

# 基于 HBase 的矢量空间数据分布式存储研究

范建永,龙明,熊伟

(信息工程大学测绘学院,河南 郑州 450052)

**摘要:**分析了分布式数据库 HBase 的存储模型;结合对 HBase 集群技术的研究,设计了基于 HBase 的矢量空间数据存储模型和一种基于 MapReduce 的并行构建网格空间索引方法,使得海量空间矢量数据的网格索引构建分配到各子节点进行,大大加快索引构建的处理速度;最后,利用 HBase 集群环境对所提出的方法进行验证,该方法具有较好的可行性和较高的效率。

**关键词:**Hadoop;HBase;矢量数据;存储模型;分布式存储;MapReduce

**中图分类号:**TP301   **文献标识码:**A   **文章编号:**1672-0504(2012)05-0039-04

面对海量和复杂的空间矢量数据,如何对其进行高效存储、管理与发布,已成为一个迫切需要解决的问题。云计算是一种新的分布式计算架构,具有大规模扩展、水平分布的特性,可以提供无限的存储能力和计算能力。将云计算的分布式存储应用于 GIS 领域是解决海量空间数据管理问题的有效手段。

Hadoop<sup>[1]</sup>是 Apache 软件基金会旗下的一个开源分布式计算平台,其核心是分布式文件系统 HDFS、并行计算模型 MapReduce 和分布式数据库 HBase,它们分别是 Google 云计算最核心技术 GFS<sup>[2]</sup>、MapReduce<sup>[3]</sup>和 Bigtable<sup>[4]</sup>的开源实现。用户可以利用 Hadoop 轻松地组织计算机资源,从而搭建分布式计算平台,并且可以充分利用集群的计算和存储能力,完成海量数据的处理<sup>[5]</sup>。目前使用 Hadoop 技术实现云计算平台的有 IBM 的蓝云、Yahoo!、微软的云计划、eBay、Facebook,还有中国移动的 BigCloud、百度云计算以及阿里巴巴云计算平台等。本文主要研究基于 HBase 的矢量空间数据存储与处理,为海量矢量空间数据的高效存储和处理提供新的解决方案和思路。

## 1 HBase 数据模型

HBase 是 Apache Hadoop 一个顶级项目,是一个支持结构化和非结构化数据存储的分布式数据库,是 Bigtable 的开源实现,能够对大型数据提供实时的读写访问。HBase 与 Bigtable 类似,是一个稀疏的、长期存储的、多维度的、排序的映射表,其表索引由行关键字、列关键字和时间戳组成,其值都是字符串类型。

### 1.1 HBase 逻辑模型

HBase 以表的形式存储数据,每个表由行和列组成,每个列属于一个特定的列族(Column Family),行和列确定的存储单元为一个元素(Cell),每个元素保存同一份数据的多个版本,通过时间戳(Time Stamp)标示。由于 HBase 是稀疏存储数据,所以某些列可以为空,如语句 create 'test', 'c1', 'c2' 创建表。表 1 是一个表的逻辑数据模型,表中每一条数据对应一个时间戳,用数字表示,编号越大表示数据越旧。

表 1 HBase 逻辑数据模型<sup>[5]</sup>  
Table 1 Logic data model of HBase

Row Key	Time Stamp	Column Family:c1		Column Family:c2	
		列	值	列	值
r1	t7	c1:1	value1-1/1		
	t6	c1:2	value1-1/2		
	t5	c1:3	value1-1/3		
	t4			c2:1	value1-2/1
r2	t3			c2:2	value1-2/2
	t2	c1:1	value2-1/1		
	t1			c2:1	value2-1/1

### 1.2 物理模型

HBase 是按照列存储的稀疏行/列矩阵,在物理存储方面,它实际就是把概念模型中的一行分割,并按照列族存储。表 1 中 r1 行在物理存储时表现如表 2 所示。

表 2 HBase 数据的物理视图  
Table 2 Physical data model of HBase

Row Key	Time Stamp	Column Family:c1	
		列	值
r1	t7	c1:1	value1-1/1
	t6	c1:2	value1-1/2
	t5	c1:3	value1-1/3
	t4	c2:1	value1-2/1
	t3	c2:2	value1-2/2



