

基于时空分区聚簇的海量时空数据性能 优化方法研究

谢炯 刘仁义 刘南

(浙江大学 GIS 重点实验室, 杭州 310028)

摘要 时空一体化的海量数据管理及相应的时序分析能力是新一代 GIS 软件体系的重要研究目标之一。当前, 基于无缝海量大表的空间及时态空间数据的存取效率亟待提高。为了对海量时空数据进行有效管理和提高时空检索效率, 以扩充关系型时空模型为基础, 对大型对象-关系型数据库平台所提供的数据分区与聚簇方法进行了时空维的扩展, 提出了基于时空分区聚簇(spatio-temporal partition clustering, STPC)的海量时空数据性能优化方法。基于 2GB ~ 60GB 的单表所进行的检索效率对比测试结果表明, STPC 机制较普通的数据组织方式时空检索效率平均提高了 10.1%。

关键词 海量时空数据 时空分区 时空聚簇 时空分区聚簇

中图法分类号: P208 文献标识码: A 文章编号: 1006-8961(2006)09-1334-08

A Study of Performance Optimization Method for Massive Spatio-temporal Data Based on Spatio-temporal Partition Clustering

XIE Jiong, LIU Ren-yi, LIU Nan

(Keylab of GIS, Zhejiang University, Hangzhou 310028)

Abstract Integrative management of massive spatio-temporal data and the relevant capabilities for time series analysis are important research targets of new generation GIS software architecture. At present, access efficiency of spatial and spatio-temporal data based on seamless massive table mechanism needs solid improvement. For managing massive spatio-temporal data effectively and improving spatio-temporal search performance, partitioning and clustering method of large object-relational database platform are extended to spatio-temporal dimension, and performance optimization method for massive spatio-temporal data based on spatio-temporal partition clustering is presented, which is based on extended relational spatio-temporal data model. The search efficiency test on single table of 2GB to 60GB shows that STPC provides better spatio-temporal search performance(about 10.1%) than the normal data organization methods.

Keywords massive spatio-temporal data, spatio-temporal partition, spatio-temporal clustering, spatio-temporal partition clustering

1 引言

时空一体化的海量数据管理及相应的时序分析

能力是新一代 GIS 软件体系的重要研究目标之一, 也迫切需要 GIS 学术界优先对海量时空数据的合理组织与高效存储技术加强研究^[1]。目前, 基于 RDBMS(relational database management system)大表

基金项目: 国家自然科学基金项目(40271087); 浙江省自然科学基金项目(401006); 国家高科技发展计划“ 863 ”项目(2001AA135180、2001AA630301)

收稿日期: 2005-04-27 改回日期: 2005-10-14

第一作者简介: 谢炯(1977 ~)男, 2005 年获浙江大学博士学位, 现在浙江大学从事博士后工作。主要研究领域为时空数据模型、海量时空数据管理及 GIS 应用系统技术开发。E-mail: xie_jiong@163.com

的空间数据无缝组织的新模式,正在普遍地取代将属性数据和图形数据分离地存储于 RDBMS 和文件管理系统中的传统数据组织模式^[2~4]。但是,这种新模式对于带有时变特性的“超海量”时空数据管理仍缺乏强有力的支持。与此同时,时态 GIS 领域尽管业已提出了大量的时空数据模型,但从当前的主流数据库技术及实际应用来看,由于扩展关系型时空数据模型(时间作为附加的属性数据)仍然占居主导^[5],因此,如何针对这类模型进一步改善数据组织与存储结构,以提高时空检索效率,仍是当前迫切而重点需要解决的问题。

本研究以扩充关系型时空数据模型为基础,对当前大型对象-关系型数据库平台所提供的数据分区与聚簇方法进行了时空维的扩展,并提出了一种基于时空分区聚簇(spatio-temporal partition clustering, STPC)的海量时空数据性能优化方法。根据时空数据的自身特点,该方法将时空分区与时空聚簇两种机制有机结合,实现了时空多维结构向 1 维存储结构的高效映射。这种时空分区聚簇总体上分为以下两个层次:第 1 层次根据时空域地理要素分布状况,将时空域划分为不同的时空分区,并使不同分区中的要素对象映射到不同的物理存储分区;第 2 层次是在每一分区内部,进一步采用“聚簇排序”方法来实现时空邻接要素在磁盘页面中的邻接存储。

