. Maximum Power Extraction Schemes & Power Control in Wind Energy Conversion System. International Journal of Scientific & Engineering Research, 2012, 3(6): 1-7.
- 6 Kazmi R, Goto H, Guo HJ, Ichinokura O. A Novel Algorithm for Fast and Efficient Speed-Sensorless Maximum Power Point Tracking in Wind Energy Conversion Systems, IEEE Trans. on Industrial Electronics, 2011, 1(58): 29-36.
- 7 Haque ME, Negnevitsky M, Member S, Muttaqi KM. A Novel

Control Strategy for a Variable-Speed Wind Turbine With a Permanent-Magnet Synchronous Generator. IEEE Trans. on Industry Applications 2010(46): IEEE Industry Applications Society Annual Meeting, 2008: 1-8.

- 8 Galdi V, Piccolo A, Siano P. Designing an adaptive fuzzy controller for maximum wind energy extraction, IEEE Trans. on Energy Conversion, 2008, (23): 559-569.
- 9 李兴鹏, 石庆均, 江全元. 双模糊控制法在光伏并网发电系统 MPPT 中的应用. 电力自动化设备, 2012, 32(8): 113-117.