

大规模空间矢量数据分布式存储与计算优化^①



张嘉¹, 白晓飞¹, 陶超², 张小桐¹

¹(中国国土勘测规划院, 北京 100035)

²(广东南方数码科技股份有限公司, 广州 510665)

通讯作者: 张嘉, E-mail: sagehn2003@163.com

摘要: 针对海量空间矢量数据分布式存储与计算需求, 研究了基于四叉树格网编码建立要素索引的方法, 设计了HBase 预分区优化策略, 提出了一种空间矢量数据分布式存储模型. 基于 MapReduce 计算框架, 构建了空间数据分布式计算与分析的优化流程. 最后, 针对空间叠加与统计场景, 采用一定规模的业务数据对所提的方法进行测试, 验证了设计方案的可行性和有效性.

关键词: 空间矢量数据; HBase; 格网编码; 预分区; 分布式计算

引用格式: 张嘉, 白晓飞, 陶超, 张小桐. 大规模空间矢量数据分布式存储与计算优化. 计算机系统应用, 2020, 29(12): 251-256. <http://www.c-s-a.org.cn/1003-3254/7724.html>

Storage and Computing Optimization of Large Scale Distributed Spatial Vector Data

ZHANG Jia¹, BAI Xiao-Fei¹, TAO Chao², ZHANG Xiao-Tong¹

¹(China Land Surveying and Planning Institute, Beijing 100035, China)

²(South Digital Technology Co. Ltd., Guangzhou 510665, China)

Abstract: Research on distributed storage and computing technology of spatial vector data is carried out. The method of quadtree grid coding to establish feature index is studied. HBase pre-partition optimization strategy is designed, a distributed storage model of spatial vector data is proposed. Based on MapReduce computing framework, the process of spatial data distributed computing and analysis is built. For the common application scenario of spatial overlay analysis and statistics, a large-scale data test is carried out. The results show that the scheme is effective.

Key words: spatial vector data; HBase; grid coding; pre-splitting; distributed computing

1 引言

国土数据类型多样、结构复杂、变化频繁, 具有很强的空间化、图形化特征. 近年来, 随着空间信息技术与资源环境管理工作普遍、深度交融, 地表基质、地表覆盖、地下空间、业务信息等各类国土空间数据日趋复杂, 涉及“山水林田湖草”及地质、矿产等数百类自然资源要素, 在地理空间上彼此映射、关联, 且数据规模迅速增长. 以全国土地调查矢量数据为例, 年度更新数据超过 2TB, 累计要素数已达 41 亿. 自然资源

政务审批和业务管理, 需要高效分析上述海量数据间的位置和语义关系以辅助决策, 这一过程往往需要进行大量复杂空间运算. 面向大规模空间数据分析, 传统关系型数据库的空间数据存储与计算框架愈发难以满足实际需要.

随着大数据技术快速发展, 以云计算为代表的分布式计算技术以其高性能和高可靠性, 特别是弹性扩展的数据存储与处理能力, 为大规模空间数据高效存储与计算提供了可靠路径^[1-3]. 其中, Apache Hadoop 项

① 基金项目: 国家重点研发计划 (2018YFD1100103-05); 自然资源部部委计划 (1902-033-14)

Foundation item: National Key Research and Development Program of China (2018YFD1100103-05); Program of Ministry of Natural Resources of the People's Republic of China (1902-033-14)

收稿时间: 2020-05-14; 修改时间: 2020-06-10; 采用时间: 2020-06-27; csa 在线出版时间: 2020-11-30

目作为普遍流行的开源分布式大数据应用框架,实现了 MapReduce 分布式数据计算模型和 HBase 分布式数据库等核心应用,支持基于集群的海量数据分布式存储和运算,在超大规模数据管理中得到广泛运用^[4].近年来,为解决矢量数据空间分析的基本难题,基于 Hadoop 应用生态构建了 SpatialHadoop、Hadoop-GIS、Spatial-Spark、GeoSpark 等扩展框架,为矢量数据存储、索引和计算提供了基础支持^[5].

数据组织作为处理和操作的基石,其结构和性能深刻影响统计、分析与服务效率,成为海量时空数据管理、应用的关键问题之一.针对 Hadoop 空间数据存储与计算,相关学者在数据组织、索引构建等方面进行了研究.王凯等提出了基于 Hadoop 的地理信息大数据处理模型,设计了矢量数据的格式转换方法^[6];范建永等利用 HBase 进行矢量空间数据管理,基于 Map-Reduce 构建了并行空间索引^[7];李振举等提出了一种四叉树-Hilbert 结合的索引设计方法^[8];朱进等提出了一种基于内存数据库 Redis 的轻量级矢量地理组织方法^[9].但当前研究主要集中在分布式数据存储的负载均衡,未对索引、分区和计算进行一体化考虑.