2 时空分区

2.1 总体思想

时空分区是一种基于地理要素时空分布特征的区域描述。为便于分析,时空分区逻辑上可降维到空间分区与时态分区。图 1 为某区域土地利用要素

空间分区(spatial partition, SP)示意图,其按照数据分布特点划分为 6 个空间分区,即 SP0001 至 SP0006。图 2 为土地利用要素时态分区(temporal partition, TP)示意图,图中描述的是某一区域在不同时段的土地利用类型发生变化的图斑数量(横坐标为有效时间轴,纵坐标为图斑数量)。通过分析可见,1991 年至 1994 年存在有变化的高峰期,可以将该时间区段划为 TP1 分区;1994 年至 2000 年数据发展平稳,可划为 TP2 分区;2000 年出现历史时态真空区间,此时土地利用未发生任何变化,2001 年至 2004 年数据量又出现大的波动,则可以将 2000 年至 2004 年归并为 TP3 分区。分区的划定可以参考如下几点:

- (1) 数据量接近准则,即不同分区的数据量相对均衡,主要防止过大数据量的分区出现;
- (2) 专题聚集准则,即根据专题数据的空间聚集状态进行划分,数据集中的区域归并到一个区,由于专题的相对聚集分布往往具有内在关联,因而有利于数据分析的进行;
- (3) 语义归并准则,即不同空间区域或时间区段的专题数据,往往存在人为限定的一些语义分类,如东部沿海地区、东北地区的划定,某一行业规则出台后的一定时间段等,分区的划分应尽量考虑这些数据的潜在语义分类信息,以有利于数据的检索和查询。

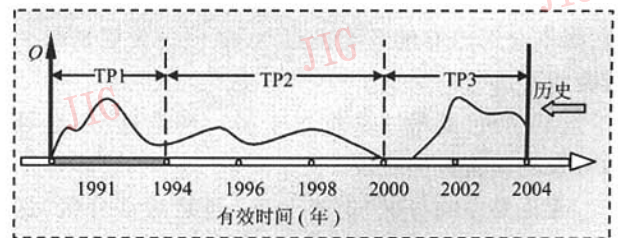


图 2 土地利用要素时态分区示意图

Fig. 2 Temporal partitions with land use features

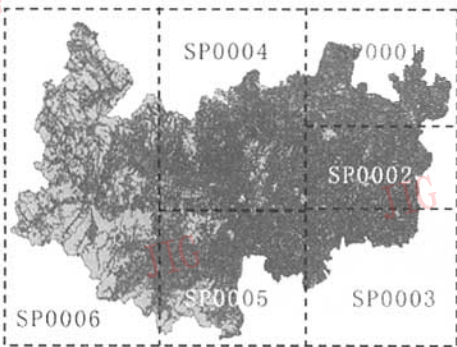


图 1 土地利用要素空间分区示意图

Fig. 1 Spatial partitions with land use features

若时空结合,则形成时空分区。时空分区(spatio-temporal partition, STP)是按照不同时空区间的专题数据量的大小和专题数据的时空分布聚集状态等来将目标时态空间划分为若干时空单元(spatio-temporal unit, STU),并标定每一地理要素对象所属的时空单元的过程。

时空分区由时态分区和空间分区复合而成,其 3 维分区模型见图 3。假设该图描述的是某一地理要素类(feature class)的时空分区结果,则将由 x, y

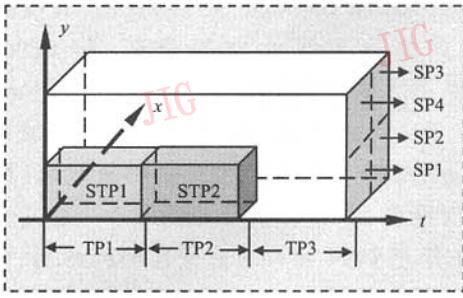


图 3 时空分区模型

Fig. 3 Model of STP

轴所构成的 2 维空间划分为 4 个空间分区,分别为 SP1 至 SP4;时间维 t 划分为 3 个时态分区,分别为 TP1 至 TP3,时空组合则划分出 12 个时空单元,分别为 STP1 至 STP12(图中仅表示了 STP1 和 STP2)。时空分区的划分准则同样遵循数据量接近准则、专题聚集准则和语义归并准则。

时空分区的基本特点为:

