文章编号：2095-2163(2019)04-0226-04 中图分类号：TP751 文献标志码：A

基于 Spark 的遥感影像金字塔瓦片构建

陕唐剑，钟宏伟，李林辉，魏景琦，国露方
（东北林业大学 信息与计算机工程学院，哈尔滨 150040）

摘 要： 随着传感器技术发展和大容量存储设备价格的下降，遥感影像数据量飞速增长并达到海量存储的程度，传统的单机集中式存储和检索技术已经不能满足数据高效存储和快速访问的要求。本文在传统的影像金字塔的构建方法上，采用基于内存计算的计算引擎 Spark 来构建遥感影像金字塔并给出了算法的构建过程，实现了基于 Spark 的遥感影像金字塔的并行构建。实验表明，相对于传统的遥感影像金字塔构建，性能和计算效率都有较大提高。

关键词： 遥感影像；金字塔；Spark

Building of remote sensing images tile pyramid based on Spark

SHAN Tangjian, ZHONG Hongwei, LI Linhui, WEI Jingqi, GUO Lufang
(College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China)

【Abstract】 With the development of remote sensing means and information acquisition technology, the amount of remote sensing image data increases exponentially. The traditional single-machine centralized storage and retrieval technology has been difficult to meet the needs of efficient storage and fast access of data.On the traditional image pyramid construction method, the paper proposes a memory-based computing engine Spark to construct remote sensing image pyramid. The construction process of the algorithm and the distributed storage structure are given, and the parallel construction of remote sensing image pyramid based on Spark is realized. Experiments show that compared with the traditional pyramid construction of remote sensing image, the performance and computational efficiency are greatly improved.

【Key words】 remote sensing image; pyramid model; Spark

0 引 言

随着信息获取技术和遥感探测手段的发展，遥感影像数据日益多元化；数据量呈指数级增长，形成 GB 级、TB 级数据的大规模集成态势[1-2]。海量的遥感影像数据为资源调查、环境监测提供了丰富的资料，但也需看到，由于数据量不断增大，传统的单机集中式存储和检索技术已经难于满足数据高效存储的要求。越来越多的基于遥感数据的应用不仅需要数据的高效存储和管理，而且还需要将结果快速反馈显示到终端上，这些都是目前遥感影像处理中的基础研究问题[1]。

金字塔模型是实现海量遥感影像可视化的基础[3]。如刘义等人[4]将传统的 MapReduce 方法加以改进，构建了适合遥感影像金字塔模型的图像存储算法，较传统单机构建金字塔方法表现出较高的时间效率和较好可扩展性；霍树民[5]提出了一种基于 MapReduce 并行编程模型的影像金字塔并行构建技术，相比单机串行方式，遥感影像建塔速度有明显的提高。上述研究都采用了金字塔模型存储遥感数据，有效提高海量遥感数据的存取效率，但构建金字塔所用的时间仍然较长，而将其应用在实际研发设计中，数据的可视化响应较慢，遥感影像处理和显示的效率也难以达到令人满意的程度。因此，亟需对遥感影像数据的金字塔构建方式做出改进，使其性能得以提升，本次研究中即选用了目前较为流行的大数据处理技术。

同时，有研究表明，Spark 是基于内存的计算框架，运用科学设计，使其对于多次迭代的数据处理，比 Hadoop 快很多倍。Spark 的特点是可以把中间数据放在内存，迭代效率高，而遥感影像的金字塔构建过程，就需要多次迭代，并且还要依据本层的金字塔

基金项目： 2018 年大学生创新训练计划项目(201810225186)。

作者简介： 陕唐剑(1997-)，男，本科生，主要研究方向：智能信息处理；钟宏伟(1998-)，男，本科生，主要研究方向：智能信息处理；李林辉(1979-)，女，讲师，主要研究方向：信息处理、数据库技术；魏景琦(1998-)，女，本科生，主要研究方向：智能信息处理；国露方(1997-)，女，本科生，主要研究方向：智能信息处理。

收稿日期： 2019-05-18

数据生成下一层的金字塔数据，但 Spark 本身却并不具备对具有空间特点的遥感影像数据的处理能力。

综上分析可知，本文在 Spark 现有框架基础上，结合 Spark 和遥感影像金字塔构建过程的特点，研发提出基于 Spark 遥感影像金字塔并行构建模型，实现金字塔瓦片数据的分布式存储，进而有效缩短了遥感影像数据金字塔的构建时间。对此拟做研究论述如下。

