

基于HBase的海量GIS数据分布式处理实践

李雪梅¹, 邢俊峰¹, 刘大伟¹, 王海洋^{1,2}, 刘玮^{1,2}

1.烟台中科网络技术研究所, 山东 烟台 264003; 2.中国科学院计算技术研究所, 北京 100080

摘要

设计了一种基于分布式数据库HBase的GIS数据管理系统。系统优化了栅格数据的生成和存储过程,将海量栅格数据直接写入HBase存储、索引。同时,针对矢量空间数据的存储、索引与检索,提出了一种新的rowkey设计,既考虑经纬度,又考虑空间数据类型和属性,使得在按空间位置检索矢量地理信息时,能通过HBase的rowkey迅速定位需要返回的数据。在HBase的集群环境上用真实GIS数据对上述方法进行了验证,结果表明,提出的系统具有较高的海量数据存储和检索性能,实现了海量地理信息数据的高效存储和实时高速检索。

关键词

大数据; HBase; 栅格数据; 矢量数据; rowkey

中图分类号: TP301

文献标识码: A

doi: 10.11959/j.issn.2096-0271.2016032

Distributed processing practice of the massive GIS data based on HBase

LI Xuemei¹, XING Junfeng¹, LIU Dawei¹, WANG Haiyang^{1,2}, LIU Wei^{1,2}

1. Institute of Network Technology, ICT(YANTAI), Yantai 264003, China

2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China

Abstract

Based on the distributed database HBase, a kind of GIS data management system was designed. The system optimized the generated and stored procedures of raster data, which could be directly written into the storage and indexing of the HBase. At the same time, in view of the storing, indexing and retrieval of the vector spatial data, a new design for rowkey was proposed that considering both the latitude and longitude, and the spatial data types and attributes. So that the data needed to be returned could be quickly located by rowkey of the HBase, when retrieving vector geographic information according to the spatial location. The above methods had been verified on the HBase cluster environment with real GIS data. The results show that the proposed system has high performance for storage and retrieval of mass data, and realizes the efficient storage and real-time high-speed retrieval of the vast geographic information data.

Key words

big data, HBase, raster data, vector data, rowkey

1 引言

在全球大数据蓬勃发展的大背景下,我国也紧抓发展机遇。2015年中国大数据技术大会发布了《2016年大数据技术发展趋势》,指出:可视化推动大数据平民化、多学科融合与数据科学的兴起、大数据提升社会治理和民生领域应用等将成为未来大数据的发展趋势^[1]。本文顺应大数据发展趋势,在大数据平民化、大数据民生应用等方面进行研究。

传统地理数据处理与存储主要使用Oracle与PostgreSQL的扩展PostGIS等关系型数据库,但关系型数据库在针对海量数据的管理、高并发读写、难扩展、对非结构化数据处理等方面已经开始制约地理信息系统的发展。而HBase采用的基于列的存储技术以及高可用、高可靠、高性能、可扩展的特点在处理非结构化地理栅格数据和具有离散稀疏特性地理矢量数据时有着天然的优势。本文已经成功地将栅格数据与地理数据中的矢量数据加载到HBase,并进行了检索测试,其检索响应时间可达到毫秒级。

本文的创新点在于当生成栅格数据时在mapnik切片的基础上做了优化,使栅格数据不经过磁盘直接写入HBase,并设计了栅格数据在HBase中存储的rowkey以及GIS(geographic information system,地理信息系统)矢量空间数据在HBase中存储的rowkey。

2 相关技术简介

2.1 GIS数据

GIS数据包括两种最基本的组织方式:栅格数据和矢量空间数据。

栅格数据是以二维矩阵的形式表示空间地物或现象分布的数据组织方式,每个矩阵单位称为一个栅格单元(cell),在各个栅格单元上给出相应的属性值来表示地理实体。栅格数据有属性明显、定位隐含的特点。随着数据压缩技术和计算机性能的提高,栅格数据量大的缺点不再突出,栅格数据在地图分析中的作用越来越明显。栅格数据可以是数字航空照片、卫星影像、数据图片,甚至可以是扫描的地图。本文所提到的栅格数据即对开放街道地图(OpenStreetMap,OSM)进行切片获得的栅格数据。

