文章编号：1001－9081(2010)S2－0148－04

数据分区在地学空间数据查询中的应用

孙雷刚[1,2]，周可法[1]，张楠楠[1,2]，许文强[1]
(1.中国科学院 新疆生态与地理研究所，乌鲁木齐 830011；2.中国科学院 研究生院，北京 100049)
(sunlgyuyn@sina.com; zhoukf@ms.xjb.ac.cn)

摘 要：通过研究 Oracle Spatial对空间数据的存储管理机制，提出使用数据分区技术来优化地学空间数据的空间查询性能。在详细介绍数据分区原理基础之上，结合实际应用，以范围分区为例，分别对常规表和分区表进行了空间查询测试，并对实验结果进行了反复的对比分析。研究表明，数据分区在提高地学空间数据的空间查询速度方面效果显著。

关键词：Oracle Spatial；数据分区；空间数据；空间查询
中图分类号：TP311　　文献标志码：A

Application of data partition in query of geo-spatial data

SUN Lei-gang[1,2], ZHOU Ke-fa[1], ZHANG Nan-nan[1,2], XU Wen-qiang[1]
(1. Xinjiang Institute of Ecology and Geography, Chinese Academy of Sciences, Urumqi Xinjiang 830011, China;
2. Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: The paper proposed to use the technology of data partition to optimize the spatial query performance of geo-spatial data after studying storage and management mechanism of spatial data based on Oracle Spatial. On the foundation of expounding the principle of data partition, considering applications in practice, taking range partition for example, carrying out spatial query test respectively to conventional table and partition table, contrasting and analyzing the experiment results again and again. The study proves that data partition can improve spatial query speed of geo-spatial data.

Key words: Oracle Spatial; data partition; spatial data; spatial query

0 引言

随着空间数据库技术的发展以及地学工作的深入研究，特别是在地学资源数据库成为共享数据源的今天，地学空间信息的查询和分析倍受关注。地学数据量剧增，空间对象、空间查询高度复杂，应用需求增多而且日趋复杂化，这些都对应用系统的运行效率及空间信息的查询响应速度提出了更高的要求，空间查询优化成为了空间数据库应用的难点和研究热点。然而，空间数据查询与纯属性数据查询有很多相似之处，关系数据库及其查询优化的基本理论和方法在空间查询优化技术研究中仍然适用，而且较多的空间数据或时空数据的查询优化研究完全采用关系数据库的查询优化方法[1]。

数据分区技术大都应用于商业数据及海量地学属性数据等非空间查询领域。本文通过深入研究 Oracle Spatial对空间数据的存储管理机制，基于 Oracle Spatial对空间数据的存储格式公开且支持 SQL语句的特性，把数据分区引入到了地学空间数据的空间查询领域，并结合实际应用，通过反复的实验设计与对比分析，探讨了数据分区在优化海量地学空间数据的空间查询性能方面的适用性和有效性。

1 技术方法

1.1 Oracle Spatial对空间数据的存储管理机制

Oracle Spatial是 Oracle的空间数据库组件，提供了一种关系对象模型表示方法来表示地理信息：现实世界中的每一个空间对象作为一个对象存储在关系数据库表的一列中，相关属性存储在此表的其他列中，并且提供了一套 SQL模式和函数，便于存储、恢复、更新和查询存储在 Oracle数据中的空间要素集合。它利用关系型数据库来存储和处理空间数据，实现了空间数据和属性数据的无缝集成和一体化存储管理[2]。Oracle Spatial为存储空间数据提供了 SDO_GEOMETRY类型，该类型主要由 5个部分组成：SDO_GTYPE, SDO_SRID, SDO_POINT, SDO_ELEM_INFO和 SDO_ORDINATES，并用元数据表来管理空间数据，还提供了 R树索引、四叉树索引等索引机制来提高空间查询和空间分析的速度[3]。SDO_GEOMETRY对象类型能存储很多空间数据类型，包括点、线、面、多面、三维合成表面、简单实体（如立方体）、合成实体（如一栋带有多个房间的建筑物）等。Oracle Spatial不仅允许用户和应用软件开发者将他们的空间数据无缝地整合到企业级应用中去，而且允许供应商的工具和应用软件直接访问 Oracle数据库的空间数据，建立空间索引、进行空间数据分析等复杂的 GIS功能均可用 Oracle Spatial所提供的函数完成，这就极大地降低了 GIS系统开发的成本，因此越来越多的 GIS系统开发商开始寻求对 Oracle Spatial的支持[4]。