本文基于 HBase 数据存储与访问的特性,使用四叉树多级格网编码算法进行矢量要素编码,构建分布式空间数据索引;针对 HBase 缺乏对矢量数据模型有

效支持的问题,提出了一种通过行键编码链接索引、分区与计算的数据组织模型和分布式处理机制,并设计了分布式计算的优化流程,为大规模空间数据存储与处理提供了一种解决方案.

2 空间数据组织与优化

2.1 四叉树格网索引

本文使用四叉树多级格网算法对矢量数据进行编码构建索引.该算法基于平面直角坐标系划分象限编码,基本思想是将全域空间范围划分为4个相等的区域并递归执行,直到剖分层次达到指定深度^[10,11].其处理流程可简单描述为:第一,确定矢量要素集合的矩形空间范围作为基准格网并逐级剖分;第二,对剖分后的各级格网进行编码;第三,计算每个矢量要素的最小外接矩形;第四,根据最小外接矩形空间上映射的格网,使用格网编码对其递归赋值;第五,存储每个矢量要素的编码集合.可见,通过计算矢量要素外接矩形所在的格网编码,可快速获取该要素在相应格网层级的空间位置,如图1所示.格网的划分层级与要素的编码长度正相关,过多层级会导致要素的编码复杂性增大,并产生数据冗余.因此,格网编码的深度应根据数据特点进行设计,遵循的基本原则是确保大部分要素拥有一个编码,允许少部分的要素拥有多个编码.

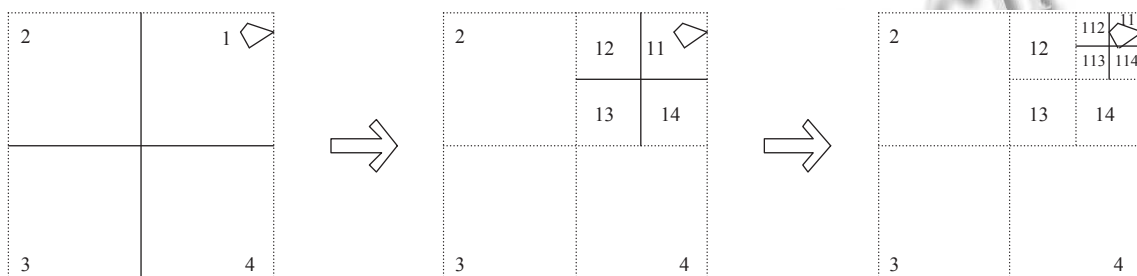


图1 四叉树编码示意图

2.2 HBase 预分区设计

将数据均衡分布到集群计算节点,是充分利用分布式计算资源的必要条件.HBase 以 region 作为数据扩展和负载均衡的基本操作单元,分配到不同的计算服务器.HBase 表中的数据按照行键 (RowKey) 排序,每个 region 通过记录起止行键管理表中的某些行数据^[12].表首次创建时,默认只分配一个 region,此时所有的读写操作都集中于这个 region 所在的服务器,从而产生“热点”(hot spotting) 问题,导致该服务器负载急剧上升、性

能下降;同时,region 中导入的数据量达到一定阈值会触发分裂 (region-split),即找到一个中间行键 (MidKey) 将其拆分为两个 region,当输入大规模数据时,分裂次数快速增加将大量占用磁盘 I/O 资源,降低集群性能.针对上述问题,本文根据矢量数据的格网索引机制设计预分区策略,分区格网采用四叉树剖分,当预设 $4n$ ($n \geq 1$, n 为剖分层级) 个分区时,以第 n 级格网编码作为预分区号,如图2.预分区号作为行键编码的前缀,数据入库时可自动根据该编码将要素均衡地向多个 region 划分.