矢量空间数据利用点、线、面及其组合体的形式表达现实世界,具有定位明显、属性隐含的特点。同时,矢量空间数据具有数据结构紧凑、冗余度低、表达精度高、图形显示质量好、有利于索引和检索分析等优点,在GIS中得到广泛的应用,特别在小区域(大比例尺)制图中充分体现了其精度高的优点。本文所提到的矢量数据是指由政府提供的路网数据、管网数据、农村地块数据及部分城市的天地图矢量空间数据。

2.2 HBase

HBase是一个基于分布式的、面向列的、可伸缩的、主要用于非结构化数据存储用途的开源数据库。HBase是Google Bigtable^[2]的开源实现。其中,Hadoop^[3] HDFS为HBase提供文件存储功能;

MapReduce为其提供海量数据计算能力；ZooKeeper为其提供协调服务；Pig和Hive为其提供高层语言支持，使其可以进行数据统计；Sqoop为其提供RDBMS数据导入功能。ETL工具也能为其提供高速的数据导入/导出功能。

HBase支持按照主键rowkey和主键的range查询，可以通过HBase提供的API（application programming interface，应用程序编程接口）进行条件过滤查询，也可以通过编写MapReduce程序，实现表数据的访问和分析。HBase有如下特点：

- 对于数据类型，HBase可以存储动态、灵活的数据模型，并不限制存储数据的种类，可以存储结构化、半结构化和非结构化数据；
- HBase不使用SQL语言，也不强调数据之间的关系；
- 存储灵活，可以在一行的某一列存储一个整数，而在另一行的同一列存储字符串，对于空的列不占用存储空间，适合存储稀疏数据；
- 在集群上运行，可存储海量数据。

3 GIS数据在HBase中的应用

传统的GIS数据存储大多是关系型数据库，但关系型数据库在海量数据的管理中面临许多问题，包括高并发读写、难扩展等，已经成为制约GIS发展的瓶颈。同时，GIS数据中的图片、影像数据等都是非结构化的，关系型数据库不能合理地处理非结构化数据。而HBase以其高可靠性、高扩展性、高容错性、高效性以及适用于海量非结构化数据存储处理分析的优势在处理GIS数据方面提供了另一种思路，即解决HBase应对GIS中的两大挑战：大规模数据处理的时延和空间位置建模^[4]。

3.1 栅格数据

3.1.1 栅格数据获取

本文栅格数据获取、写入流程如图1所示。

传统的栅格数据的存储方式主要有两种：以文件目录的形式存储和存储在传统数据库中。

若栅格数据以文件目录形式存储，其目录结构为zoom_level/x_coordinate/y_coordinate.png。图2展示的是对北京市切片获得的第9层栅格数据。