Oracle Spatial对空间数据的存储格式公开以及支持 SQL语句的特性，既方便用户利用数据库的操作方法来操作空间数据，又便于用户根据自己的需求进行扩展，本文就是基于

收稿日期：2010－04－21。　基金项目：新疆科技厅科技项目（200815116；200733145-4）；国家 973计划项目（2006BAB07B07；2007BAB25B06）；中国科学院知识创新项目（KZCX2-YW-107）。

作者简介：孙雷刚（1984－），男，河南周口人，硕士研究生，主要研究方向：空间数据库、GIS；周可法（1972－），男，河南南阳人，研究员，博士，主要研究方向：地图学、GIS；张楠楠（1980－），女，新疆乌鲁木齐人，助理研究员，博士，主要研究方向：遥感地质；许文强（1979－），男，甘肃高台人，副研究员，博士，主要研究方向：干旱区碳循环、GIS。

Oracle Spatial的该特性,通过研究数据分区原理及相关理论知识,把数据分区引入到了地学空间数据的空间查询优化领域。

1.2　数据分区的原理

数据分区技术是为了简化对数据库大表的管理而推出的一项重要技术,是构建千兆字节数据系统中增加性能、可管理性及可用性的关键工具。数据分区是按约定方式或约定逻辑划分库表结构,并将数据分散部署到多个相对较小的子分区中,各子分区还可以从物理存储上分隔开,即不同的子分区数据保存到不同或相同磁盘的不同表空间的数据文件中[5]。分区表中每个分区可以在逻辑上认为是一个独立的对象,可以在一个表中的一个或多个分区上进行诸如删除、移动等的维护操作,而不会影响到其他分区,具有分区独立性的特点,如果选择合适的数据分区策略,会大大加快数据的查询速度。

数据库对象的存储单位是页,页链将多个页连接成表,所以将一个表进行分区实际上就是为了一个表建立多个页链,分区可增加并行查询的并行度,可把对表的 I/O操作均匀地分布在多个磁盘上,从而提高查询速度[6]。分区能够将表、索引或按索引组织的表进一步细分为小块,每个分区都有自己的名称,还可以选择自己的存储特性。从数据库管理员的角度来看,分区后的对象具有多个小块,这些小块既可以集中管理,也可以单独管理,这就使得管理员在管理分区后的对象时具有相当大的灵活性,数据库管理能够用分而治之的手段进行数据管理。此外,分区可以大大降低数据的总拥有成本,它使用"分层存档"方法在低成本存储设备上使较旧的相关信息保持联机状态[7]。

1.3　数据分区的类型

数据表分区分为:范围分区、列表分区、散列分区和组合分区。

范围分区是指按照列值范围将表行的数据分布到不同分区段中,具体说是将表中记录按某一字段或若干个字段的取值范围进行分区,基于建立每个分区的分区键值的范围将数据映射到不同分区。

列表分区用于将离散数据有效地部署到不同分区中,是将表中记录按照某一字段的离散值的列表进行分区,列表分区由列表值的分区关键字指定每个分区,能够准确地控制数据行到分区的映射,用一种自然的方式组织无序和无关联的数据集。

散列分区又称为 Hash分区,是指按照 Oracle所提供的散列(Hash)函数计算列值数据,并最终按照函数结果分区大表数据,是将表中数据均匀地分布到若干个指定的分区,主要用来均衡 I/O控制数据分区均匀。

