

引文格式：刘世永,吴秋云,陈 萃,等.基于高层级地图瓦片的低层级瓦片并行合成技术[J].地理信息世界,2015,22(6):51-55.

基于高层级地图瓦片的低层级瓦片并行合成技术

刘世永, 吴秋云, 陈 萃, 李 军, 景 宁

(国防科学技术大学 电子科学与工程学院, 湖南 长沙 410073)

基金项目：

国家863计划项目(2015AA123901); 国家863计划项目(2012AA12A405)资助

作者简介：

刘世永(1992-), 男, 江西赣州人, 摄影测量与遥感专业硕士研究生, 主要研究方向为高性能地理计算、遥感影像并行处理等。

E-mail:

lsy97_cug@163.com

收稿日期：2015-11-18

【摘要】瓦片金字塔具有多个层级, 以传统方法来说, 创建瓦片金字塔需先对影像创建金字塔, 而后读取各层级金字塔数据进行切片。随着单幅遥感影像的数据量越来越大, 其影像金字塔的数据量也相应越来越大, 传统方法构建瓦片金字塔在面对大数据量遥感影像时其效率远远达不到应用需求。针对这种问题, 本文提出一种名为paraTileM的并行算法, 该算法可以利用高层瓦片来快速合成低层瓦片, 实验结果表明: 本文算法极大地提高了瓦片金字塔的构建效率, 为WebGIS的数据快速发布提供了有力的支持。

【关键词】遥感; 瓦片金字塔; MPI; 并行

【中图分类号】P208; TP311

【文献标识码】B

【文章编号】1672-1586(2015)06-0051-05

Low-level Tile Parallel Synthesis Technology Based on High-level Map Tiles

LIU Shiyong, WU Qiuyun, CHEN Luo, Li Jun, JING Ning
(College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract: Pyramid tile having a plurality of levels, for the traditional method, the creation of an image tile pyramid should create an image pyramid firstly, then read the data at all levels of the pyramid slice. With the volume of remote sensing image data is growing, and correspondingly the image data pyramid is growing. The traditional way to build up pyramids of large amount of remote sensing data is far less than its application requirements. This paper presents a parallel algorithm called paraTileM, which can be used to synthesize the lower level tile, and experimental results show that the proposed algorithm greatly improves the efficiency of the construction of the pyramid tiles for WebGIS data.

Key words: remote sensing; pyramid; MPI; parallel calculate

0 引言

随着遥感卫星传感器的发展, 遥感影像的时间、空间分辨率不断提高^[1], 如何对大数据量, 高时效性的遥感数据快速生成瓦片金字塔, 是高性能WebGIS快速可视化急需解决的问题。传统方法生成瓦片金字塔, 必须先生成影像金字塔, 而后读取影像金字塔各个层级, 再对该层级数据进行切片形成瓦片金字塔。当影像数据量非常大的时候, 建立各层级金字塔, 以及分别读取各层级金字塔数据进行切片, 这是非常耗时的一个操作, 目前, 国内外在瓦片金字塔的快速生成技术上都进行了大量的研究, 欧阳锋等人设计了一种通用瓦片生成引擎, 生成效率与已有工具相比有大幅提高^[2], 黄梦龙研究了瓦片地图技术在桌面端GIS中的应用^[3], 刘义等提出利用MapReduce进行批量遥感影

像瓦片金字塔构建的方法^[4], 霍亮等对瓦片金字塔模型技术进行了研究与实践^[5], 徐虎等研究了基于中间件的瓦片地图服务设计与实现^[6], 殷福忠等研究以瓦片金字塔地图的形式在Web上发布各种地图数据的关键技术, 包括瓦片金字塔地图的生成、存储、发布、应用等^[7], 王慧, 申家双等在影像金字塔模型的基础上提出了一种能够快速提供多分辨率影像、具有高分布性能的大区域遥感影像管理的瓦片金字塔模型^[8]。但是, 基于已有层级瓦片数据来合成低层瓦片数据的研究尚为数不多, 本文研究了一种基于高层级瓦片来并行合成低层瓦片的技术, 算法先通过预先生成某一高层级的瓦片金字塔, 而后采用MPI并行技术, 利用该层瓦片金字塔以重采样的方式并行合成其低层的瓦片金字塔, 这种方法减少了建立影像金字塔的时间, 并进一步提高了瓦片金字塔整体的生成效率。