如果栅格数据以文件目录形式存储在本地磁盘，对大范围地图切片会产生海量

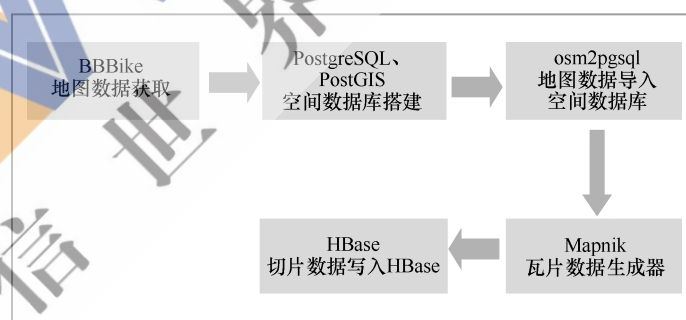


图1 栅格数据获取流程

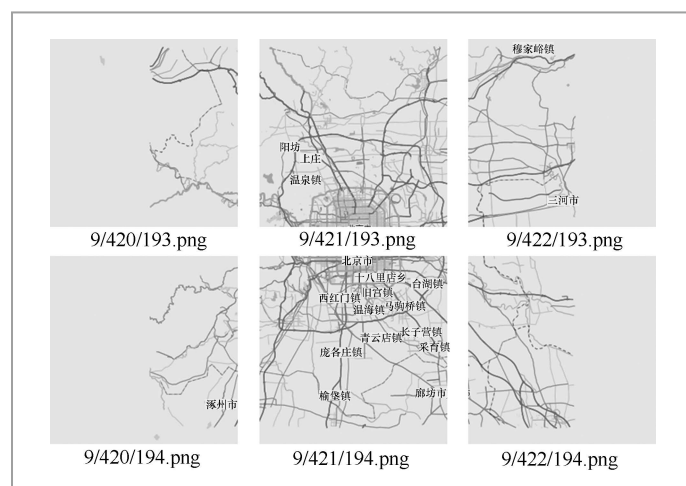


图2 栅格数据存储形式和目录结构

栅格数据。在生成栅格数据时速率逐步降低,影响切片性能。本文对切片脚本做了修改,略去文件写磁盘步骤,将栅格数据文件进行base64转码后直接存至HBase。

如果海量栅格数据存储在传统数据库(如MySQL、Oracle等),主键和索引问题会导致读写速度越来越慢,并且存储量有所限制。

基于以上原因,结合HBase在大数据存储方面的优势,本文研究

了栅格数据在HBase上的存储和索引,在HBase的集群环境上对栅格数据进行了验证,该方法具有较高的存储能力和检索能力。

3.1.2 栅格数据的索引

如上所述,层级编码、X坐标与Y坐标表示的不仅是一个瓦片数据的索引,同时还是瓦片数据的地理位置标志,通过数学公式在查询和检索时,行列编码和经纬度之间可以实现相互转换。本文中栅格数据存储于HBase, rowkey的表示方式为zoom_level、x_coordinate、y_coordinate,列簇(column family)为i,列(column)为png_info。栅格数据在系统中的展示形式如图3、图4所示。

3.1.3 栅格数据的检索

栅格数据的检索主要是返回指定区域的所有栅格信息,以编排好的一张地图的形式返回。栅格数据的索引方式使得需要先确定X、Y坐标的范围,再在返回的结果上进行筛选,编排出符合条件的所有切片合成一张地图。本文栅格数据的检索是以服务形式提供,若检索第9层,X坐标为421,Y坐标为194的栅格数据如图5所示。

在栅格数据服务上,通过OpenLayers可将栅格数据以地图的形式返回。图6展示的是切片数据第12层拼接的地图。

3.2 矢量空间数据

矢量空间数据的组织形式比栅格数据复杂得多,不仅有点(point)、线(line string)、面(polygon)、多点(multi-point)、多线(multi-line string)、多面(multi-polygon)等不同的矢量空间数据格式以及图层和复杂的拓扑关系信息^[5,6]。为了使用HBase方便、高效地管理矢量空



图3 栅格数据存储方式

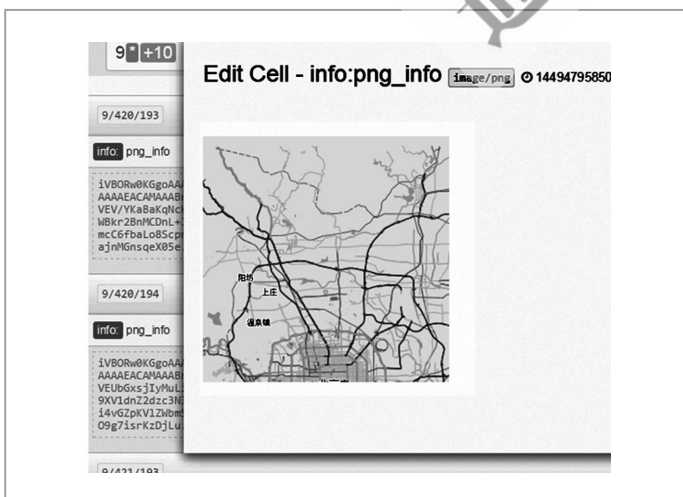


图4 栅格数据展示

间数据,并且使地理位置相邻的空间对象尽可能地在逻辑存储上也相邻,需要设计出合理的rowkey来提高检索效率。

3.2.1 矢量空间数据的索引

无论是点、线、面等何种矢量空间数据格式,都是由一个个二维(经度和纬度)的点组成,其中,点的表示为 (x, y) ;线的表示为 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$;面的表示为 $(x_1, y_1), (x_2, y_2), \dots, (x_1, y_1)$ 。不同于一维数据可以通过排序方式创建索引,在检索时可考虑二分查找法进行快速查找。在对矢量空间数据进行存储时主要考虑如何将二维数据转换成一维数据,进而作为HBase的rowkey索引与检索。

二维或多维空间坐标降维至一维字符串的方法有很多,如Hilbert空间降维法^[7-9],该方法能够通过数学模型最大限度地保证空间对象之间的逻辑相关性,无较大突变;