- (1) 是一种区划描述,由于其是对由空间维与时间维所组成的时态空间进行划分,因此具有时空多维特性;
 - (2) 每一要素对象必需记录所属分区信息,这与时空分区作用原理有关;
 - (3) 以基于数据库的无缝大表一体化存储时空数据为前提;
 - (4) 是一种动态描述性分区,如果重新划定分区,则不会对已有的数据造成破坏,但需要更新每一对象的所属分区信息。
 - (5) 分区是基于数据之上的一种描述,不会导致对象的割裂存储现象。
- 无论是空间分区、时态分区,还是时空分区,这

些概念的提出并非为了单纯引入一种新的数据区划策略,其目的在于提高海量时空数据的存取效率和数据安全性。时空分区的作用原理是,将不同的时空单元数据通过编码映射到数据库表分区,以便利用数据库分区技术来提高数据操作的并发度、数据安全性及数据存取效率,同时,通过时空分区一级的过滤机制,时空查询可以直接指定分区查询,并可进一步结合分区内索引机制来提高海量时空数据的检索效率。

2.2 数据库表分区技术

企业级数据库以及数据仓库等通常需要维护和管理 GB 甚至 TB 级海量数据,这类数据库为超大规模数据库(very large databases, VLDB),而支持 VLDB 海量数据管理的关键技术之一即为分区技术(partitioning)。现以大型商用数据库平台 Oracle 为例进行分析。分区技术采用一个或多个字段组成分区键(partition key),其可以先将大型数据表(分区表)或索引(分区索引)分解为更小的、更易于管理的若干区块,即分区(partition),然后允许将这些分区存放不同的物理存储单元(如磁盘一级)上,但逻辑上仍维护一个完整的表或索引,以使不同分区的数据共享相同的字段及约束定义^[6]。一旦定义了分区,则 SQL 查询、操作或定义语句都可以直接针对特定分区进行。图 4 为普通表和分区表的结构示意^[5],该图显示了以下特点:首先,无论是普通表还是分区表,都可以创建分区索引;其次,分区表假设以产品数量字段 f 为分区关键字段来划分为 3 个区, $f < 10\,000$ 划分到 P_1 分区, $10\,000 \leq f < 20\,000$ 划分到 P_2 分区,而 $20\,000 \leq f < 30\,000$ 则划分到 P_3 分区。

使用分区机制的目的,一方面主要是为了提高数据管理操作和查询的效率,另一方面是为了提高

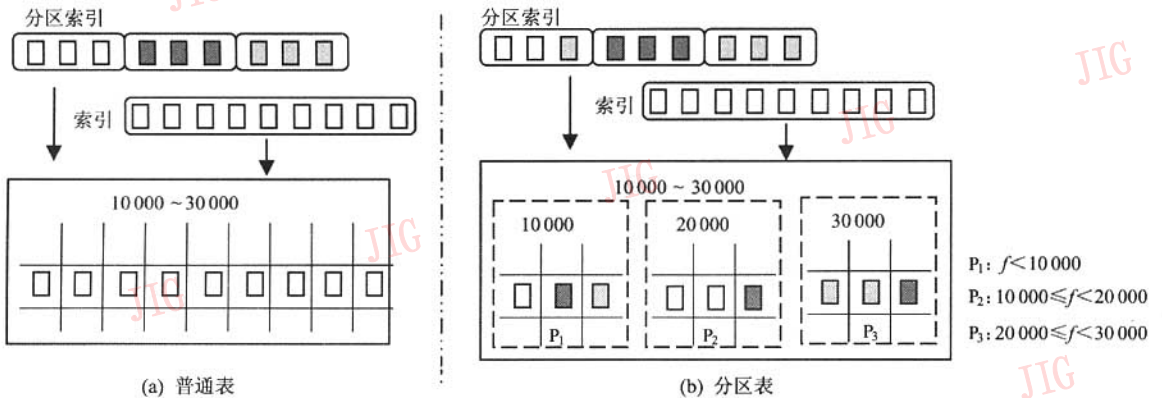


图 4 普通表和分区表的结构示意

Fig. 4 The structure of normal table and partition table

数据的可用性和安全性。

2.3 时空分区对表分区的映射关系

关系表分区机制为海量数据的存储和管理提供了技术基础。假设每一个要素对象存储为一个元组,则将时空分区映射到表分区的总体思路是,首先需要对时空单元进行编码,然后判断每个对象归属于哪个时空单元,最后将所属时空单元的编码作为对象所在元组的分区键值进行存储,以实现时空分区到表分区的映射。