1　遥感影像数据金字塔存储方法

遥感影像金字塔是地图服务的基础。地图服务包含 2 方面的内容，分别是：格网和编码的选择，遥感数据的组织和管理。研究后可推得阐释解析如下。

1.1　全球剖分网格

地图服务的指定操作区域都是矩形，建立遥感影像金字塔和编码这些矩形的块。假设遥感影像的原始分辨率为 r_0，像素矩阵大小为 $w \times h$，原始遥感影像经过重采样被分成了很多大小相等的图像块，在金字塔结构中，称之为瓦片。瓦片的大小为 $d \times d$ 像素，瓦片矩阵的大小为 $t_r \times t_c$。t_r 可由公式(1) 计算得出：

$$t_r = \lfloor \frac{w}{d} \rfloor, \tag{1}$$

其中，‘⌊ ⌋’表示向下取整。

t_c 由公式(2)计算求得：

$$t_c = \lfloor \frac{h}{d} \rfloor, \tag{2}$$

其中，‘⌊ ⌋’表示向下取整。

遥感影像的金字塔的构造最多级数可以根据原始影像的大小 L_D 和分辨率进行计算。设金字塔的级数为 N，金字塔级数满足的条件如式(3) 所示：

$$\frac{L_D}{d \times 2^N} \geqslant r_0 \geqslant \frac{L_D}{d \times 2^{N+1}}, \tag{3}$$

其中，L_D 表示原始遥感影像的覆盖范围。

遥感影像金字塔的构建中，要对原始遥感影像进行重采样。一种典型的方法就是将 2×2 个像素转化成一个像素。假设存在某一图层的遥感影像，对于第 l 级的分辨率 r_l，运算时可用到如下公式：

$$r_l = r_0 \times 2^{N-l}, \tag{4}$$

第 l 层的像素矩阵大小为 $t_{rl} \times t_{cl}$。其中，t_{rl}，t_{cl}. 的取值则如式(5)、式(6) 所示：

$$t_{rl} = \lfloor \frac{w}{d \times 2^l} \rfloor, \tag{5}$$

$$t_{cl} = \lfloor \frac{h}{d \times 2^l} \rfloor, \tag{6}$$

假设 *latitude* 和 *longitude* 分别代表某点的经度和纬度，则该点所在的 *level* 级的瓦片编码的运算可写作如下数学形式：

$$r = \lfloor \frac{latitude + 90}{L_D \times 2^{level}} \rfloor, \tag{7}$$

$$c = \lfloor \frac{longitude + 180}{L_D \times 2^{level}} \rfloor.$$

1.2　金字塔结构

拼接后的遥感图像的大小超过 300 MB，有时甚至超过 10 GB 或 100 GB，金字塔结构是海量遥感影像可视化的基础[6]。金字塔的最低层是分辨率最高的，也就是原始图像，数据量最大；随着层数的增加，其分辨率逐渐降低，数据量也按比例减少。每一层影像金字塔都有其分辨率，当在前端进行图像的放大、缩小、漫游时，后台需计算出执行该操作后所需的影像分辨率及在当前视图范围内会显示的地理坐标范围，再根据这个分辨率去和已经建好的影像金字塔分辨率匹配，选择适合的分辨率的瓦片遥感数据，将其从后台取出来显示给用户。采用金字塔模型存储数据的优点是，当对数据进行显示操作时，不需要读取全部原始数，只需选择合适分辨率的数据进行数据操作，这样可以减少数据的 I/O 操作和网络间的数据传输。

通常，影像金字塔的构建方法有 2 种。一种是本身已有的多种分辨率的数据源直接构建相应分辨率的金字塔；另一种是只有原始图像，而没有其它低分辨率的图像，通过已有的遥感数据利用重采样的方式进行逐层构建。本文中采用的是第二种方法。

从原始影像数据中抽取数据构建金字塔时，形成一个多分辨率层次。从金字塔的底层到顶层，分辨率越来越低，但是表示的范围却是一致的。而通过对低一层影像的重采样建立了本层图像，通过构建不同层的图像，最终建立了同一范围、但不同分辨率的一系列图像层。其中，底层分辨率最高，上层逐渐减少，图像金字塔如图 1 所示。

在金字塔中将原遥感影像数据作为最底层数据，记为 0 层，采用等分方式来划定分块，每一个数据块称为瓦片。在金字塔中，一个瓦片可由坐标 (*level*，*row*，*col*) 提供唯一标识。其中，*level* 标识瓦片所在影像金字塔的层数，*row* 标识瓦片所在的行，

col 标识瓦片所在的列。通过前文设定,可以对瓦片数据进行编码。分层、分块后所得的瓦片编码如图 2 所示。

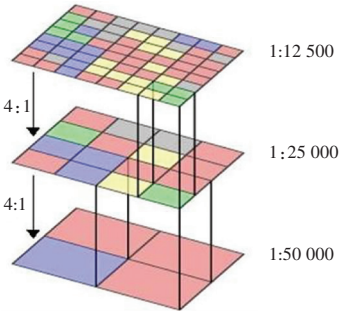

图 1 金字塔结构

Fig. 1 Pyramid structure

Level2

(0,1)	(0,1)