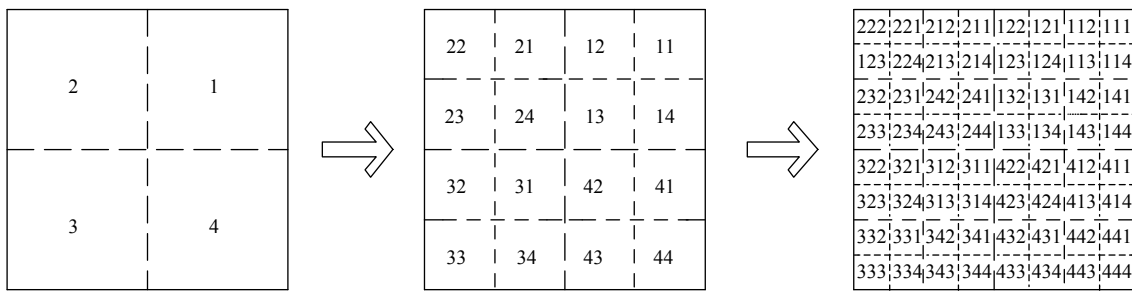


图2 格网分区示意图

使用 MapReduce 框架进行并行计算时, 分区数量决定了 Map 函数的数量, 即每个分区将启动一个 Map 函数进行计算, 为保证计算和负载均衡, 需合理预设分区数量, 并考虑将相邻空间要素聚集. 分区个数过小, 会造成单个分区的数据量过大, 不利于负载均衡; 分区个数过多, 会导致单个分区内数据量不足, 不利于集中处理. 实践中, 应根据计算资源和数据量采用适当的分区数, 原则上是充分利用集群的分布式特性, 进行大范围空间计算时, 尽可能减少节点间数据传输, 并使局部范围计算在单个节点完成, 以节约 I/O 资源. 一般而言, 可以以集群中 region server 数量的倍数作为参考值.

2.3 空间数据分布式存储设计

2.3.1 行键编码方法

HBase 作为面向列存储的数据库, 通过行键检索数据的效率较高, 每秒钟可获取 1000~2000 条记录, 而检索非 Key 列时效率相对偏低. 因此, 建立空间数据格

网编码与行键编码的关联, 使行键编码集成空间信息, 是提高空间检索效率的关键环节. 本文提出以预分区号、要素格网编码和随机码组合构建行键编码的方法, 该编码集成了集群和要素空间信息, 便于分布式处理.

$$RowKey = Num[i] + GridCode[i] + Random[i]$$

GridCode 为格网编码, 使用 2.1 节中通过格网编码算法生成的定长字符串; Random 为随机码, 根据实际应用中数据集的大小确定, 用于保证行键编码的唯一性; 预分区号 (Num) 根据 2.2 节确定的方法获取, 并根据格网剖分后空间相邻矢量要素编码通常前缀相同的特点进行优化: 仅有一个编码的要素, 直接以其编码值的后两位作为预分区号; 有多个编码的要素, 采用多个编码共同前缀的后两位作为预分区号. 这种方法实现了索引、分区、计算的一体化设计 (如图 3), 使地理空间临近的数据相对聚集, 整体数据较均衡分布到各数据节点, 有助于提升空间分析的计算效率, 同时便于索引的动态更新维护.

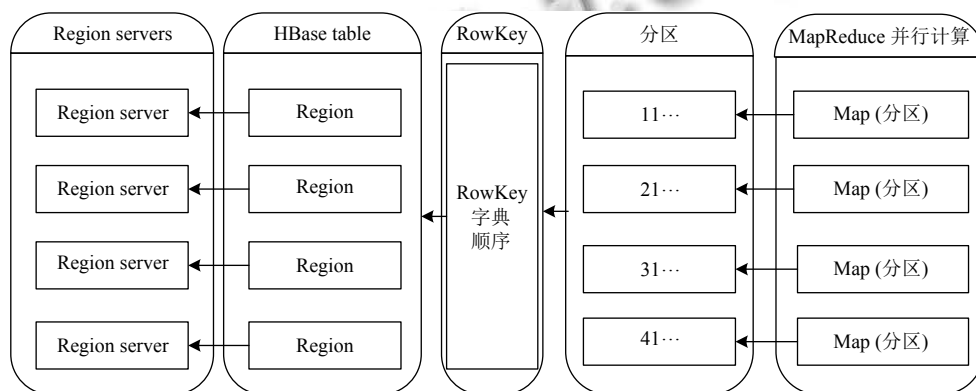


图3 行键编码与数据存储计算

2.3.2 矢量数据表存储

基于上述行键编码方法, 空间矢量数据在 HBase 中的表存储设计如表 1. 其中, $num[i]+gridCode[i]+$

$random[i]$ 构成行键 (“+”表示连接, 行键字符串中不含该符号) 唯一标识表中的一条记录. 矢量要素存储表设计了一个 Column Family(列族) 矢量信息, 该列族下按

照要素字段定义 Qualifier(列限定符), 列限定符与原矢量数据属性名称保持一致, 存储图斑编号、地类、权属等信息, 其中 SHAPE 存放 JSON 格式的坐标数据. 至此, 实现了土地空间矢量数据从关系型数据组织向分布式数据存储的转换.