时空单元编码(即时空分区编码)采用时、空编码复合的方法生成,复合公式如下:

$$C_{stp} = C_{vt} + C_{sp} \quad (1)$$

其中 C_{stp} 代表时空单元编码, C_{vt} 代表有效时间编码, C_{sp} 代表空间分区编码。有效时间是指实体在现实世界为真的时间,为简单起见,若时间粒度取“年”为单位,则 C_{vt} 即为年份。以图2为例,共划分为3个时态分区,TP1的 C_{vt} 为1994(年),TP2的 C_{vt} 为2000(年),TP3的 C_{vt} 为2004(年)。这里可以以地理要素的开始有效时间点作为时态分区的参考点,即按照对象的产生时间为参考来确定其归属于哪个时态分区。

C_{sp} 可按4位十进制数顺序编码,共可以表达

9999个空间分区,一般足够。以图1为例,共划分为6个空间分区,其 C_{sp} 分别为0001~0006。

由以上实例的3个时态分区和6个空间分区结合可得到18个时空单元,再将两者编码复合即为相应的时空单元编码,图5为时空分区结果,图6为时空单元编码映射到分区关键字的结果。

对于空间上或时态上跨分区的地理要素,有以下两种处理的基本策略:

(1) 将所有跨分区的地理要素单独放于某个特定的编码分区中。查询时,对于能够确定分区的,则直接指定按分区查询,否则,按普通查询处理。这样做的优点是,当查询能够确定地理要素的具体分区时,由于只需要查询两个分区,即只查询所在的分区和跨分区地理要素存放的分区,因此查询速度快;其缺点是,管理上增加一定的复杂性,且需要对用户的相关SQL进行一些系统的内部处理。

(2) 依据某种策略来将跨分区的地理要素放于其所跨的某一个分区中,如利用空间维的面积占优法或取首点法来确定空间分区归属,而有效时间维则用创建时间点或生命周期中间时间点等来确定时态分区归属。该策略的优缺点与前一个策略的优缺点相反。

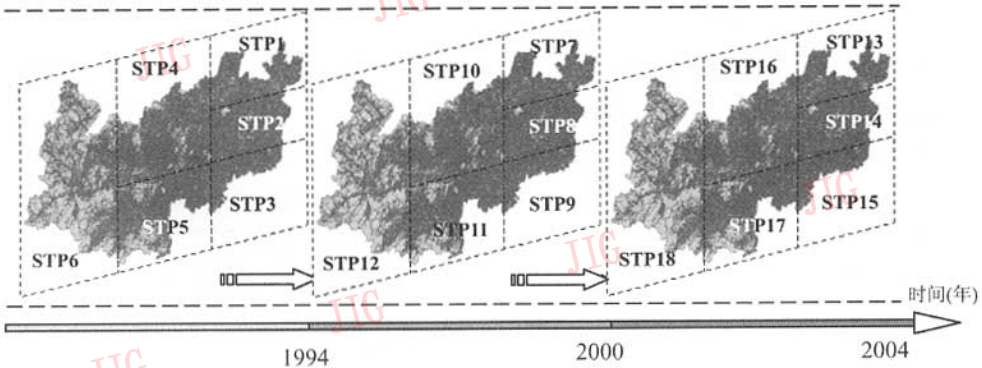
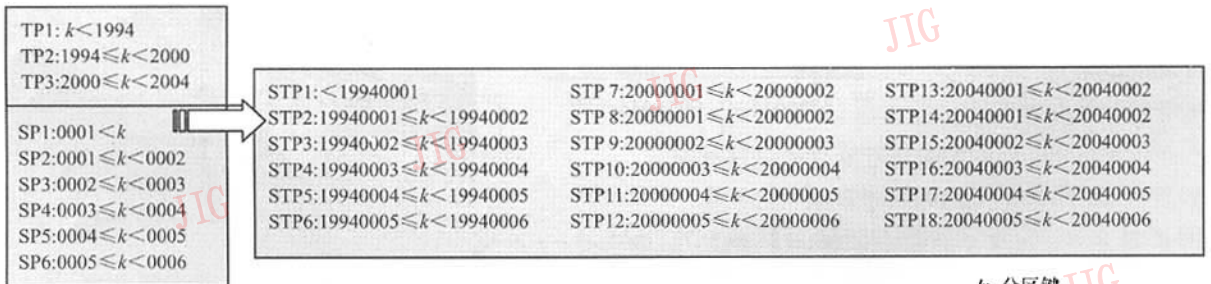