传统建立空间索引通常是以空间换取时间的串行方式,效率较低。本文利用 MapReduce 编程模型并行构建矢量数据空间索引(图 2),可有效缩短索引建立的时间。如图 2 所示,MapReduce 算法分为 Map 阶段和 Reduce 阶段。Map 函数读取一条空间要素记录,计算要素所在的所有网格索引 ID,并以索引 ID 为 Key、以空间要素的 ID 为 Value 写入本地 HDFS;Reduce 阶段把从不同 Mapper 计算的索引 Key-Value 数据整合并进行排序,然后调用 Reduce 函数对输入的<索引 ID, list(Fea\_ID)>对进行合并处理,把得到的<索引 ID, Fea\_IDS>写入相应比例尺、相应图层的 HBase 索引表中。

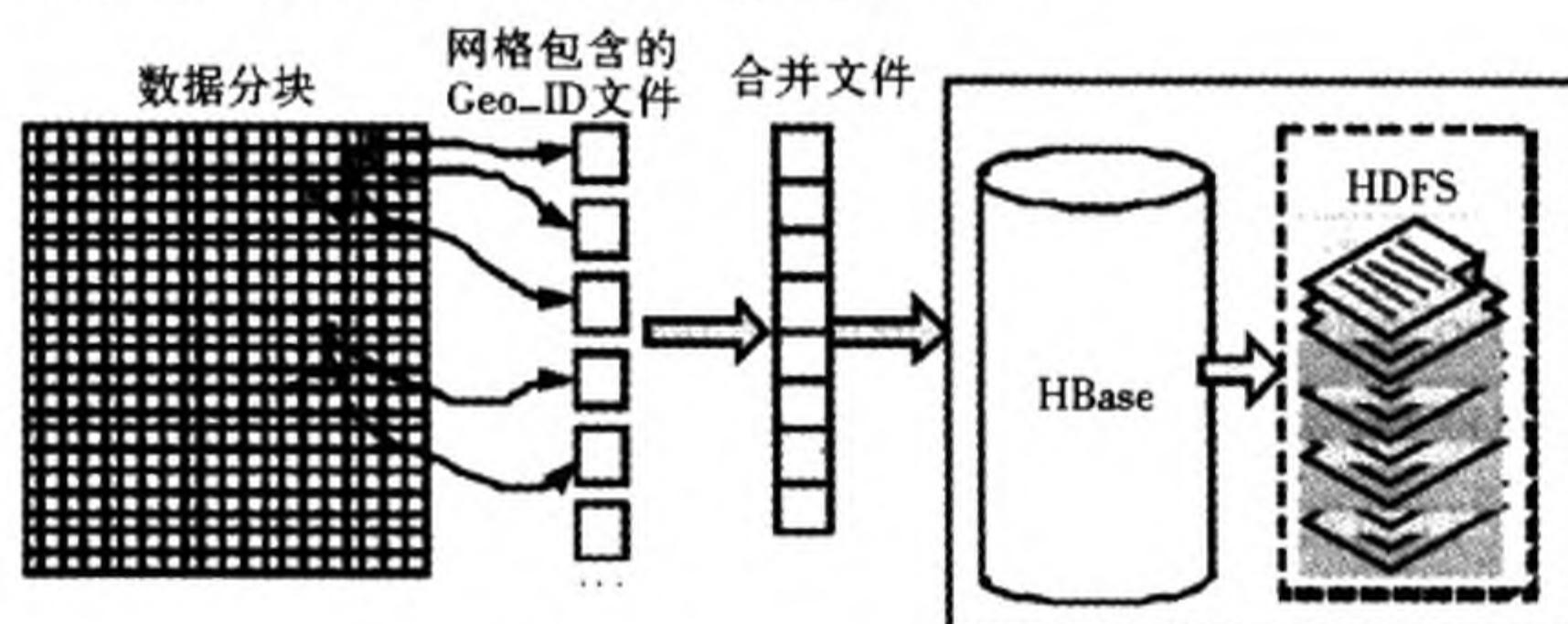


图 2 基于 MapReduce 的空间索引建立  
Fig. 2 Spatial index building base on MapReduce

利用 MapReduce 最大的优点是将空间索引计算扩展到多个计算机节点上处理,相比串行计算方式可有效减少内存占用量,并提高计算效率,缩短计算时间。

### 3 试验设计与结果分析

#### 3.1 试验环境及数据

利用 3 台计算机作为宿主机,每台均安装 VMware 虚拟机,1 台宿主机虚拟出 3 台计算机,另外 2 台各虚拟出 4 台计算机;1 台虚拟机作为 Web 服务器,1 台部署 HDFS、MapReduce 和 HBase 作为 MasterNode,6 台部署 HDFS、MapReduce 和 HBase,作为 SlaveNode,另外 3 台部署 Zookeeper。其配置如图 3 所示。宿主机:CPU 为 Duo T7700,内存 4 GB,操作系统 Windows XP;虚拟机:操作系统均为 Ubuntu