1 关键技术

1.1 瓦片金字塔

瓦片金字塔的概念是源于影像金字塔，其主要应用于eb小浏览器上数据的实时发布，由于遥感影像一般数据量较大，受限于网络带宽，以及浏览器的性能，数据如果一次性全部加载则速度很慢，无法达到实时显示的要求，为了加快浏览速度，通过对原始影像切割成各个级别分辨率的瓦片数据，每次显示只加载屏幕显示范围内的瓦片，这些不同分辨率的瓦片就构成了瓦片金字塔。

本文的瓦片金字塔采用Google Map提出的瓦片解决方案，其地图投影采用Web墨卡托投影。假设地球被套在一个圆柱中，赤道与圆柱相切，然后在地球中心放一盏灯，把球面上的图形投影到圆柱体上，再把圆柱体展开，这就形成了一幅墨卡托投影的世界地图，其原点在经纬度(0, 0)处。由于理论上南北极是永远无法投影到圆柱体上，并且随着纬度的增高其变形越大，为了方便，Web墨卡托投影忽略了墨卡托投影中南北两级变形较大的区域，把椭圆形的地球投影成平面上边长等于赤道周长的正方形，其大地坐标范围为

[-180, -85.051 128 779 806 59, 180, 85.051 128 779 806 59]，投影坐标范围为

[-20 037 508.342 789 2, -20 037 508.342 789 2, 20 037 508.342 789 2, 20 037 508.342 789 2]，单位为 m。

瓦片金字塔则是基于这个投影坐标系统，进行不同分辨率的切分。以一幅世界地图为例， n 表示地图的级别，Google Map把一幅世界地图分为0到22个层级，每个层级对应相应分辨率的瓦片金字塔图层。各层级瓦片金字塔分辨率Resolution计算公式如式(1)。

$$Resolution[n]=\frac{2\ 003\ 758.342\ 789\ 2\times 2}{256\times 2^n}$$

每层金字塔共有 4^n 个 256×256 的瓦片， n 为瓦片金字塔级别。各层级瓦片划分采用四叉树方法，即以赤道和本初子午线的交点作为中心，不断对地图进行四分，直到每个格网大小为 256×256 为止。如图1所示，0级世界地图由一个瓦片表示，1级世界地图应由4个瓦片表示，往下每一级以此类推。因此，可以按相应规

则将四块相邻高层级瓦片拼合成一块瓦片，然后再将该瓦片进行重采样，形成下一级瓦片。当对一个遥感影像进行切片时，首先，根据遥感影像的分辨率找到与其最接近的瓦片金字塔层级，而后，通过以上所述的世界地图切分规则，计算影像所在该层世界地图瓦片中的位置，进行切片。

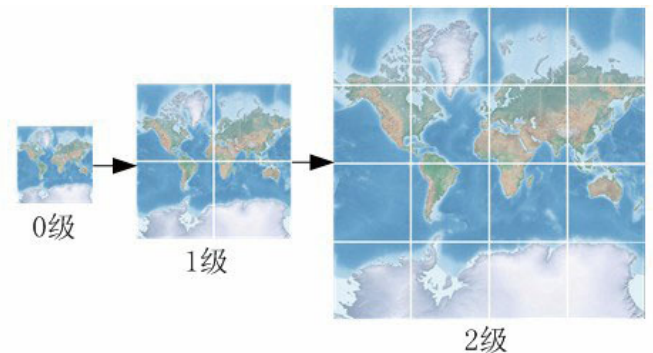


图1 四叉树划分
Fig.1 Quadtree partitioning

空间索引是空间数据库和GIS中的一项关键技术，是根据空间对象之间存在的某种空间关系，按照一定顺序排列成的一种空间数据结构，它包含着空间对象的概要信息^[9]，空间索引直接影响着空间对象的操作速度与效率。目前，绝大多数瓦片地图索引都采用网格索引，这种方式简单方便，本文中采用的Google Map瓦片组织方式，其索引策略就属于其中的一种。具体来说，Google Map中瓦片的索引机制是TMS(Tile Map Service)规范的一个变种，如图2所示，地图被划分为 4^n 个瓦片后(n 为瓦片金字塔级别)，与图像的像素坐标系类似瓦片坐标的原点位于左上角，瓦片存储采用文件系统，分为三级目录其中第一级为瓦片级别(level)，第二级为瓦片列号(col)，第三级为瓦片行号(row)。只需要正确解析地址中的level/col/row就可以获取相应的地图瓦片。通过这个原理，我们可