图5 土地利用要素时空分区结果

Fig. 5 Spatio-temporal partition result of land use features



k: 分区键

图6 时空分区编码映射到分区键

Fig. 6 Mapping of STP code to partition key

3 时空分区聚簇

3.1 总体思想

时空分区映射关系表分区虽可以使时空聚集要素分块集中存储,但在每一个时空单元所对应的存储区块内部,地理要素元组在存储页面中的排列仍处于无序状态,即虽时空相邻,但存储位置并不相邻。时空聚簇(spatio-temporal clustering, STC)可使空间上相

邻,且生命周期相近或相关联的地理要素实现物理存储页面上的相邻存储,这将有效地减少时空查询时的磁盘页面往返搜寻的时间,以提高时空检索效率。但测试表明,单独的时空聚簇机制并不能取得最佳的时空检索效率,而只有将时空分区与时空聚簇结合,在分区之上聚簇,以形成时空分区聚簇,才能获得最佳的时空检索效率。时空分区聚簇的效果见图 7,图 7(a)为某时空单元内的地理要素分布,图 7(b)为地理要素记录在页面中的聚簇排序效果。

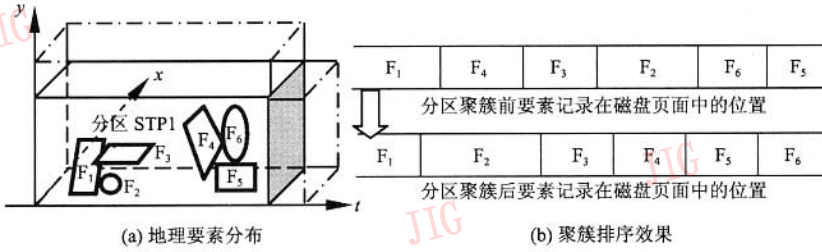


图 7 时空分区聚簇的效果

Fig. 7 Effect of spatio-temporal clustering

时空分区聚簇是对时空分区的内部对象进一步通过聚簇排序算法来使空间上相邻且生命周期相近或相关联的地物要素在物理存储页面上也相邻的过程。其作用原理是,先通过一定的算法对时空单元(时空分区)内的地理要素进行邻近顺序编码来保持时空邻近对象编码值接近,然后将编码值作为数据库聚簇键(cluster key)值进行存储,最后插入到一个主码文件结构中,再通过 B+ 树聚簇排序来实现时空聚簇的目的。

3.2 时空分区聚簇编码

时空分区聚簇编码是在时空分区编码的基础上,进一步对分区内对象进行时空聚簇编码,其目的

是使时空多维结构按邻接性映射到 1 维结构,以便为基于 1 维结构的记录排序建立基础。时空聚簇编码可以采用时态聚簇编码和空间聚簇编码复合的方法生成(图 8),见式(2):

$$O_{stc} = O_{tc} + O_{sc} \quad (2)$$

其中, O_{sc} 代表对象时空聚簇编码, O_{tc} 代表对象时态聚簇编码, O_{sc} 代表对象空间聚簇编码。这里的对象是指元组所表达的地理要素。假设以线性时间结构为前提,则对象时态聚簇编码的方法是,首先对该时态分区所占据的时间区间内部指定分段数,每一段为一个时间格(time grid),每一时间格赋予一个编码,如针对时态分区 TP2 划分为 3 段,分别编码为

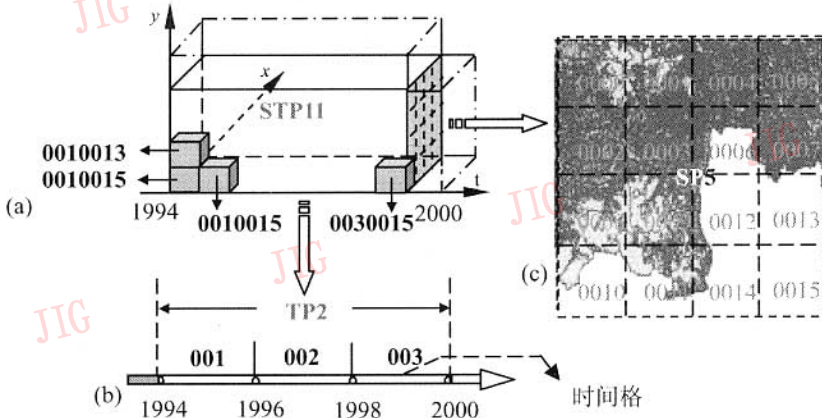


图 8 时空单元中进行时空聚簇编码

Fig. 8 Spatio-temporal cluster coding in SPU

001、002 和 003 ,然后根据对象的生命周期,按一定的标准(如取生命周期的中点来判断所落入的时间格等)将所在时间格的编码作为对象的时态聚簇编码。这里分段数的指定一般是根据时空分区的时态跨度来确定,并与具体所选择的时间粒度有关。