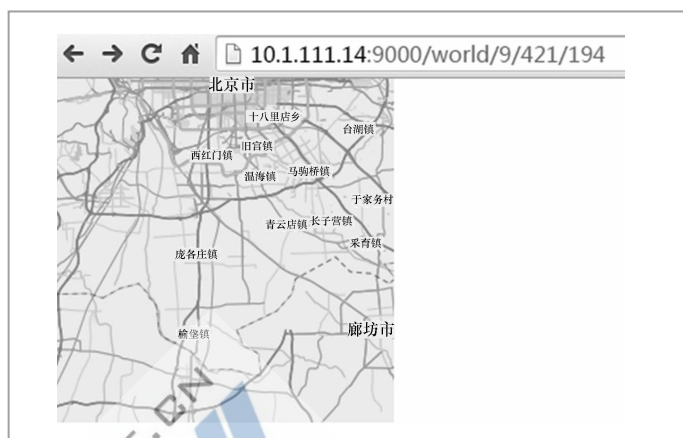


图5 切片服务

缺点是数学模型较复杂,降维过程中运算量较大。四叉树^[10]也是一种有效的空间降维方法,四叉树其实是一种字典树,因为树节点的值不依赖于插入的数据,在进行数据降维时需要先建好树,再将需要降维的数据划分到适当的区间即可,但算法相对复杂。本文使用Geohash^①方法降维。

① <http://en.wikipedia.org/eohash>



图6 栅格数据拼接地图

Geohash即将一个经纬度信息转换成一个既可以排序又可以比较的字符串编码。具体操作步骤如下。

步骤1 将纬度范围(-90, 90)平分成两个区间(-90, 0)、(0, 90), 如果目标纬度位于前一个区间, 则编码为0, 否则编码为1。

步骤2 若步骤1编码为0, 将纬度范围(-90, 0)平分成两个区间(-90, -45)、(-45, 0), 如果目标纬度位于前一个区间, 则编码为0, 否则编码为1。

步骤3 若步骤1编码为1, 同步步骤2, 将纬度范围(0, 90)平均分区。

步骤4 依据步骤2和步骤3对纬度进行划分, 直到精度符合要求为止, 得到纬度编码。

步骤5 经度编码同纬度算法, 对(-180, 180)依次细分, 直到精度符合要求为止, 得到经度编码。

步骤6 将经度和纬度的编码合并, 奇数位是纬度, 偶数位是精度。

步骤7 对步骤6合并的编码进行base32编码。

为了使得Geohash具有可比性^②, 在进行base32编码时需参考字符在ASCII码表的顺序。用0~9、b~z(去掉a, i, l, o)32个字符进行base32编码, 具体的码表见表1。

例如, 纬度为39.923 24°的编码是1011 1000 1100 0111 1001, 经度为116.390 6°的编码为1101 0010 1100 0100 0100, 经纬度合并编码为11100 11101 00100 01111 00000 01101 01011 00001, base32转码后编码为wx4g0ec1。资料显示base32编码长度为8时, 精度在19 m左右, 具体的编码长度需要根据数据情况进行选择, 为了既可以定位到某个位

置, 又保护隐私性, 并且考虑到HBase的rowkey选择遵循越短越好的原则, 本文规定base32的长度最长为8。

对经纬度进行Geohash降维后, 不仅大大缩短了rowkey的长度, 对大部分而言, 编码相似的距离也相近, 见表2实际编码实例中的前两行。但该算法的缺点是有数据突变, 有些Geohash值相差很远, 但实际位置很近, 如图7中的WX4G08和WX4FBX。解决这一问题可使用扩大返回结果的方式, 如在查询时, 除了返回与Geohash值匹配的区域, 同时返回其周围的8个区域。

针对在HBase上的多条件查询, 一般是将所有可能作为查询条件的字段一一拼接到rowkey中, 这种拼接的rowkey使得在查询时排序越靠前的字段越有优势。为了提高空间查询的效率, 本文除了考虑经纬度信息, 又要考虑空间数据组织形式和图层属性, 为了保证线和其经过的区域以及面和其包含的区域表达出来, 需要在rowkey上拼接网格ID。本文的rowkey拼接顺序为: 表名; 空间数据类型; 经纬度Geohash编码; 网格ID, 如: line;L;wx51053b;102203822。列的设计即将源数据表中所有的列信息一一写入HBase, 为了加快检索速率, 添加一列图层属性信息。