组合分区首先按照列值范围进行逻辑的范围分区,将表分成几个大的分区,然后在每个大分区上使用 Hash或列表方法进一步进行子分区。因此组合分区包括范围—Hash分区和范围—列表分区。

各类型分区的原理如图 1所示。

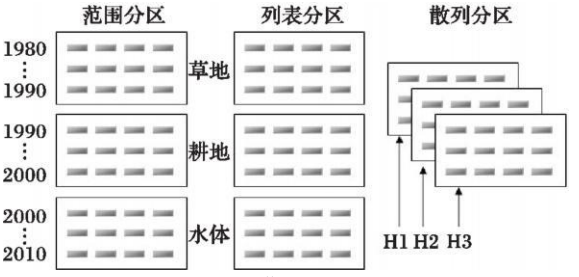

图 1　分区原理图

2　实验设计与分析

2.1　数据来源

本实验的数据为新疆 2000年的土地利用数据和行政区划图,其部分数据分别如表 1和表 2所示,由于空间数据列中的数据较大,故在此省略,新疆 2000年土地利用数据表中的类型名和类型边号分别是中国科学院土地利用/土地覆盖分类系统中 6大类名称和细分的子类编号。

表 1　新疆 2000年土地利用(部分)(tdlp)

地物编号	类型编号	类型名	…	空间数据
1	21	Wood	…	…
2	21	Wood	…	…
3	42	Water	…	…
4	52	Resident	…	…
5	11	Plough	…	…
6	31	Grass	…	…
7	65	Undevelopped	…	…
8	12	Plough	…	…
9	23	Wood	…	…
…	…	…	…	…

表 2　新疆行政区划图(部分)(region)

编号	区划名	拼音名	…	空间数据
1	和田地区	Hetian	…	…
2	阿克苏地区	Akesu	…	…
3	塔城地区	Tacheng	…	…
4	伊犁地区	Yili	…	…
…	…	…	…	…

2.2　数据分区方法的选择

在数据分区的各种类型中,散列是通过将 Hash算法用于分区关键字来确定指定行的分区,通过 Hash函数来控制数据行到分区的映射,使得数据行均匀地分布到各分区上,各分区大小趋于一致,这样对大型数据表的存储和对磁盘空间的利用均能带来益处;然而,空间数据中各点、线、面的大小及复杂程度都不一样,如果强行均匀分布,势必会将原来相邻或类型相同的数据行分到不同的分区,用户在查询时,反而会跨分区查询,这对空间查询效率提高并不大,对数据的维护效率也带来不便。

列表分区是按照某一字段的离散值的列表进行分区,基于每个分区的分区键值将数据分区映射到不同分区,而范围分区是按某一字段或若干个字段的取值范围进行分区,基于建立每个分区的分区键值的范围将数据映射到不同分区。从某种意义上说列表分区是可以转换为范围分区的,转换后对数据表的操作会变得很灵活。例如对分区表进行插入数据的操作,列表分区的表要求数据中的分区键值必须是指定的几种,而范围分区的表只要求待插入数据的分区键值在某个范围之内即可,从整个表来看范围分区几乎没有要求。

表 3　实例表 tdlp的表结构

列名	数据类型
地物编号	NUMBER
类型编码	NUMBER
类型名	VARCHAR2
长度	NUMBER
面积	NUMBER
空间数据	SDO_GEOMETRY

建立列表分区的代码如下:

```
CREATE TABLE TDLY
( ID NUMBER, CLASSID NUMBER, CLASSNAME VARCHAR2
  (30), LENGTH NUMBER, AREA NUMBER, GEOMETRY SDO_
  GEOMETRY)
PARTITION BY LIST (CLASSNAME)
  (PARTITION P1 VALUES('GRASS') TABLESPACE GRASS,
  PARTITION P2 VALUES('PLOUGH') TABLESPACE PLOUGH,
  PARTITION P3 VALUES('RESIDENT')
    TABLESPACE RESIDENT,
  PARTITION P4 VALUES('UNDEVELOPED')
    TABLESPACE UNDEVELOPED,
  PARTITION P5 VALUES('WATER') TABLESPACE WATER,
  PARTITION P6 VALUES('WOOD') TABLESPACE WOOD);
```