空间的多维特性,致使对象的空间聚簇编码相对更为复杂。编码的目的是为了使空间邻近对象的编码值接近。这里,作者借鉴 Morton 编码的思想来实现 2 维空间下如何生成地理要素的聚簇编码。

Morton 码是对空间格网进行编码,而在此是要对地理要素编码,因此,要对生成格网 Morton 码的算法进行演变。生成要素聚簇码的算法分以下两个步骤:

(1) 根据四分深度(FTLevel)和空间分区范围进行十进制的 Morton 编码(I 行、 J 列的网格,其 Morton 码记为 C_{Morton})。图 8(c)为对空间分区 SP5 进行 FTLevel 为 2 的 Morton 码的编码结果。 C_{Morton} 的计算方法可以采用数学公式法^[7]。如 2 维空间下,先将十进制的行号和列号转换为一种伪码,行号伪码 f_i 的计算公式如下:

$$\begin{cases} f_i = \text{MOD} \sum_{k=0}^{\log_2(I)} (I_k / 2^k) \times 4^k \\ I_k = I & \text{when } k = 0 \\ I_k = \text{INT}(I_{k-1} / 2) & \text{when } k > 0 \end{cases} \quad (3)$$

式中,MOD 为取余函数,INT 为取整函数, I 为十进制行号, k 为中间循环变量。同理可计算列的伪码 f_j 。最后用式(4)计算网格的 C_{Morton} 。

$$C_{Morton} = 2 \times f_i + f_j \quad (4)$$

另一种更快的算法即按位取值法。设十进制的行号 I 和列号 J 分别以二进制表示为

$$I = (i_n i_{n-1} \dots i_3 i_2 i_1) \quad J = (j_n j_{n-1} \dots j_3 j_2 j_1)$$

则十进制 C_{Morton} 可以直接通过行号 I 和列号 J 的二进制位交叉结合来获得(如图 9 所示)。

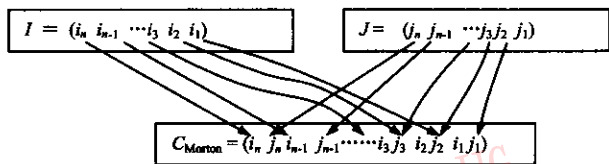


图 9 通过二进制位交错获得 Morton 码^[7]

Fig. 9 Get Morton code by intercross of binary digits

(2) 对分区中的每一个地理要素,可先采用一定的准则来判断要素归属于哪个网格,然后将该网格 C_{Morton} 作为该要素的 Morton 码。对于点状地理要素

若其落入某一个 Morton 码格网内(如图 10(a)所示)则其编码为此网格的 C_{Morton} ;若其落在格网的边线上,则任取一个以该边线为边的网格的 C_{Morton} 即可。对于线状、面状要素,若其完全落入某一个 Morton 码网格内(如图 10(b)中的 L_1 、图 10(c)中的 P_1)则其编码为此网格的 C_{Morton} ;若其占据了 n 个网格(图 10(b)中的 L_2 、图 10(c)中的 P_2),则取所占网格数最多、最邻近树底层的 C_{Morton} ,并在其后补 0,当形成与叶子节点位数相同编码后,则将其作为此要素的编码。

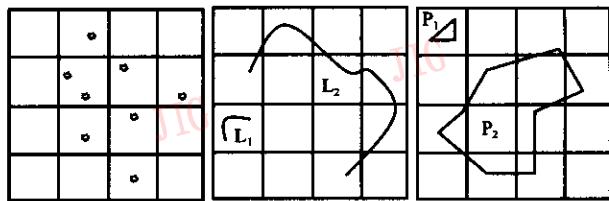


图 10 Morton 码的格网与地理要素

Fig. 10 Morton grids and features

如果要求保证地理要素的编码唯一,则可以在基于第 2 步形成的地理要素编码之后追加码段以保证编码的唯一性,追加的码段可以表示它是此网格内的第几个要素。

综上所述,参考图 8,STP11 分区将时间格和空间网格相结合,共划分出 16×3 个体素,落入每一体素内的要素对象都具有相同的时空聚簇码,如 0010015 等。

