

基于瓦片金字塔模型的海量空间数据快速分发方法

殷君茹, 侯瑞霞, 唐小明, 罗 鹏

(中国林业科学研究院 资源信息研究所, 北京 100091)

摘要: 针对分布式并行环境下海量空间数据的快速显示和浏览问题, 提出一种基于分层、分专题的海量空间数据金字塔模型及基于数据库存储方案的瓦片数据快速分发方法。对比分析了瓦片数据在文件式管理和数据库管理两种模式下的存储机制和响应流程, 并以时间跨度 59 年的沙尘暴观测数据为例, 验证了该方法不仅能有效地组织管理海量空间数据, 实现高效数据互操作, 而且在多用户并发访问时, 能快速响应客户端请求。

关键词: 瓦片金字塔模型; 海量空间数据; 数据库; 分布式并行环境

中图分类号: TP393 **文献标志码:** A **文章编号:** 1671-5489(2015)06-1269-06

Fast Distribution Method for Large Data Set of Spatial Data Based on Tiles Pyramid Model

YIN Junru, HOU Ruixia, TANG Xiaoming, LUO Peng

(Institute of Resource Information Techniques, Chinese Academy of Forestry, Beijing 100091, China)

Abstract: In view of the problem of showing and browsing large data set of spatial data in the distributed parallel environment quickly, not only one kind of massive spatial pyramid model based on a hierarchical and special topic mechanism, but high access performance tile map service based on database storage scheme under the tiles pyramid model technology were proposed. The mechanism of tile data file management and database management under the storage performance and response process were comparatively analyzed. Moreover, sand and dust storm observation data for 59 years was taken as an example to validate that this method can not only effectively organize and manage massive spatial data to achieve efficient data interoperability, but also quickly respond to client requests in the multi-user concurrent access.

Key words: tiles pyramid model technology; large data set of spatial data; database; distributed parallel environment

目前, 基于网络地理信息系统(Web geographic information system, WebGIS)技术实现海量空间数据的发布和共享应用广泛, 但当大量用户并发访问地图服务时, 采用即时动态渲染地图服务技术对其进行可视化表达, 会出现网络负载过大及响应速度慢等问题^[1]。同时, 信息获取技术的飞速发展使国家积累了丰富的历史和现势沙尘暴监测数据资源^[2], 因而采用动态渲染地图服务技术已不能满足海

收稿日期: 2014-11-17.

作者简介: 殷君茹(1986—), 女, 汉族, 博士研究生, 从事 GIS 开发与应用的研究, E-mail: yinjr1986@163.com. 通信作者: 侯瑞霞(1975—), 女, 汉族, 博士, 助理研究员, 从事 GIS 开发与应用的研究, E-mail: hrx2004126@126.com.

基金项目: 国家林业公益性行业科研专项基金(批准号: 201304215).

量沙尘暴观测数据快速显示和浏览的需要。

目前 google 地图、百度地图、天地图等通用的互联网地图均采用瓦片地图方案实现在线地图服务, 因此基于地图预生成思想的瓦片技术以其访问流畅、数据丰富等特点, 已逐渐成为新一代 WebGIS 的事实标准^[3-5]。瓦片技术是一种地图预缓存技术, 通过金字塔模型构建地图缓存库, 将地图服务的实时动态渲染转化为对瓦片数据文件的静态请求, 有效减轻了服务器端实时渲染结果地图的处理压力, 提高了系统的响应能力。而影响访问速率的因素, 还包括对瓦片数据文件的有效存储和管理, 主要有文件式管理和数据库管理两种方式^[3,6-10]。当瓦片数据包含大量小文件时, 基于文件方式进行管理虽简单易操作, 但会出现管理和存储效率的问题。因此, 本文在基于瓦片金字塔模型构建海量空间数据的瓦片数据文件基础上, 通过分析文件式管理和数据库管理两种管理模式下的存储机制及瓦片数据文件的响应流程, 并通过实验对比两者的存储性能和大量用户并发访问时的响应时间, 同时结合沙尘暴监测数据分布式架构应用场景的特点, 提出一种基于文件数据库管理的方法, 以利于海量空间数据的快速分发和访问。