转换为范围分区的代码如下:

```
CREATE TABLE TDLY
( ID NUMBER, CLASSID NUMBER, CLASSNAME VARCHAR2
  (30), LENGTH NUMBER, AREA NUMBER, GEOMETRY SDO_
  GEOMETRY)
PARTITION BY RANGE (CLASSNAME)
  (PARTITION P1 VALUES LESS THAN
    ('GRASSZZZZZZ') TABLESPACE GRASS,
  PARTITION P2 VALUES LESS THAN
    ('PLOUGHZZZZZZ') TABLESPACE PLOUGH,
  PARTITION P3 VALUES LESS THAN
    ('RESIDENTZZZZZZ') TABLESPACE RESIDENT,
  PARTITION P4 VALUES LESS THAN
    ('UNDEVELOPEDZZZZZZ')
    TABLESPACE UNDEVELOPED,
  PARTITION P5 VALUES LESS THAN ('WATERZZZZZZ')
    TABLESPACE WATER,
  PARTITION P6 VALUES LESS THAN
    ('WOODZZZZZZ') TABLESPACE WOOD);
```

从上面的转换过程可以看出:列表分区到范围分区的转换是很容易实现的,只需要修改一下分区方法的参数以及对分区键的处理技巧。

另外,从数据表分区的原理上来讲,无论是哪一种分区都是将一张大的数据表分为若干个小的数据分区表,唯一不同的是分区的方式,数据表的物理隔离和数据的读取方式都是一样的。范围分区是应用最为广泛的一种分区类型,因而本文以范围分区为例,来详细阐述数据分区在地学空间数据查询中的应用。

2.3 基于范围分区的实验过程

1)创建表空间。

建立用于存放数据表各分区的用户表空间,以下是创建用户表空间的代码:

```
CREATE TABLESPACE P1
DATAFILE 'D:\ORADATA\DATA01.DBF' SIZE 1000M
//将表空间 P1的数据文件 P1存放在磁盘 D的 ORADATA目录下
```

在默认情况下,当建立表空间或者为表空间增加数据文件时,如果不指定 AUTOEXTEND选项,那么该数据文件将不允许自动扩展,也就是说用户将无法为表空间追加更多数据,否则会显示“ ORA-01653”错误消息,为便于以后数据的自动扩展,还需以下操作:

```
ALTER DATABASE DATAFILE 'D:\ORADATA\DATA01.DBF'
AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
```

使用以上相同的代码,可分别建立 DATA02、DATA03、DATA04、DATA05、DATA06等表空间,用于存储不同的表分区,这些分区既可以放在同一服务器上的不同磁盘上,也可以放在不同服务器上,这样更能提高数据的 I/O性能。

2)建立范围分区。

根据新疆 2000年的土地利用数据表,选择类型名(CLASSNAME)作为范围分区键,创建不同的数据表分区,分别存放在用户定义的表空间中,创建分区的代码如下:

```
CREATE TABLE TDLY
( ID NUMBER, CLASSID NUMBER, CLASSNAME VARCHAR2
  (30), LENGTH NUMBER, AREA NUMBER, GEOMETRY SDO_
  GEOMETRY)
PARTITION BY RANGE (CLASSNAME)
  (PARTITION P1 VALUES LESS THAN
    ('GRASSZZZZZZ') TABLESPACE DATA01,
  PARTITION P2 VALUES LESS THAN
    ('PLOUGHZZZZZZ') TABLESPACE DATA02,
  PARTITION P3 VALUES LESS THAN
    ('RESIDENTZZZZZZ') TABLESPACE DATA03,
  PARTITION P4 VALUES LESS THAN
    ('UNDEVELOPEDZZZZZZ') TABLESPACE DATA04,
  PARTITION P5 VALUES LESS THAN
    ('WATERZZZZZZ') TABLESPACE DATA05,
  PARTITION P6 VALUES LESS THAN
    ('WOODZZZZZZ') TABLESPACE DATA06);
```