最后,作者将时空分区编码所对应的分区键和时空聚簇编码所对应的聚簇键集中统一到一个字段中,该字段即时空分区聚簇字段(spatio-temporal partition clustering field,缩写 STPCF),其结构为

$$STPCF = C_{stp} + O_{stc} \quad (5)$$

其中, C_{stp} 为时空分区码,对应表的分区键; O_{stc} 为对象时空聚簇码,对应表的聚簇键。通过 STPCF 字段,最后每一时空元组形成如下格式(ID, Attr1, Attr2, ..., Shape, STPCF)。STPCF 字段既是分区键,又是聚簇键,然后通过该字段值实施 B+ 树排序,即可达到时空分区聚簇的目的。

进一步,作者根据海量数据存储的要求,对 STPCF 的具体存储位进行了设计,其结构见图 11。图中 STPCF 字段前 8 位为时空分区码,后 18 位为时空聚簇码。时空分区码的前 4 位为地理要素的开始有效时间点,后 4 位是空间分区码,假定每个分区

为 1.0GB ,则最多可以存储 10TB 左右的单表 ;时空聚簇码的前 3 位为时态聚簇码 ,其 3 位长度可最多表示 999 个时间格 ;后 15 位为空间聚簇码 ,可以表示的四叉树的最大深度为 24。

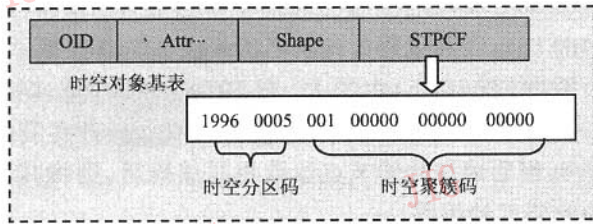


图 11 时空分区聚簇字段结构

Fig. 11 Structure of spatio-temporal partition clustering field

4 测试结果及分析

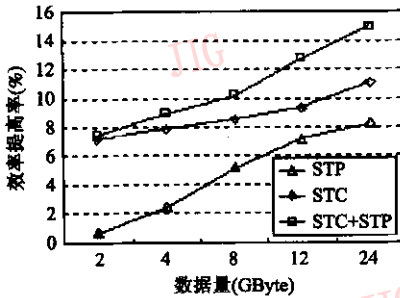
测试旨在对本文所提出的 STP、STC 和 STPC 机制对系统响应效率的提高程度进行评定 ,其中所基于的不同数据组织与存储模式包括 :普通数据组织、时空分区组织、时空聚簇组织和时空分区聚簇组织 4 类。测试主要针对的是多时态的地理要素数据 ,几何属性采用矢量方式 ,数据量为数据库下的单表

主体数据和对应的时态、专题及空间索引等辅助数据的总和 ,数据量从 2G 到 60G 不等 ,共划分 8 个等级。值得提及的是 ,测试评估的数据量是单表数据量 ,一个 60G 的单表已将近包含了 2 700 万条记录。

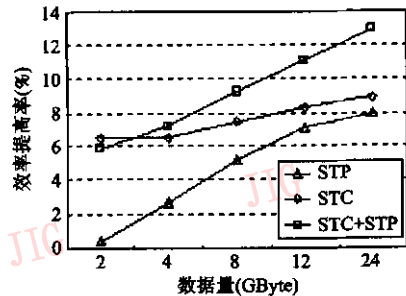
本文实验是对时属数据查询和时空数据查询进行了各自的比较测试 ,时属数据查询最高单表数据量达到 60GB ,时空数据查询最高数据量达到 24GB。需要说明的是 ,在大数据量的时空测试过程中 ,随着数据量的增大 ,空间索引创建过程相当耗费机时 ,单 2G 的大表重建分区空间索引就需约 50min ,且随着数据量的增大 ,空间索引还时常出现创建中途失败的异常 ,这给测试工作带来相当大的困难。基于以上原因 ,海量时空数据查询的最大单表测试数据量定为 24G ,而更大数据量的测试则有待环境配置的改善及 Oracle Spatial 底层库版本的进一步改进。以下本文给出时空数据查询的测试结果。图 12(a)为目标记录不跨分区的时空数据查询结果 ,图 12(b)为跨分区时空数据查询结果。图中的纵坐标为效率提高率 ,其计算公式为

$$R = (t - t_1) / t_1 \times 100\%$$

其中 t 为普通无缝大表数据组织下的执行时间 , t_1 为优化后执行时间。



(a) 分区内 STP 与 STC 时空数据查询效率提高率曲线



(b) 跨分区 STP 与 STC 时空数据查询效率提高率曲线

图 12 STP、STC 与 STPC 时空数据查询效率提高率曲线

Fig. 12 Curve of spatio-temporal query performance improving ratio of STP , STC and STPC