1 海量空间数据金字塔模型的构建

1.1 瓦片金字塔模型技术原理

瓦片金字塔模型技术是指先将数据库中由空间数据生成的地图数据文件按指定尺寸和指定格式切成若干行或列的矩形图片(即地图瓦片, 简称瓦片), 再采用分级、分块的金字塔模型构建多尺度缓存地图库, 并存储在服务器端。金字塔模型是一种多分辨率层次模型, 但从底层到顶层所表达的地理范围不变。分辨率采用倍率方式构建^[8]。设地图数据的原始分辨率为 R_0 , 倍率为 m , 则第 K 层地形数据的分辨率可表示为 $R_k = R_0 \times m^{-k}$ 。基于四叉树结构构建模型具有以下特征:

- 1) 树中任意相邻的层及分辨率层呈双倍递增关系;
- 2) 树中每个节点对应一块区域。

1.2 基于瓦片金字塔模型的构建与算法

地图服务的坐标系统和应用场景影响构建瓦片金字塔模型时所采用的地图投影和构建方法。本文采用 Web Mercator 投影。

1) 在构建瓦片金字塔时, 先确定要显示的层级数 $L = \{l_0, l_1, \dots, l_k\}$, 再确定由空间数据生成的地图数据文件。

沙尘暴监测数据时间跨度为 59 年(1951~2009 年), 即 $T = \{T_0, T_1, \dots, T_{58}\}$, 包括 17 个不同类型的专题信息, 即 $Z = \{Z_0, Z_1, \dots, Z_{16}\}$ 。首先, 将沙尘暴监测数据 S 按照 T 分成 59 个互不相交的子集, 即 $S = S_{t0} \cup \dots \cup S_{ti} \cup S_{tj} \cup \dots \cup S_{t58}$, 且 $S_{ti} \cap S_{tj} = \emptyset$; 其次, 针对 S_{ti} , 按照 Z 分成 17 个互不相交的子集, 即 $S_{ti} = S_{tz0} \cup \dots \cup S_{tzm} \cup S_{tzn} \cup \dots \cup S_{tz16}$ 且 $S_{tzm} \cap S_{tzn} = \emptyset$; 最后针对 S_{tzn} , 根据其所包含的空间数据生成地图数据文件。

- 2) 将 S_{tzn} 作为金字塔的底层 l_0 , 并对其进行分块, 形成底层瓦片矩阵 A_0 。

分块所需的参数主要包括 S_{tzn} 的范围($\min X, \min Y, \max X, \max Y$)、金字塔的层数 l_0 、比例尺 R_0 、瓦片大小、图片格式和空间参考等, 并根据如图 1 所示的原理把 S_{tzn} 分成大小相等的瓦片数据。

3) 以 A_0 为基础, 从地图原点开始, 从左到右、从上到下按 2×2 个像素合成一个像素的原则生成新的像素矩阵, 并进行分块, 形成 l_1 层的瓦片矩阵 A_1 。以此类推, 直到生成 l_k 的瓦片矩阵 A_k 为止, 如图 2 所示。

- 4) 重复操作 2) 和 3) 直到所有的地图文件生成瓦片数据为止。

生成的瓦片数据文件名包含三方面信息(层数 L 、行数和列数), 以方便用户在客户端浏览该区域, 根据客户端地理坐标计算得出瓦片数据的行列数, 从而定位瓦片文件名, 快速获取瓦片存储位置。设投影原点坐标为 (OriginalLatitude, OriginalLongitude), 当前点的坐标为 (CurrentLatitude, CurrentLongitude), 瓦片大小为 TileSize, 则当前层数为 l_i , 比例尺为 R_i 时, 行列数计算公式如下:

$$\text{row} = \text{abs}(\text{OriginalLatitude} - \text{CurrentLatitude}) / (R_i \times \text{TileSize}),$$

$\text{col} = \text{abs}(\text{OriginalLongitude} - \text{CurrentLongitude}) / (R_i \times \text{TileSize})$,

当所得结果为非整数时, 向下取整.