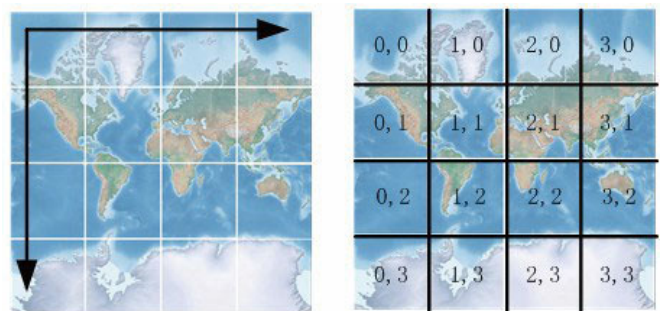


图2 瓦片坐标系统
Fig.2 Tile coordinate system

以通过高级瓦片的索引号来计算下一层级瓦片的索引号。

1.2 瓦片并行合成方法的任务划分

各进程瓦片数据划分方法如图3所示, 根据前文所述的瓦片坐标系统规定, 每个进程一次处理一组瓦片,

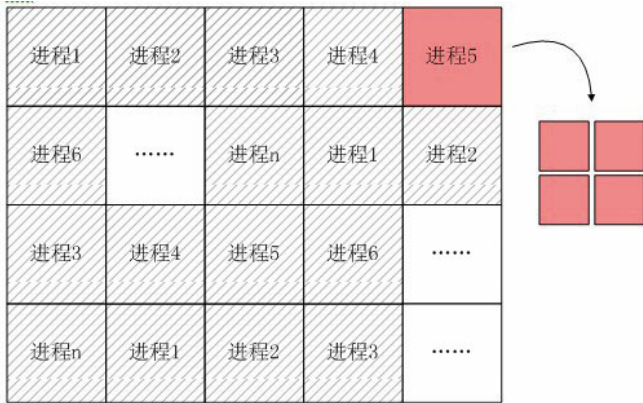


图3 瓦片数据划分方法
Fig.3 Tile data partition method

每组包含四块上下左右相邻的瓦片。各进程采用车轮法, 依次处理所有的瓦片数据。

因此只需确定每组瓦片中的左上角瓦片的坐标 (X, Y) 即可确定该组其余瓦片的坐标, 假设在该层瓦片集合中的瓦片最小列号为 $tminX$, 最小行号为 $tminY$, 最大列号为 $tmaxX$, 最大行号为 $tmaxY$, 根据TMS规范, 用于合成的每组瓦片中左上角瓦片的值应为偶数, 所以当 $tminX$ 和 $tminY$ 为奇数的时候, 其不是第一组瓦片的左上角瓦片, 应对其向左扩展一个瓦片。用于任务分配的瓦片外包框 $[\min X, \min Y, \max X, \max Y]$, 应满足式 (2) 式 (3)。

$$\min X = \begin{cases} tminX & \min X \% 2 = 0 \\ tminX - 1 & \min X \% 2 \neq 0 \end{cases} \quad (2)$$

$$\min Y = \begin{cases} tminY & \min Y \% 2 = 0 \\ tminY - 1 & \min Y \% 2 \neq 0 \end{cases} \quad (3)$$

$$\max X = tmaxX \quad (4)$$

$$\max Y = tmaxY \quad (5)$$

从瓦片坐标为 $(\min X, \min Y)$ 的瓦片开始, 每四个相邻的瓦片划分为一组, 每个进程以组为对象进行处理。假设下一层瓦片集合外包框的长为 $width$, 宽为 $height$, 则 $width, height$ 满足式 (6)、式 (7)。