Level1

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)

Level 0

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)

图 2 编码方式

Fig. 2 Coding mode

对一幅 $2^n \times 2^n$ 的遥感影像进行金字塔内瓦片数据构建的过程是从原始影像数据开始,按照倍率是 4 进行逐级采样,接着将得到的遥感数据设定为 $2^m \times 2^m$ 后,再进行倍率是4的逐级采样,由此得到的遥感影像的层数记为 L,则 L 的数学公式可表示为:

$$L = n - m + 1, \quad n \geqslant m, \tag{8}$$

采用上面的方法进行分层后,可以计算求出瓦片的数量 T 为:

$$T = \sum_{k=0}^{n-m} (2^{n-m} \times 2^{n-m}) \times (\frac{1}{4})^K = \frac{1}{3}(4^{n-m+1} - 1) = \frac{1}{3}(4^L - 1), \tag{9}$$

在此基础上,可以推得计算瓦片数据的存储空间比单独存储原图像所超出的百分比的数学公式如下所示:

$$R = \frac{T - T_0}{T_0} = \frac{1}{3}(1 - 4^{1-L}). \tag{10}$$

分析后可知,通过金字塔的构建后,数据存储所用的空间会比单独存储原数据多出三分之一,在大数据时代,大容量存储设备的价格较为低廉,因此对于应用而言,产生的影响可以忽略。对于瓦片数据的大小,没有固定的划分标准,在理论上可以按任意大小划分,但在技术应用层面上存在一些困难,如网络传输、金字塔索引的构建、I/O 的效率。如果瓦片数据尺寸太大,导致浏览遥感影像数据时会有多余的数据加载到内存;如果瓦片数据尺寸太小,瓦片的大小多会根据具体情况和多次实验来求得,从瓦片求取以及容易建立索引的角度出发,通常瓦片遵循如下规则:瓦片数据的大小是一样大小的,瓦片按一定顺序存储。瓦片数据的大小在 SDE 中为 128×128 像素。在 Oracle 的 GeoRaster 中,瓦片数据的大小为 256×256 像素[7-8],通过实验,本文中设置瓦片的大小为 256×256 像素。

2 基于 Spark 的遥感影像金字塔瓦片数据的并行构建

使用 Spark 进行遥感影像的构建是基于传统金字塔构建的模型,并在 Spark 中利用算法将原始图像通过重采样,逐层生成瓦片数据,具体就是由下层数据生成上层数据,运算产生的中间结果存于内存,最终结果存入到 HDFS 中。每层构建瓦片数据设定成一个独立的任务,通过 Spark 在集群中实现构建。根据 Spark 的特点,研究推得构建生成瓦片数据的算法详述如下。

(1)读入遥感影像原数据,生成 Level0 层的瓦片数据。在研究中,将根据遥感影像的坐标范围和全球格网划分方法,确定影像的范围,再根据范围使用公式(8)即可确定此遥感数据总共分为多少层。原始数据的分辨率不同,遥感数据的分层数也会不同。分辨率高的遥感数据所分的层数要高于分辨率低的遥感数据所分层数。

(2)对每层遥感数据进行瓦片化的处理。通过对低层数据的重采样,采用倍率为 2 的采样得到瓦片数据,由上层遥感数据的坐标可计算求出每个瓦片数据的坐标,此过程使用 Spark 实现。传统的大数据处理通常采用 MapReduce 进行处理,Spark 处理相比 MapReduce 处理的优势就在于瓦片数据的生成是通过 rasterRdd 计算得到的,在计算过程中得到的中间数据可以被缓存到内存中,且比传统的构建时间要短,这是因为操作在内存中,从内存中就可以直接读取中间数据,而不需要再从磁盘中读取,这样做可以减少从磁盘读取数据的次数,缩短了影像计算处理的时间,提高了瓦片数据的构建速度,提高了效率。

(3)瓦片数据的属性确定。通过 Spark 计算得

到的瓦片数据最终存储到 HDFS 的各个节点中，每个瓦片数据都有各自的属性，重点包括每个瓦片数据的坐标信息、瓦片数据所存的具体位置、瓦片数据的偏移量。为了做到快速检索，通常对同一层的瓦片数据建立索引，在本研究中则将这些特征存入 Hbase 中。