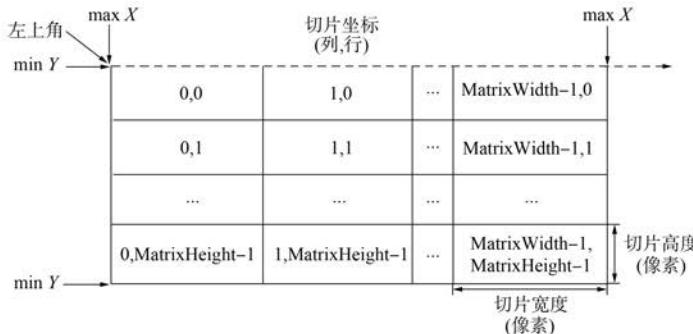


图 1 分块原理

Fig. 1 Principle of partition

1.3 海量瓦片数据的存储和管理

针对已生成大数据量的瓦片数据文件, 必须采取高效的存储和管理机制, 避免数据冗余、互操作差、检索缓慢等情况. 因此下面分别用文件系统和数据库方式对其进行存储和管理, 并对比分析两者的异同.

1.3.1 文件系统存储 基于文件方式存储是通过将相同比例尺下的瓦片数据存储在同一文件夹下, 直接存储和管理地图切片.

采用 ArcGIS Server 的文件方式存储和管理沙尘暴瓦片数据, 将每个图层数据下的瓦片按照不同比例尺组织文件夹, 每个比例尺文件夹下再根据瓦片的行号进行存储, 如图 3 所示.

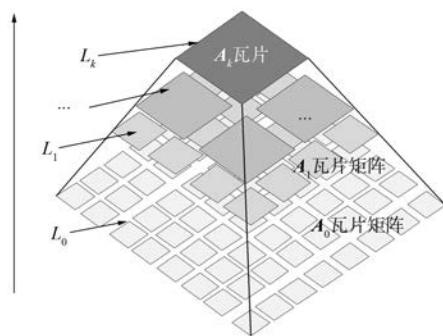


图 2 分层策略

Fig. 2 Hierarchical strategy

图 3 文件系统存储

Fig. 3 File system storage

1.3.2 数据库存储 在瓦片数据文件存储的基础上, 设计基于自定义 vtb 格式的数据库文件, 存储沙尘暴瓦片数据. vtb 格式的数据库文件能被 SQLite^[12]文件型数据库读取, 因此采用 vtb 格式存储的沙尘暴瓦片式地图服务在应用和管理时, 能充分利用 SQLite 数据库的存储数据量大、执行速度快、使用简单等优点^[9]. 且该数据文件采用网格编码的原理对存储的瓦片数据建立索引, 使对目标瓦片的查找次数减到最少, 以提高瓦片的检索效率.

采用“数据库-专题”方式管理沙尘暴瓦片数据, 按专题类型生成 17 个 vtb 格式的数据文件. 每个沙尘暴专题类型数据库文件包含上百个按时间整合的图层数据, 其中每个图层数据又包含不同比例尺下的切片数据. 将该专题下所有图层的瓦片数据建立数据表, 通过数据表存储瓦片的行号、列号、缩放级别、切片数据等信息, 如图 4 所示.

数据库还对该专题所包含的图层信息进行管理, 包括图层名称、所属专题、瓦片尺寸、瓦片格式、投影系统等. 这种方式利于在客户端拼接显示时图层信息的初始化, 如图 5 所示.

1.3.3 瓦片地图服务响应流程 大数据量瓦片地图服务采用基于多层系统结构设计的 WebGIS 平台, 包括客户层、服务层和数据层. 当用户在客户端请求地图服务时, 由应用服务器负责地图请求响

	RecNo	TI	TX	TY	TL	TD
F_SDS_QX_R_SCB_FORT103	56	288230376420147197	-3	0	4	
F_SDS_QX_R_SCB_FORT104	57	288230376688582653	-3	1	4	
F_SDS_QX_R_SCB_FORT105	58	288230376957018109	-3	2	4	
F_SDS_QX_R_SCB_FORT106	59	288230377225453565	-3	3	4	
F_SDS_QX_R_SCB_FORT107	60	288230377493889021	-3	4	4	
F_SDS_QX_R_SCB_FORT108	61	28823037762324477	-3	5	4	
F_SDS_QX_R_SCB_FORT109	62	360287969921204222	-2	-2	4	
F_SDS_QX_R_SCB_FORT110	63	360287970189639678	-2	-1	4	
F_SDS_QX_R_SCB_FORT111	64	288230376420147198	-2	0	4	
F_SDS_QX_R_SCB_FORT112	65	288230376688582654	-2	1	4	
F_SDS_QX_R_SCB_FORT113	66	288230376957018110	-2	2	4	
F_SDS_QX_R_SCB_FORT114	67	288230377225453566	-2	3	4	
F_SDS_QX_R_SCB_FORT115	68	2882303777403880177	-3	4	4	