$$width = \text{ceil}((\max X - \min X) / 2) \quad (6)$$

$$height = \text{ceil}((\max Y - \min Y) / 2) \quad (7)$$

假设进程总数为 n , 则第 i 个进程 $\text{rank}(i)$ 所属的每组瓦片左上角瓦片坐标 (X, Y) 满足式 (8) 式 (9)。

$$X = ((k \times n + i) \% width) \times 2 \quad (8)$$

$$Y = ((k \times n + i) / width) \times 2 \quad (9)$$

$$k = m \times n \quad m \in [0, size/n] \quad (10)$$

$$size = width \times height \quad (11)$$

1.3 基于高层地图瓦片的低层瓦片并行合成技术

如图4所示, 首先将瓦片的坐标与其文件路径构建一个Hash表, 使得每个瓦片坐标与其索引路径能够一一映射, 而后根据前文所述的任务划分规则, 每个进程采取车轮法按顺序处理其所属瓦片组的四块邻近瓦片。具体步骤如下: 各进程根据式 (8) 和式 (9) 计算所属瓦片组的左上角瓦片 Tile1 的坐标 (X, Y) , 然后推导出其邻近的3个瓦片的坐标分别为 $\text{Tile2}(X+1, Y)$, $\text{Tile3}(X, Y+1)$, $\text{Tile4}(X+1, Y+1)$ 。各进程通过之前Hash表中坐标与路径之间的映射关系, 根据瓦片坐标获取瓦片所在路径, 而后利用GDAL库的 RasterIO 函数依次以降采样方式读取这四块瓦片到缓冲区中, 即每个瓦片长宽各重采样至原瓦片数据的一半大小。

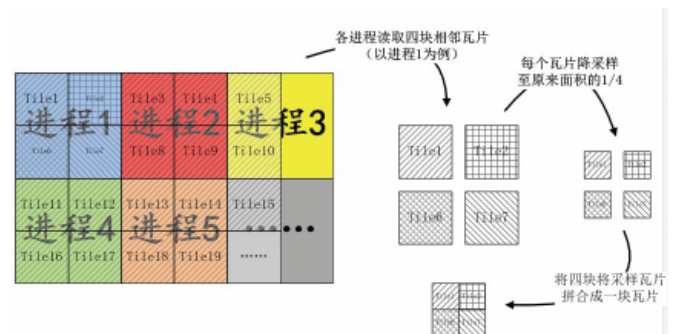


图4 处理流程

Fig.4 Processing flow

数据重采样完成后, 计算下一层级结果瓦片新的索引坐标。假设当前瓦片金字塔级别为 $level$ 级, 则其下一级 $level-1$ 层瓦片的坐标 (nX, nY) 满足式 (12)、式 (13)。

$$nX = X/2 \quad (12)$$

$$nY = Y/2 \quad (13)$$

根据瓦片坐标 (nX, nY) 构建结果瓦片的索引路径, 假设存放路径根目录为 $root$, 则该瓦片的索引路径为 $root/level-1/nX/nY.jpg$, 按照瓦片的索引路径把缓冲区中的重采样数据保存成瓦片文件输出。往下各层级瓦片操作类似于上述步骤, 采用递归的方式, 不断基于

上一层瓦片来生成下层瓦片，直至所有瓦片最终合并成一张瓦片文件为止。

2 试验结果分析

为了保证实验结果的准确性，各算法在相同实验平台下进行。实验平台采用高性能集群，节点间通过万兆网络进行通信，并行文件系统采用IBM的GPFS(General parallel File System)，集群详细硬件环境见表1。

表1 集群环境
Tab.1 Cluster environment

类型	描述
处理器	Intel Xeon16core*12,2GHz + 8core*8,2.4GHz
内存	64G*12 + 16G*8
互连网络	Infiniband万兆网
操作系统	CentOS6.3.X86_64
文件系统	GPFS(General parallel File System)