根据以上时空查询测试结果算出的平均效率提高率见表 1 ,综上所述 ,可初步得出以下结论 :

(1) STP 机制随数据量的增大 ,效率提高的程度表现逐渐突出 ,比较适合(超)海量数据组织 ,但在相对小数据量下 ,则表现不出优越性 ,这一点与时属查询类似。同样 ,在跨分区执行频繁的搜索环境下 ,需要进一步通过多 CPU 分区并行检索来提高性能 ,这点尚需要进一步的实验验证。

(2) STC 机制在时空检索方面总体表现良好 ,

且在不同数据量级别下 ,都能在一定程度上通过减少磁盘页面的往返搜寻操作来提高数据检索效率。在跨分区检索条件下 ,时空检索效率提高率下降 ,推测其与分区之间的往返搜寻时间增加有关。

(3) 无论查询目标是否跨分区 ,STP 与 STC 机制对时空检索效率的提高都具有一定的效果 ,从平均来看 ,STC 机制的时空检索效率(8.2%)要明显高于 STP 机制的时空检索效率(4.7%) ,但前提是最大数据量为 24G。

表1 时空查询平均效率提高率

Tab.1 Spatio-temporal query performance improving proportion

组织与存储机制	分区内平均效率提高率 (%)	跨分区平均效率提高率 (%)	总体效率提高率 (%)
时空分区(STP)	4.7	4.7	4.7
时空聚簇(STC)	8.8	7.5	8.2
时空分区聚簇(STPC)	10.8	9.3	10.1

(4) STC 与 STP 两种机制结合下的 STPC 机制具有更高的时空检索效率(10.1%)。

5 结 论

本文基于扩展关系型的数据组织方式,以矢量结构和线性时间结构为前提,对海量数据的分区和聚簇方法进行了时空维的扩展研究,同时提出了基于时空分区聚簇的性能优化方法,并通过测试证明了该方法的有效性。下一步工作将针对其他类型的海量时空数据进行方法的扩展和完善,以便更好地解决海量时空数据的高效管理问题。

参考文献(References)

1 FANG Yu , ZHOU Cheng-hu , JING Gui-fei , et al. Research of 4th

generation GIS software[J]. Journal of Image and Graphics , 2001 , 6A(9) : 817 ~ 823. [方裕,周成虎,景贵飞等.第四代GIS软件研究 J]. 中国图象图形学报,2001,6A(9):817~823.]

2 LIU Nan , LIU Ren-yi. Geographic information system[M]. Beijing : China Higher Education Press , 2002 : 108 ~ 113. [刘南,刘仁义.地理信息系统 M]. 北京:高等教育出版社,2002:108~113.]

3 LI De-ren , GUAN Ze-qun. Integration and implementation of spatial information system[M]. Wuhan : Publishing House of Wuhan Technical University of Surveying and Mapping , 2000 : 197 ~ 198. [李德仁,关泽群.空间信息系统的集成与实现 M]. 武汉:武汉测绘科技大学出版社,2000:197~198.]

4 LI Ai-qin , GONG Jian-ya , LI De-ren. Seamless organization of GIS large geographic database [J]. Journal of Wuhan Technical University of Surveying and Mapping , 1998 , 23(1) : 57 ~ 61. [李爱琴,龚建雅,李德仁.大型GIS地理数据库的无缝组织 J]. 武汉测绘科技大学学报,1998,23(1):57~61.]

5 WU Li-xin , SHI Wen-zhong. Theory and algorithm of geographic information system[M]. Beijing : Science Publishing House , 2003 : 111 ~ 118. [吴立新,史文中.地理信息系统原理与算法 M]. 北京:科学出版社,2003:111~118.]

6 Oracle Corporation. Oracle9i database concepts release 2(9. 2) [EB/OL]. URL : http : //www. lc. leidenuniv. nl/awcourse/oracle/server. 920/a96524/c12parti. htm#464767. 2005-01-25.

7 GONG Jian-ya. Basic of geographic information system[M]. Beijing : Science Publishing House , 2003 : 106 ~ 108. [龚建雅.地理信息系统基础 M]. 北京:科学出版社,2002:106~108.]