图 4 数据库存储

Fig. 4 Database storage

RecNo	LYRID	LYRNAME	LYRALIAS	LYRTHEM	TILESIZE	MAXLEVEL	MINLEVEL	MINX	MINY	MAXX
12	12	F_SDS_QX_R_SCB_FORT248	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
13	13	F_SDS_QX_R_SCB_FORT247	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
14	14	F_SDS_QX_R_SCB_FORT246	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
15	15	F_SDS_QX_R_SCB_FORT245	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
16	16	F_SDS_QX_R_SCB_FORT244	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
17	17	F_SDS_QX_R_SCB_FORT243	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
18	18	F_SDS_QX_R_SCB_FORT242	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
19	19	F_SDS_QX_R_SCB_FORT241	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
20	20	F_SDS_QX_R_SCB_FORT240	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
21	21	F_SDS_QX_R_SCB_FORT239	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
22	22	F_SDS_QX_R_SCB_FORT238	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
23	23	F_SDS_QX_R_SCB_FORT237	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
24	24	F_SDS_QX_R_SCB_FORT236	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429
25	25	F_SDS_QX_R_SCB_FORT235	沙尘暴	沙尘暴	256	5	0	-5018242.04444594	-1258664.11973429	3627141.4429

图 5 专题图层元信息

Fig. 5 Meta-information of the thematic layer

应, 应用服务器从瓦片数据资源中取得需要的瓦片地图, 再返回给客户端.

文件方式需要依赖 ArcGIS Server 对沙尘暴大数据瓦片地图服务进行管理和调用, 但不需要额外建立空间索引, 可根据瓦片数据的存储路径直接对其进行访问.

基于数据库方式进行管理, 与文件方式管理不同的是, 可通过应用服务器直接调用沙尘暴大数据瓦片式地图服务, 但需要建立索引读取专题切片数据库文件. 两种管理方式下瓦片式地图服务的响应流程如图 6 所示.

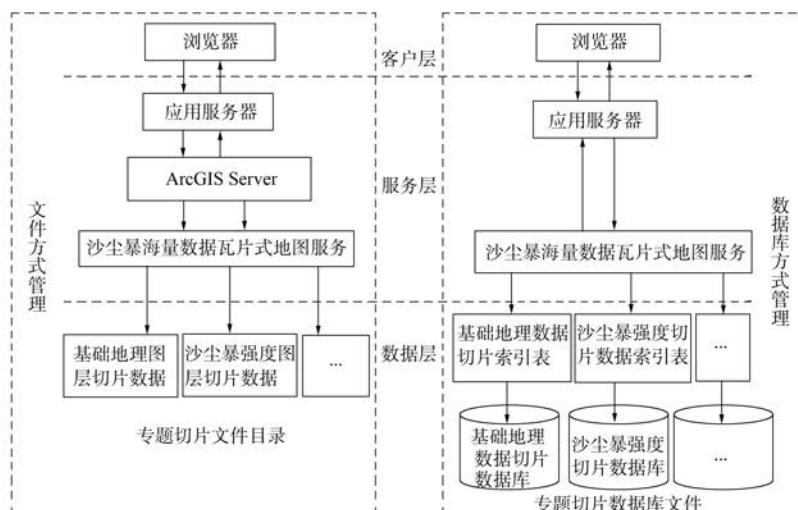


图 6 响应流程对比

Fig. 6 Comparison chart of the two response processes

2 性能对比分析

基于上述分析, 文件方式和基于 vtb 格式的数据库管理方式对沙尘暴大数据量瓦片地图服务的管理机制并不相同。在分布式架构下, 当大量用户并发访问沙尘暴大数量瓦片地图服务时, 存储和管理机制的不同将直接影响地图服务的性能。因此本文通过实验对比分析两者的存储性能, 并进一步分析和对比基于 ArcGIS Server 的文件管理和基于数据库的不同管理机制下, 系统调用沙尘暴大数据量瓦片地图服务时的响应时间。