其中的 RANGE指定分区类型为范围分区,P1为分区名,DATA01为数据表分区要存放的表空间。土地类型为草地的数据记录存放在 DATA01分区中;土地类型为耕地的数据记录存放在 DATA02分区中;土地类型为城乡居民用地的数据记录存放在 DATA03分区中;土地类型为未利用土地的数据记录存放在 DATA04分区中;土地类型为水域的数据记录存放在 DATA05分区中;土地类型为林地的数据记录存放在 DATA06分区中。

3)插入数据。

在数据表分区建立以后,将数据分别插入到普通表和分区表中,插入到分区表中的数据将根据分区键的范围自动插入到相应的表分区中。

4)查询。

为了便于对比分析实验结果,分别对普通表和范围分区表执行相同的空间查询,查询语句如下:

```
Select B.* from REGION A, TDLY B
Where A.NAME='区划名' AND B.CLASSNAME='类型名'
  AND SDO_RELATE(B.GEOMETRY, A.GEOMETRY,
  'MASK=INSIDE')='TRUE';
```

上述查询语句的意义为:查询出某一地区内的所有符合指定土地类型的所有空间数据。

如表 4给出了查询测试时使用的机器配置。

表 4 测试机的配置

配件名称	配置参数
CPU	Intel Pentium 4 2.8 GHz
主板	SiS-648 FX
内存	1.5 GB
硬盘	日立 7 200转,160 GB
显卡	GeForce 5 200 256 MB显存
显示器	DELL液晶

2.4 实验结果与分析

为了详细说明分区的效果,表 5给出了对常规表和范围分区表进行空间查询时,并且测试数据量和数据相同时各自所消耗的查询时间,表 6给出了当增加相同数据和相同数据量时的各自所消耗的查询时间。

从表 5及表 6中都可以看出:在查询相同数据量的空间数据时,查询经范围分区后的空间数据所消耗的时间远远少

于对常规表的查询;在随机的空间查询中,查询出的空间数据的数量和所消耗的查询时间没有特定的关系。这主要是由于空间数据的复杂性所引起的,比如说,查询出 5条复杂的空间数据所消耗的时间可能高于查询出 10条简单的空间数据所消耗的时间,空间数据越复杂,在进行空间运算时所消耗的时间就会越长。

表 5　常规表和范围分区表空间查询时间比较(随机)

测试数据数量	查询常规表耗时/s	查询范围分区表耗时/s
23	15.23	2.85
2	17.00	0.60
203	41.21	12.34
388	47.28	4.48
622	53.00	1.50
679	58.79	6.60
77	59.78	7.81
120	60.39	18.00
143	72.23	20.03
148	75.87	21.00
465	104.84	11.71
351	122.81	33.92
4279	167.12	25.43
1502	172.46	4.64
381	183.96	7.68
504	188.89	110.68
769	225.90	12.93
1003	234.62	7.73
5177	238.01	8.96
4044	245.31	95.50
1391	269.56	10.00
1468	323.82	17.85
1293	324.81	89.70
370	329.12	3.68
2305	500.00	113.06
1765	505.76	7.68
1379	687.00	415.06
3369	693.95	41.34
855	793.62	241.93

表 6　常规表和范围分区表空间查询时间比较(增量相同)

测试数据数量	查询常规表耗时/s	查询范围分区表耗时/s
23	15.23	2.85
120	60.39	18.00
143	72.23	20.03
148	75.87	21.00
…	…	…
855	793.62	241.93