空间数据类型可在源数据中获取, 在第3.2节介绍了空间数据组织形式有6种, 为了遵循rowkey的越短越好原则^②, 将组织形式进行了映射, 见表2。

若将线或面上的标记点及整条线或整个面的信息一一写入HBase, 当要展示某个区域的地理信息时, 若一条线穿过这个

② <http://hbase.apache.org/0.94/book/rowkey.design.html#keysize>

表1 base32 码表

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 十进制 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| base32 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | b | c | d | e | f | g | h | j | k | m | n | p | q | r | s | t | u | v | w | x | y | z |

区域,但线上的标记点都没落在该区域,那么这条线就不能正常地显示出来,如图8所示。同理,若一个面的边界在这个区域之外,面上的标记点也都没落在该区域,也会出现面不能正常显示的问题,如图9所示。网格ID的设计是为了解决该类问题。

对于网格ID设计的步骤如下。

步骤1 获取数据库对象的边界。

步骤2 对步骤1的边界进行四叉树划分,直至预先设定的地理位置精度。本文规定划分后的每个单元格使用6位base32表示即可。

步骤3 将每个单元格的经纬度信息转换成Google的900913类型几何值。

步骤4 在将GIS数据写入HBase前与单元格几何值进行相交,得到每条记录与网格相交的信息。

步骤5 对于线数据表或面数据表中的每一条记录,除了本身的信息(标记点及整条线或整个面的信息)之外,相交信息也需一同写入HBase相应数据库。

3.2.2 矢量空间数据的检索

对矢量空间数据的检索主要有以下3种。

- 返回指定区域的点、线、面等信息,该种情况需要遍历rowkey的前缀为:表名;空间数据类型的各种情况,返回base32值小于或等于指定区域的所有数据。

- 返回整条线或整个面的信息,该种情况只需指定rowkey的前缀,返回整条线或整个面的边界信息。

- 返回指定图层的信息,该种情况若直接通过检索rowkey的方式获取需要遍历的整个HBase表,为了加快该种情况的检索速度,本文在列的设计方面增加了一项fliter,用来存储从矢量信息中获取的图层信息。检索时只需返回指定列信息即可。

表2 空间数据组织形式映射

| 空间组织形式 | 映射 |
|-----------------|-----|
| POINT | P |
| MULTIPOINT | MP |
| LINSTRNG | L |
| MULTILINESTRING | ML |
| POLYGON | PY |
| MULTIPOLYGON | MPY |

| | | |
|--------|--------|--------|
| WX4G09 | WX4G0C | WX4G11 |
| WX4G08 | WX4G0B | WX4G10 |
| WX4FBX | WX4FBZ | WX4FCP |

图7 编码实例

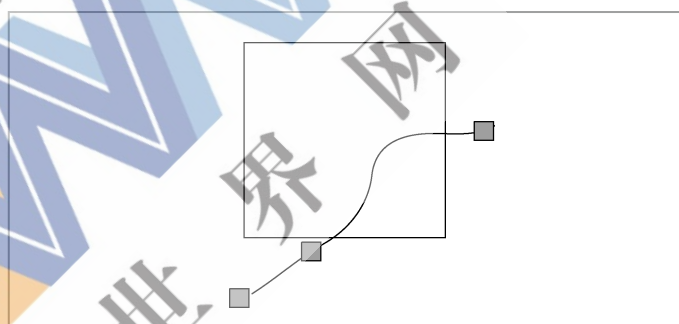


图8 线穿过区域

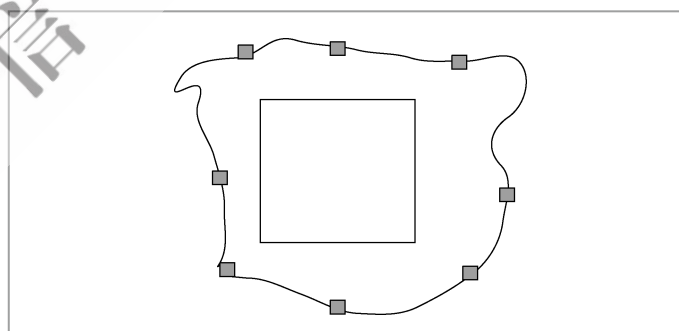


图9 面包含区域

4 实验与结果分析

4.1 实验环境

本文的实验环境为3台服务器搭建的集群环境。服务器的相关配置见表3。

表3 服务器配置

| 名称 | 配置 |
|------|----------------------------|
| 操作系统 | CentOS release 6.4 (final) |
| CPU | x86_64, 6核心 |
| 内存 | 16 GB |
| 硬盘 | 999.1 GB |