(4)瓦片数据可视化。通过前述步骤完成了瓦片数据的构建，但要进行可视化时，可以通过需要显示的影像范围确定需提取的是哪一层数据，再根据不同层的编码值对瓦片来执行分布式计算，通过 master 节点将具体的 request 分发到有相应资源数据的计算节点上，计算节点通过索引参数较快地定位原数据，进而将需要的数据返回给用户。

由于 Spark 的特点，在对遥感数据进行瓦片数据的构建过程中，中间生成的影像数据，将会存入内存中，进而生成高层瓦片，中间处理过程中产生的无用数据则无需存入 HDFS，减少了 I/O 操作，提升了构建效率，与传统的基于 MapReduce 模式的构建方式相比，有较大优势。

3　实验结果及分析

实验环境配置为：Master，选用 IBM System x3850 X5(Xeon E7-4807 6 核 1.87 GHz，16 GB 内存，500 GB 硬盘)。Slave，选用 4 台曙光 I450-G10(Xeon E5-2407 四核 2.2 GHz，8 GB 内存，HDFS 总容量 3.34 GB)。通过 Xen 虚拟化为 16 个节点。采用 Tenda TEG1 024 G 千兆以太网交换机连接。操作系统，选用 Red Hat Enterprise Linux 6.2(内核版本2.6.32)，Hadoop 版本选用 2.5。

将金字塔构建算法部署到面向 Spark 计算框架和面向 Hadoop MapReduce 计算框架下的分布式实验环境中，实验中的数据源和参数保持不变，同时保证计算节点的资源一致，实验的数据量大小为 1 GB、10 GB，实验的结果如图 3 所示。

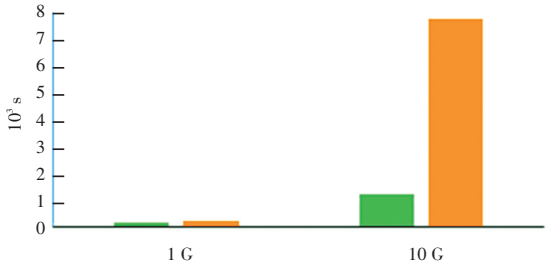

图 3　Spark 框架和 Hadoop MapReduce 框架构建金字塔时间对比

Fig. 3　Time comparision of pyramid construction between Spark framework and Hadoop MapReduce framework

通过实验结果表明，在数据量不大的情况下，Spark 和传统 MapReduce 的构建金字塔数据所用时间相差不大，但随着数据量的增大，Spark 构建金字塔数据的优势较为明显，所用时间较少，因此使用 Spark 框架进行海量遥感影像数据的金字塔构建在时间上具有更好的效率。

4　结束语

由于传统的单机集中式存储和检索技术已经不能满足数据高效存储和快速访问的要求，本文采用基于内存计算的计算引擎 Spark 来构建遥感影像金字塔并给出了算法的构建过程，实现了基于 Spark 的遥感影像金字塔的并行构建。性能和计算效率都有较大提高。

参考文献

[1] 李德仁，张良培，夏桂松. 遥感大数据自动分析与数据挖掘[J]. 测绘学报，2014，43(12)：1211-1216.

[2] 吕雪峰，程承旗，龚健雅，等. 海量遥感数据存储管理技术综述[J]. 中国科学：技术科学，2011，41(12)：1561-1573.

[3] 邓雪清. 栅格型空间数据服务体系结构与算法研究[J]. 测绘学报，2003，32(4)：362.

[4] 刘义，陈荦，景宁，等. 利用 MapReduce 进行批量遥感影像瓦片金字塔构建[J]. 武汉大学学报(信息科学版)，2013，38(3)：278-282.

[5] 霍树民. 基于 Hadoop 的海量影像数据管理关键技术研究[D]. 长沙：国防科学技术大学，2010.

[6] CHEN Shizhi，TIAN Yingli. Pyramid of spatial relations for scene-level land use classification[J]. IEEE Transactions on Geoscience & Remote Sensing，2015，53(4)：1947-1957.

[7] LV Xuefeng，CHENG Chengqi，GONG Jianya，et al. Review of data storage and management technologies for massive remote sensing data[J]. Science China Technological Sciences，2011，54(12)：3220-3232.

[8] LV Zhenhua，HU Yingjie，ZHONG Haidong，et al. Parallel K-means clustering of remote sensing images based on MapReduce[M]//WANG F L，GONG Z，LUO X，et al. Web information systems and mining. WISM 2010. Lecture Notes in Computer Science. Berlin/Heidelberg：Springer，2010，6318：162-170.