表 1 空间数据存储表结构设计

RowKey	Column Family: 矢量信息		
	Qualifier: TBBH	Qualifier: ...	Qualifier: SHAPE
num1+gridCode1+random1	tbbh1	...	{[[[x0,y0], ..., [x0,y0]]]}
num2+gridCode2+random2	tbbh2	...	{[[[x1,y1], ..., [x1,y1]]]}
num3+gridCode3+random3	tbbh3	...	{[[[x2,y2], ..., [x2,y2]]]}

3 空间数据分布式计算

3.1 预处理与计算优化

(1) 重复计算规避. 基于格网的空间数据存储本质上是一种以空间换时间的设计模式. 矢量要素空间大小不同, 格网编码算法生成的编码值个数也随之不尽相同, 导致存入 HBase 表中的同一要素可能存在多个记录, 不利于 MapReduce 并行计算^[13]. 为降低重复数据对计算性能的影响, 本文将同一要素的多个记录存储在同一分区, 通过矢量要素标识信息, 在 Map 函数计算阶段构建缓存记录参与计算的要素, 进行去重操作, 避免了数据重复计算和传输.

(2) 空间数据关联映射. 针对参与空间计算的多元矢量数据集, 按照格网编码方法进行统一编码, 矢量要素采用本文提出的行键编码算法构建数据对象间的编码映射. 计算时对无编码映射关系的空间数据进行过滤, 可有效降低计算数据的规模. 同时, 通过要素编码可快速查找关联要素, 避免全表要素扫描, 从而大幅减少程序遍历次数, 提升了分布式计算的性能.

3.2 分布式计算流程

分布式计算一定程度上可以有效改善传统模式下处理大规模空间数据计算能力不足的问题. 以空间叠加操作为例, 典型的业务场景为: 针对存储于分布式数据库中的地类图斑矢量要素, 使用外部存储的、对应县级行政区划内的监测图斑与之进行空间叠加, 将叠加得到的裁切后矢量数据输出, 计算要素面积, 并统计某类属性下的土地面积汇总数. 以此为列, 基于上述数据组织策略, 本文设计了分布式计算方法, 主要思路为:

基于 MapReduce 计算框架, 结合 Esri GIS Tools for Hadoop 工具, 实现空间数据 Map 函数计算, Reduce 函数执行结果汇总并输出. 具体处理流程为: (1) 将待叠加矢量要素转换为 JSON 文件, 读入 HDFS 存储, 对其进行格网编码; (2) 建立与被叠加地类图斑矢量要素的编码映射, 获取关联的要素集合; (3) 将关联集合加载到分布式缓存中; (4) Map 函数逐条读取地类图斑矢量数据, 根据要素编码获取分布式缓存中集合中的对应要素, 并执行空间叠加操作; (5) 输出裁切后的数据, 并将过程数据传送 Reduce 函数处理; (6) Reduce 函数完成计算结果的汇总及输出. 总体流程如图 4.

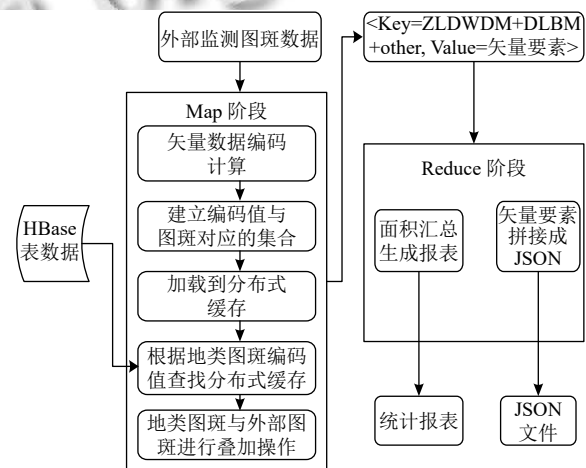


图 4 空间数据裁切操作流程

4 测试与分析

针对本文提出的格网编码索引、预分区及行键优化设计的空间数据组织策略和计算流程, 使用真实数据, 采用空间叠加与统计场景进行测试, 验证方案的有效性.