为了更直观、鲜明地对比分析查询常规表和分区表所消耗的时间,根据实验结果分别绘制了如图 2和图 3。图 2是在查询相同数量的空间数据时,随着常规查询耗时的增减,优化后耗时的变化图;图 3是随着常规查询耗时的增加,它们之间时间差的变化趋势图。从图 2中可以看出:随着常规耗时的增减,优化后耗时表现出跳跃性的变化,这主要还是与之前所说的空间数据的复杂性有关;另外,无论优化后耗时呈现怎样的跳跃性变化,与优化前相比,其优化效果还是很明显的,查询时间消耗远低于优化前的时间消耗。图 3表明随着常规查询耗时的增加,它们之间的时间差会表现出上升趋势,这在一定程度上也说明了随着空间数据量的增加,其优化效果会越来越好。

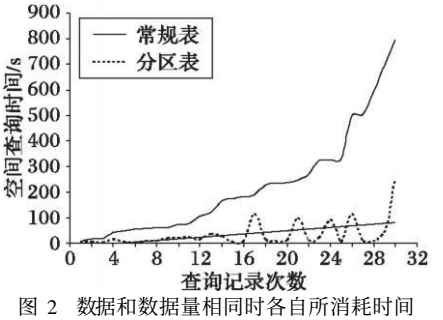

图 2　数据和数据量相同时各自所消耗时间

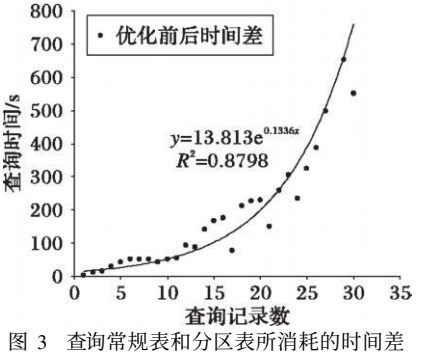

图 3　查询常规表和分区表所消耗的时间差

为进一步说明查询优化的效果,从表 6可以得出图 4。图 4是指当数据增量和增加的数据都相同时,也就是在排除空间数据复杂度的差异的情况下,分别对常规表和分区表执行相同空间查询所消耗的时间变化趋势。从两者变化的趋势及变化斜率可以看出,随着数据量的增大,优化的效果非常明显。

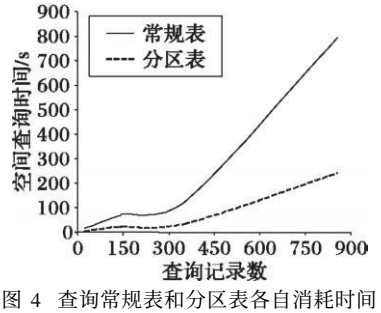

图 4　查询常规表和分区表各自消耗时间

3　结语

空间数据库的查询效率一直是衡量空间数据库性能的重要指标。然而,目前我国各类型、运行于各层次的地学空间数据库大都不同程度地面临着空间查询效率低的问题。本文针对该问题,把数据分区应用到了地学空间数据的空间查询领域,分区后的空间查询效率比分区前有显著的提高。

参考文献:

[1] 张波. 时空数据库查询优化研究[J]. 郑州航空工业管理学院学报:社会科学版, 2005, 24(5):156—157.

[2] 潘农菲. 基于 Oracle Spatial的 GIS空间数据处理及应用系统开发[J]. 计算机工程, 2002, 28(2):32—35.

[3] 彭明军, 李宗华. 基于 Oracle Spatial的空间数据互操作[J]. 计算机工程与应用, 2006, 42(32):154—157.

[4] 芮小平, 杨崇俊, 高积粮. 用 OCI4.0实现 Oracle Spatial接口[J]. 计算机应用, 2003, 29(12):48—50.

[5] 钟鸣, 石永平. Oracle性能优化技术内幕[M]. 北京:机械工业出版社, 2002:45—88.

[6] 陈苗, 杨毅恒, 王永志. 表分区在优化海量地质数据检索中的应用[J]. 世界地质, 2008, 27(1):100—104.

[7] BEER H. Partition In Oracle Database 11g[EB/OL]. (2007—06)[2010—01—08]. http://www.oracle.com/technetwork/database/enterprise-edition/partitioning-11g-whitepaper-159443.pdf.