实验数据采用不同大小级别的遥感影像数据，涵盖单波段、多波段，以及多种数据类型。分别对其进行切片构建0-16级瓦片金字塔，测试数据详细信息见表2。

表2 实验数据
Tab.2 Testing data

名称	尺寸 (pixel)	数据类型	波段 (个)	大小 (GB)
1.TIF	25 600 × 25 600	Byte8	3	1.83
2.TIF	38 048 × 38 734	Byte8	3	4.63
3.TIF	68 224 × 45 184	Byte8	3	11.48
4.TIF	72 001 × 48 001	Float32	1	12.90

2.1 实验1：评价程序的并行化程度

为了测试本文算法paraTileM的并行化程度，实验选取上述3. TIF影像的17级瓦片为测试数据，为其构建0-16瓦片金字塔，记录算法总耗时随进程总数的变换情况。取5次测试结果的平均值作为实验结果，实验结果如图5所示。

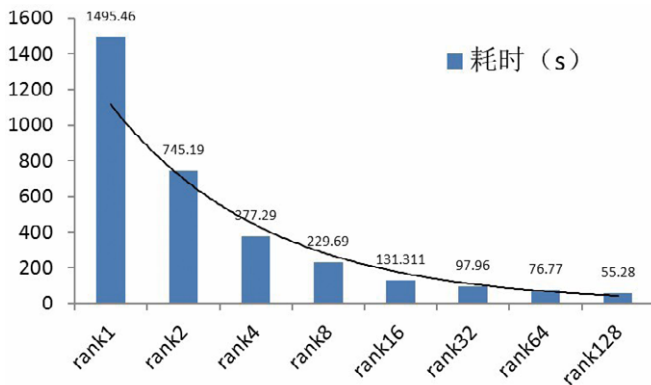


图5 算法总耗时随进程总数的变换情况
Fig.5 The algorithm time cost width in different process

图5为算法总耗时随进程总数的变换情况，横坐标为进程数，纵坐标为算法耗时，从实验结果上可以看到，随着进程数的增加算法速度逐步提升，当进程数到达一定程度后，算法速度逐渐趋于一个稳定值不再继续升高，其原因主要在于，当进程数逐渐增加后，算法I/O最终达到硬盘最大读写性能，受限于硬盘的读写速度，算法速度无法再继续提升，最终保持在一个相对稳定的值附近。图6为算法加速比，横坐标为进程数，纵坐标为加速比，从图中可以看到当进程数达到16个后，其加速比的增长速率开始下降，在16个进程以前，算法效率与进程数大致是呈线性增长关系，可以看到算法的并行化程度还较高。

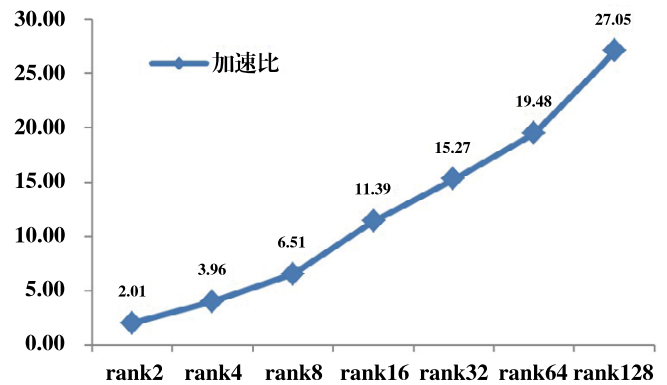


图6 算法加速比
Fig.6 The speed-up ratio of the algorithm

2.2 实验2：paraTileM算法与MapNik切片算法性能对比

Mapnik是一个用来开发 GIS 应用程序的工具包，其核心是一个 C++ 的共享库，用以提供空间数据访问和可视化的算法和模式，其支持ESRI的Shape文件、PostGIS、TIFFraster、OSM XML、Kisme和OGR/GDAL等325种格式，可以在多种操作系统下运行，并且支持多线程环境，现在主流的GIS服务的Web应用开发都采用其切片和渲染功能。为了验证本文算法与其性能优劣，实验在相同的集群环境下运行，进程数都采用16，测试数据采用1. TIF、2. TIF、3. TIF、4. TIF上述四幅遥感影像，比较在不同影像规模下两种算法的性能差异，实验结果如图7所示。