4.1 测试环境

基于 Hadoop 2.7.3 搭建一个主节点, 4 个数据节点的集群环境 (图 5), 在该集群部署 HBase 1.3.1. 数据节点服务器配置为: CPU Intel(R) Xeon(R) CPU E5-2609 v4 @ 1.70 GHz; 内存: 128 GB; 操作系统: CentOS7.3; 网络带宽为 1000 Mb/s.

4.2 测试过程与结果分析

4.2.1 测试数据

180 个县级行政区划单位的地类图斑 (DLTB) 和监测图斑 (JCTB), 两类数据都是空间矢量要素. 为便于对比, 将其分为 8 个测试组, 每组所含要素和数据量表 2 所示.

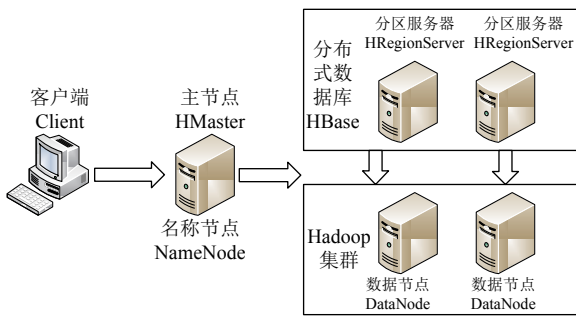


图5 测试环境

表2 数据分组及要素数

测试组号	区县个数	监测图斑个数	地类图斑个数
1	3	213	4902
2	8	5772	128828
3	15	10929	459231
4	18	15225	596246
5	30	21858	918462
6	80	66672	2801523
7	150	125695	4922505
8	180	152250	5962460

4.2.2 测试内容与流程

自然资源管理中,地类图斑刻画了土地利用的现状情况,按照规定的时点更新.遥感监测是维护土地现状信息准确性和现势性的重要手段,通过对比、判读不同时相的遥感影像,能够识别发生变化的国土特征信息,这些变化信息经矢量化形成监测图斑.业务实践中,一种典型的应用场景是获取监测图斑空间范围内变更前的土地利用类型和面积等数据,以辅助数据更新、变化分析和行政监管.本文将监测图斑与地类图斑进行空间叠加,获取监测图斑的地类面积、权属等信息,测试研究提出的分布式存储与计算方案下空间数据分析和汇总统计的效率,主要流程如下:(1)将地类图斑数据作为叠加底图,创建格网索引,存储在HBase分布式数据库;(2)以监测图斑作为外部叠加数据,采用GIS Tools for Hadoop插件工具集将其转换为以县级行政区为单位的JSON文件,存储于HDFS分布式文件系统;(3)基于MapReduce框架对监测图斑和地类图斑进行叠加,提取地类信息并计算面积,汇总形成监测图斑地类面积汇总表,并输出叠加后的图形数据,验证其正确性.

4.2.3 结果分析

按照测试流程对8个测试组按照执行数据操作,记录数据上传、数据转换、空间叠加、统计输出几项主要处理环节的耗时情况,如表3.

表3 测试耗时结果记录(单位:s)

测试组号	数据上传	JSON数据转换	叠加	统计输出	总时间
1	0.03	0.23	61.21	0.32	62.44
2	0.27	0.81	83.15	2.90	85.96
3	1.79	1.99	101.36	5.96	106.35
4	2.37	2.94	118.09	9.48	125.03
5	2.95	3.56	198.46	13.13	207.02
6	13.27	8.36	545.21	41.20	559.57
7	23.74	17.49	1010.77	78.00	1035.26
8	27.51	21.44	1170.32	94.00	1199.76

对表3数据进行分析,结果表明:

(1)数据处理耗时与数据量正相关,但随着数据量的增加,计算效率呈现阶段特征.在地类图斑数由5000向50万量级增长过程中,处理要素的平均耗时显著下降,由平均0.29秒/要素降为0.01秒/要素,说明单位时间系统处理要素的能力在迅速提升,充分体现了分布式环境处理大规模数据的优势;当要素数据超过50万后,单位时间处理要素的数量变化曲线基本平滑,这是由于服务器数量固定导致计算资源受限,可以预测增加计算节点,能够进一步提高性能.