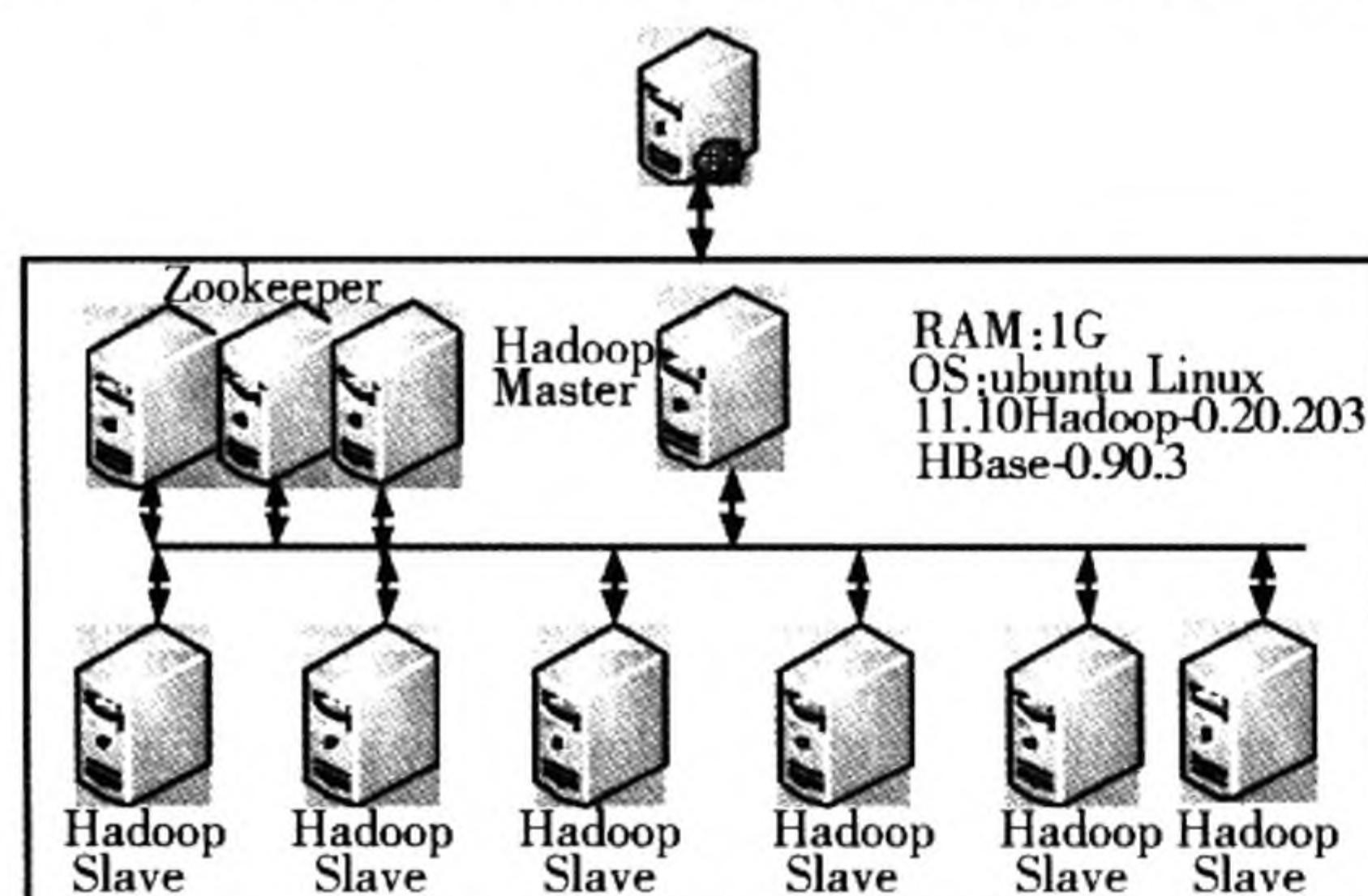


图 3 试验环境配置  
Fig. 3 Test environment configuration diagram

Linux10.1,内存 1 GB, Hadoop-0.20.203, HBase-0.90.3,开发环境是 Eclipse3.6。试验数据为 1:25 万矢量数据(538 MB)、1:5 万矢量数据(2.38 GB),全部为 Shape 数据。

#### 3.2 矢量数据入库

本文矢量数据入库采用客户端入库的方式,利用 GeoTools-2.7.4<sup>[8]</sup>开源项目读取客户端本地的 shp 数据,将数据导入相应的表中。入库时间包括数据读取和写入时间两部分。在 7 个节点组成的集群环境中利用 put()方法将浙江省的 1:5 万等高线层数据(1.2 GB, 434 527 条记录)、交通层数据(164 MB, 111 545 条记录)导入 HBase 数据库的时间分别是 3 283.641 s、119.718 s。

#### 3.3 矢量数据空间索引并行构建

本试验主要验证运用 MapReduce 构建矢量数据中线状要素网格索引的效率。对 3.2 节入库的等高线层和交通层数据利用 LineDDA 构建网格索引的方法分别在 5 个 Slave 节点、6 个 Slave 节点和 7 个 Slave 节点所组成的集群中并行构建索引所耗时间如表 6 所示。

表 6 矢量数据并行构建索引时间  
Table 6 Vector data parallel building index time

	5 节点	6 节点	7 节点
等高线层	2 255.327 s	1 752.546 s	1 703.054 s