4.2 栅格数据写HBase

本文在栅格数据切片及写HBase上主要是略去了磁盘写步骤,默认的栅格数据获取是以文件形式存储在本地磁盘,再通过读磁盘文件写入HBase。对北京市分别切片至16层、18层,并将栅格数据写入HBase测试结果,见表4。

从表4可以看出,改进后略去磁盘写操作,在栅格数据量较少时性能反而下降。但在处理数据规模较大(如对中国地图或世界地图进行切片)时,会因持续写磁盘操作,生成文件夹及切片文件数太多导致inode(索引节点)耗尽。经过约24 h的持续切片,生成海量切片小文件,导致服务器inode用尽,即使硬盘空间仍有空余,也无法在硬盘上创建新文件,进而切片工作无法正常完成。

若将切片直接写入传统数据库(如MySQL、Oracle、PostgreSQL等),写入速度不断降低,导致写入时间不可接受,后期的检索时间也较慢。另外,海量切片小文件写入传统数据库会对数据库造成较高的负载,系统的稳定性和扩展性难以控制。

经过改进测试,对世界地图连续切片

一个月,形成83 GB的栅格数据,未经过磁盘写操作直接写入HBase,未出现任何异常情况。

本文对栅格数据的检索也进行了验证,在124 217 416条记录的HBase表上检索指定切片的响应时间为毫秒级(0.004 s)。同时,HBase也支持指定范围检索,一次返回万级别切片的响应时间为秒级(5.077 s)。

4.3 矢量空间数据写HBase

为了对海量矢量空间数据进行高效存储、管理与检索,本文设计了矢量空间数据的rowkey,并将数据存储于HBase。实验结果显示,对中国地图的矢量空间数据在24 201 991条记录的Hbase表上检索指定空间矢量点的响应时间为毫秒级(0.021 s)。检索线或面的响应时间为秒级,若返回一个面上的千级别的坐标点时间约为3.551 s。

5 结束语

本文针对智慧城市建设中的数据管理问题,利用HBase分布式数据库的列存储模型特点,设计了一种基于HBase的GIS数据管理系统,实现了对矢量空间数据与栅格数据的高效存储、索引与检索。GIS矢量空间数据在HBase中存储,首先确定rowkey的设计,使得在检索空间位置时返回尽可能少的数据请求,即在检索时既考

表4 栅格数据写HBase

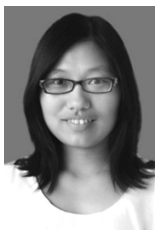
| 切片层数 | 切片数/张 | 切片+写HBase | 耗时/s |
|------|-----------|-----------|-------|
| 16层 | 84 576 | 磁盘写 | 182 |
| | | 略去磁盘写 | 224 |
| 18层 | 1 340 384 | 磁盘写 | 2 437 |
| | | 略去磁盘写 | 3 810 |

考虑经度与纬度,又考虑空间数据类型和图层属性。设计了基于表名、空间数据类型、经纬度Geohash编码、网格ID的rowkey方法。为了加快切片效率和减少存储成本,本文在生成栅格数据时在mapnik切片的基础上做了优化,使栅格数据不经过磁盘写操作,直接写入HBase。设计了栅格数据在HBase中存储的rowkey。最后,在HBase的集群环境上对上述方法进行了验证。实验表明,本文提出的方法具有较好的可行性和执行效率。

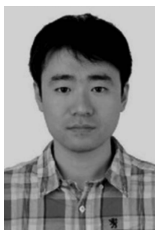
参考文献:

- [1] 潘柱延,程学旗,袁晓如,等. CCF大专委2016年大数据发展趋势预测——解读和行动建议[J]. 大数据, 2016, 2(1): 105-113.
PAN Z Y, CHENG X Q, YUAN X R, et al. Developing trend forecasting of big data in 2016 from CCF TFBD: interpretation and proposals [J]. Big Data Research, 2016, 2(1): 105-113.
- [2] CHANG F, DEAN J, GHEMAWA S, et al. Bigtable: a distributed storage system for structured data[C]//The 7th USENIX Symposium on Operating Systems Design and Implementation, November 6-8, 2006, Seattle, WA, USA. New York: ACM Press, 2006: 205-218.
- [3] 陈嘉恒. Hadoop实战[M]. 北京: 机械工业出版社, 2011: 1-2, 260-261.
CHEN J H. Hadoop in action[M]. Beijing: China Machine Press, 2011: 1-2, 260-261.
- [4] 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[J]. 地球信息科学学报, 2013, 15(2): 166-174.
CHEN C C, LIN J F, WU X Z, et al. Massive geo-spatial data cloud storage and services based on NoSQL database technique[J]. Journal of Geo-Information Science, 2013, 15(2): 166-174.
- [5] 范建永,龙明,熊伟. 基于HBase的矢量空间数据分布式存储研究[J]. 地理信息与地理信息科学, 2012, 28(5): 39-42.
FAN J L, LONG M, XIONG W. Research of vector spatial data distributed storage based on HBase[J]. Geography and Geo-Information Science, 2012, 28(5): 39-42.
- [6] 张叶,许国艳,花青. 基于HBase的矢量空间数据存储与访问优化[J]. 计算机应用, 2015, 35(11): 3102-3105.
ZHANG Y, XU G Y, HUA Q. Storage and access optimization of vector space data based on HBase[J]. Journal of Computer Applications, 2015, 35(11): 3102-3105.
- [7] WANG L, CHEN B, LIU Y. Distributed storage and index of vector spatial data based on HBase[C]//The 2013 21st International Conference on Geoinformatics, June 20-22, 2013, Kaifeng, China. New Jersey: IEEE Press, 2013:1-5.
- [8] 陆锋,周成虎. 一种基于Hilbert排列编码的GIS空间索引方法[J]. 计算机辅助设计与图形学学报, 2001, 13(5): 424-429.
LU F, ZHOU C H. A GIS spatial indexing approach based on Hilbert ordering code[J]. Journal of Computer-Aided Design and Computer Graphics, 2001, 13(5): 424-429.
- [9] SHEKHAR S, CHAWLA S. 空间数据库[M]. 谢昆青,译. 北京: 机械工业出版社, 2004: 108-113.
SHEKHAR S, CHAWLA S. Spatial database[M]. Translated by XIE K Q. Beijing: China Machine Press, 2004: 108-113.
- [10] 滕连泽,刘洪斌,王伟. 基于空间数据不同索引方法的比较[J]. 计算机与现代化, 2006(1): 60-62.
TENG L Z, LIU H B, WU W. Comparison of indexes based on spatial data[J]. Computer and Modernization, 2006(1): 60-62.
- [11] HSU Y T, PAN Y C, WEI L, et al. Key formulation schemes for spatial index in cloud data managements[C]// The 2012 IEEE 13th International Conference on Mobile Data Management, July 23-26, 2012, Bengaluru, India. New Jersey: IEEE Press, 2012: 21-26.

作者简介



李雪梅 (1984-), 女, 烟台中科网络技术研究所研发工程师, 主要研究方向为数据挖掘与分析、自然语言处理、大数据存储管理与挖掘等。



邢俊峰 (1985-), 男, 烟台中科网络技术研究所研发工程师, 主要研究方向为数据存储检索和展现等。



刘大伟 (1984-), 男, 博士, 烟台中科网络技术研究所助理研究员, 主要研究方向为海量数据检索、多媒体内容分析、社交网络分析等。



王海洋 (1980-), 男, 中国科学院计算技术研究所网络数据科学与技术重点实验室、烟台中科网络技术研究所工程师, 主要研究方向为数据采集、存储、检索、大数据存储管理等。



刘玮 (1977-), 男, 中国科学院计算技术研究所网络数据中心、烟台中科网络技术研究所工程师。历任中国科学院计算技术研究所项目负责人, 北京中科天玑信息技术有限公司总经理助理, 北京中科智源育成信息技术有限公司事业部经理。以项目负责人或核心骨干成员承担和参与了国家信息关防与网络安全保障持续发展计划、国家信息安全专项、国家242信息安全计划、发展和改革委员会产业化项目等10余项国家重大项目的组织实施。

收稿日期: 2016-03-07

基金项目: 山东省自主创新及成果转化专项基金资助项目 (No. 2014XGA06014); 山东省科技发展计划基金资助项目 (No. 2015GGX106001); 烟台市科技发展计划基金资助项目 (No. 2014LGS005)

Foundation Items: Technology Innovation Program of Shandong Province of China (No. 2014XGA06014), Shandong Technology Research and Development Program (No. 2015GGX106001), Yantai Technology Research and Development Program (No. 2014LGS005)