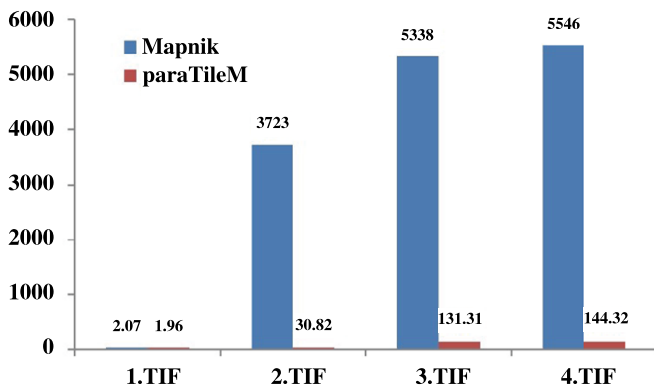


图7 paraTileM与Mapnik性能对比
Fig.7 Performance compare of paraTileM and with Mapnik

通过实验结果可以看到, Mapnik在切小影像时, 能够保持较高的效率, 但是一旦影像的数据量达到一定的规模后, 其效率开始迅速下降, 而paraTileM则可以一直保持较高的效率。通过深入比较, 可以看到随着影像数据量规模的增大本文算法效率相比Mapnik具有数量级的提升, 并且随着数据量增大这种优势愈加明显。通过上述测试结果, 表明paraTileM算法的可扩展性强, 稳定性高。

3 结束语

本文通过深入研究瓦片金字塔的结构类型以及其索引机制, 针对当前Mapnik切片算法效率低等不足, 提出一种基于已有层级瓦片来生成下层瓦片的并行合成算法, 算法可以在缺乏影像金字塔数据时, 基于影像当前层的瓦片来快速合成瓦片金字塔的其他层级瓦片, 减少需事先生成影像金字塔的时间。实验表明paraTileM算法与传统算法一层层切片构成瓦片金字塔相比, 性能得到了极大提升, 特别是随着数据规模的增大, 这种优势

愈加明显, 并且算法稳定性好, 并行程度高, 目前, 该算法已经实际应用国土测绘部门, 极大地提高了数据发布以及成果检测的工作效率。

参考文献

- [1] 李德仁, 童庆禧, 李荣兴, 等. 高分辨率对地观测的若干前沿科学问题[J]. 中国科学: 地球科学, 2012, 42(6) :805-813.
- [2] 杨超伟, 李琦, 承继成. 遥感影像的Web发布研究与实现[J]. 遥感学报, 2000, 4(1) :71-75.
- [3] 欧阳峰, 杨春成, 谢鹏, 等. 通用瓦片生成引擎设计与性能优化[J]. 测绘科学技术学报, 2014, 31(2) :177-184.
- [4] 黄梦龙. 瓦片地图技术在桌面端 GIS 中的应用[J]. 地理空间信息, 2011, 9(4) : 149-151.
- [5] 刘义, 陈莹, 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.
- [6] 霍亮, 杨耀东, 刘小勇, 等. 瓦片金字塔模型技术的研究与实践[J]. 测绘科学, 2012, 37(6) :144-146.
- [7] 殷福忠, 孙立民. 基于瓦片金字塔技术的地图发布平台开发研究[J], 测绘与空间信息, 2010, 33(5) :16-20.
- [8] 王慧, 申家双, 陈冬阳, 等. 一种高性能的大区域遥感影像管理模型[J], 海洋测绘2006, 26(3) :71-74
- [9] 许虎, 聂云峰, 舒坚. 基于中间件的瓦片地图服务设计与实现[J], 地球信息科学学报, 2010, 12(4) :562-567.
- [10] Bo LI, Hui ZHAO, ZhenHua LV. Parallel ISODATA Clustering of Remote Sensing Images Based on MapReduce[C]. 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, IEEE, 2010:380-383.
- [11] CZ Qin, LJ Zhan, A Zhu, How to Apply the Geospatial Data Abstraction Library(GDAL) Properly to Parallel Geospatial Raster I/O[J]. Transaction in GIS, 2013.

本刊现入编《中文科技期刊数据库》(维普网), 作者著作权使用费与本刊稿酬一次性给付, 不再另行发放, 敬请注意。如作者不同意将文章入编, 请在投稿时特别说明。