交通层	217.386 s	209.325 s	198.540 s

由表 6 可知,随着节点数的增加,对于交通层数据创建索引的效率提高不大。主要是基于 HBase 根据不同的数据量(默认 256 M)划分 Region<sup>[9]</sup>,每一个 Region 由一个 RegionServer 管理。当数据量一定时,所分 Region 的数量相同,执行 Map 任务的数量相同。交通层实行默认分块,共有一块,执行一个 Map 任务,需要一个节点,其余节点闲置。等高线数据可划分 5 个 Region,需 5 个 Map 任务,在 4 个 RegionServer 的情况下只有 4 个节点可执行 4 个 Map 任务,1 个任务等待,这就是 5~6 个节点效率大幅提高、7 个节点效率提高不明显的原因。

#### 3.4 矢量数据查询试验

根据空间范围查询矢量数据的步骤如下:1)根据空间范围计算网格的 ID;2)从不同图层的索引表中读取网格 ID 所包含的 Fea\_ID;3)根据 Fea\_ID 读取不同图层的空间数据和属性数据;4)将查询的数据写入本地客户端。

查询时间包括计算网格 ID、读取 Fea\_ID、读取矢量空间数据和写入本地时间。表 7 列出了在 6 个节点环境下从 HBase 查询等高线层和交通层所需时间。

表 7 矢量数据查询时间  
Table 7 Vector data inquiry time

查询范围	查询数量	平均时间(s)
等高线层 (119.9,30.5 120.1,30.6)	1 079(6.2 M)	9.248
交通层 (119.9,30.5 120.1,30.6)	448(0.512 M)	1.185