(2)空间叠加环节在总体分析流程中的耗时居支配地位,且远高于其他环节,占比区间为94.45%至98.02%,这说明空间数据并行计算是数据分析的核心,也是计算优化的重点.

根据上述分析结果,进一步测试本文提出的索引、预分区及行键编码一体化设计的空间数据组织策略(记作GRID),与常规采用随机散列构建行键编码方法(记作RHASH)的空间分析效率对比.为降低计算资源对数据处理效率的影响,测试数据选择18个县级行政单元的地类图斑和监测图斑矢量要素(共90万个要素),分为5个测试组,记录空间叠加的处理时间,如表4所示.

表4 空间叠加测试记录

测试组号	区县个数	地类图斑个数	监测图斑个数	空间叠加耗时(s)	
				GRID	RHASH
1	1	2787	51	29.32	41.05
2	3	7149	174	55.26	90.09
3	6	7338	256	61.05	108.27
4	12	152428	4402	81.47	143.58
5	18	204676	6714	85.51	176.67

可见,GRID优化方法空间叠加的耗时在各测试组均显著低于RHASH方法,且随着数据规模的增加,其效率优势进一步显现,测试要素数据量区间最大侧的

执行效率提升了1倍。这主要是由于未优化方案不能根据叠加数据从HBase表中快速过滤与其相关的信息,需进行全表扫描,并判断每条数据与叠加数据的空间关系。本文提出的优化策略弥补了这一不足,大幅降低了MapReduce处理的数据量,从而有效提升了计算性能。

总体来看,分布式存储与计算环境下,空间数据处理效率较关系型数据库大幅增加。在硬件资源要求并不苛刻的条件下,近600万地类图斑矢量要素的全流程叠加操作耗时仅20分钟,而关系型数据库下的同量级相似数据处理需投入大量作业人员使用不同客户端执行操作,时间单位往往以工作日计。因此,在海量国土空间矢量数据管理与分析中,将分布式存储、计算技术与空间数据特性充分优化结合,能够大幅提升工作效率。

5 结语

本文对空间矢量数据的分布式存储与计算技术进行了探索,研究了基于四叉树格网构建空间索引的方法,结合矢量数据的特性,将其与HBase预分区策略和分布式存储设计有效融合。同时,基于Hadoop处理模型,设计了与之衔接的分布式计算流程,形成了一种分布式场景下矢量数据组织与计算的优化方法。后续工作中,拟扩大集群和数据规模,全面调试并进一步提升其实用性。

参考文献

1 陈崇成,林剑峰,吴小竹,等.基于NoSQL的海量空间数据

云存储与服务方法.地球信息科学学报,2013,15(2):166-174.

2 李德仁.展望大数据时代的地球空间信息学.测绘学报,2016,45(4):379-384.

3 李斌,郭景维,彭睿.面向大数据存储的HBase二级索引设计.计算技术与自动化,2019,38(2):124-129.

4 魏玲,魏永江,高长元.基于Bigtable与MapReduce的Apriori算法改进.计算机科学,2015,42(10):208-210,243.

5 周经纬.矢量大数据高性能计算模型及关键技术研究[博士学位论文].杭州:浙江大学,2016.

6 王凯,曹建成,王乃生,等.Hadoop支持下的地理信息大数据处理技术初探.测绘通报,2015,(10):114-117.

7 范建永,龙明,熊伟.基于HBase的矢量空间数据分布式存储研究.地理与地理信息科学,2012,28(5):39-42.

8 李振举,李学军,谢剑薇,等.基于HBase的海量地形数据存储.计算机应用,2015,35(7):1849-1853.

9 朱进,胡斌,邵华,等.基于内存数据库Redis的轻量级矢量地理数据组织.地球信息科学学报,2014,16(2):165-172.

10 祝琳莹,张丰,杜震洪,等.基于HBase与静态多级格网索引的地表覆盖数据高效检索方法.浙江大学学报(理学版),2018,45(5):595-604.

11 郑美霞,王彦兵,马翔旭.基于四叉树索引构建TIN的高效合成算法.地理与地理信息科学,2012,28(2):20-23.

12 陈俊欣.基于Hadoop的空间矢量数据的分布式存储与查询研究[硕士学位论文].成都:电子科技大学,2016.

13 付仲良,胡玉龙,翁宝凤,等.M-Quadtree索引:一种基于改进四叉树编码方法的云存储环境下空间索引方法.测绘学报,2016,45(11):1342-1351.