通过以上试验可以看出,将 HBase 应用于海量空间矢量数据管理可以提高空间数据的存储和管理效率,并可以通过扩充节点获得较大的存储空间和较高的空间分析计算效率。

#### 4 结语

本文利用 HBase 分布式数据库的列存储模型特点,对空间数据的分布式存储机制进行研究,设计存储模型,建立索引表及索引存储格式,并提出了利用并行方法构建空间数据的步骤。最后,验证了本文模型的有效性和实用性。对于基于 HBase 的空间数据存储效率的提高和发布将是下一步研究的内容。

#### 参考文献:

- [1] Hadoop. <http://hadoop.apache.org/> [EB/OL]. 2011-10-31.
- [2] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google file system[A]. Proceeding of 19th ACM Symposium on Operating Systems Principles[C]. 2003. 20-43.
- [3] DEAN J, GHEMAWAT S. MapReduce: Simplified data processing on large clusters[A]. OSDI'04; 6th Symposium on Operating Systems Design and Implementation[C]. 2004. 137-150.
- [4] CHANG F, DEAN J, GHEMAWAT S, et al. Bigtable: A distributed storage system for structured data[A]. OSDI'06: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation[C]. 2006. 205-218.
- [5] 陆嘉恒. Hadoop 实战[M]. 北京:机械工业出版社,2011. 1-2, 260-261.
- [6] OGC. <http://www.opengis.org/techno/> [EB/OL]. 2009-12-31.
- [7] SHEKHAR S, CHAWLA S. 谢昆青(译). 空间数据库[M]. 北京:机械工业出版社,2004. 108-113.
- [8] GeoTools. <http://www.geotools.org/> [EB/OL]. 2012-01-31.
- [9] GEORGE L. HBase the Definitive Guide[M]. O'Reilly Media, 2011. 324-327.

### Research of Vector Spatial Data Distributed Storage Based on HBase

FAN Jian—yong, LONG Ming, XIONG Wei

(Institute of Surveying and Mapping, Information Engineering University, Zhengzhou 450052, China)

**Abstract:** To resolve the problem of massive vector spatial data's storage and query, this paper brings forward a vector spatial data distributed storage method based on HBase. At first, this paper analyzes the distributed storage model of HBase. And then, designs a distributed storage model of vector spatial data based on HBase, and proposes a parallel build index algorithm based on MapReduce. Finally, through experiments, this method is proved to be feasibility, reliability. This experiment shows that using HBase can effectively enhance the speed of massive vector spatial data processing. It provides a good solution for large-scale data-processing.

**Key words:** Hadoop; HBase; vector data; storage model; distributed storage; MapReduce

(上接第 38 页)

### Mapping Fire Occurrence Susceptibility in Northeast China: Comparison of Frequency Ratio and Binary Logistic Regression

ZHANG Hai-jun, QI Peng-cheng

(School of Environmental Science & Tourism, Nanyang Normal University, Nanyang 473061, China)

**Abstract:** Fire occurrence susceptibility mapping provides the information on the spatial distribution of fire danger and it is valuable to fire management. Take Northeast China as the study area, seven static fire-influencing factors involving topography, accessibility and vegetation cover are selected and multi-temporal random sampling methodology is employed to create the training subset, and then the training subset is utilized to develop the frequency ratio(FR) and binary logistic regression(BLR) models which are used to map the fire danger of Northeast China. The cumulative frequency curve of fire danger ranks is drawn and the area under ROC curve(AUC) procedure is employed to validate the reliability and the discrimination capacity of the developed FR and BLR models respectively. Evaluated by the inner testing and independent validation, good reliability and reasonable discrimination capacity of the two developed models can be concluded. When the difference of the reliability is compared between the FR model and the BLR model, it is six of one and half a dozen of the other. However, the discrimination capacity of the BLR model is more robust than that of the FR model. The study suggests that the static predictors employed play a structural role in forming spatial pattern of fire danger in Northeast China. In addition, those regions that are sensitive to fire events from 2000 to 2009 are illuminated.

**Key words:** fire danger mapping; fire danger assessment; logistic regression; frequency ratio; Northeast China