2.1 存储性能对比

采用 ArcGIS Server 对沙尘暴 1 934 层图层数据进行地图切片的过程中, 会出现生成步骤繁琐、效率不高、远程连接数据库不稳定等情况。针对这些问题, 本文采用自行研发的基于 ArcGIS Engine 二次开发的专题地图缓存工具, 并将切好的瓦片数据直接生成数据库能读取的瓦片数据文件, 进而对两种不同存储方案的性能进行对比。

2.1.1 实验环境 两种切片工具安装在同一台电脑上, 处理器为 Intel(R)Core(TM)i7-2620M, 内存为 4 GB。沙尘暴数据搭建在 Windows7 平台上, 由 Oracle 10g 和空间数据引擎 ArcSDE 10 管理, 并作为生成沙尘暴专题地图切片的数据源。而沙尘暴专题地图切片由沙尘暴数据通过切片工具生成, 生成的地图切片分别用基于 Arcgis Server 的文件方式和基于 vtb 的数据库文件方式进行存储。

2.1.2 存储性能对比分析 沙尘暴数据为由站点多年观测数据生成覆盖全国范围的栅格数据, 为了在宏观上分析全国范围的沙尘暴数据在 59 年间的变化趋势, 对切片的存储和管理采用可视 6 级缩放级别。对应的比例尺从 1 : 160 000 000 到 1 : 5 000 000。切片的生成和存储分别基于文件方式和数据库方式, 对比结果列于表 1。

表 1 性能对比

Table 1 Performance comparison

管理方式	文件大小/GB	磁盘占用空间/GB	磁盘备份时间	瓦片生成时间
文件方式	1.29	3.97	2 h 46 min 12 s	110 h 24 min
数据库方式	1.33	1.33	2 min 39 s	105 h 34 min

由表 1 可见:

1) 由沙尘暴空间数据分别生成的文件方式瓦片数据和数据库方式瓦片数据的大小基本相同, 但在瓦片生成时间上数据库方式所用时间优于文件方式;

2) 在瓦片数据量相同的情况下, 采用瓦片数据库按专题存储所占磁盘空间极大减小, 存储空间仅占文件方式的 33.5%;

3) 数据库方式比文件方式的数据备份时间显著缩短, 提高了分布式架构下沙尘暴大数据量瓦片数据的可迁移性。

2.2 瓦片式地图服务响应时间

2.2.1 实验环境 为了对比基于 ArcGIS Server 的文件管理和基于 SQLite 数据库的不同管理机制下, 系统调用每个专题瓦片地图服务的响应时间, 数据服务器搭建在 Windows Server 平台上, 配置为 Intel(R)Core(TM)2 Quad CPU 处理器和 4 GB 内存, 分别采用 ArcGIS Server 管理瓦片地图文件和 SQLite3 管理瓦片数据库。应用服务器搭建在 Windows7 平台上, 配置为 Intel(R)Core(TM)i7-2620M 处理器和 4 GB 内存, Web 服务器为 Apache Tomcat6, 开发环境为 eclipse Helios 和 Flash Builder4.5。网络访问为同一局域网。在客户机上运行 LoadRunner 性能测试软件进行测试。

2.2.2 并发式响应分析 沙尘暴 17 个专题瓦片地图服务被系统调用的流程相同, 因此测试数据采用其中沙尘暴强度专题的全国分布图, 由按年度组成的 136 个图层数据组成, 瓦片总数为 44 336 个。对比分析在多用户并发访问时, 这两种管理方式下每个专题服务的系统响应时间。

在客户机运行 LoadRunner 性能测试软件, 模拟 100~1 000 用户并发访问, 在两种管理方式下分别对地图进行缩放操作, 针对每种情况对应某一用户数进行 5 次重复测试, 取其平均值作为响应时间,

结果如图 7 所示。由图 7 可见, 在大量用户并发访问的情况下, 数据库瓦片地图服务的响应时间比文件式瓦片服务时间略长 20~68 ms。但随着用户量的不断增加, 两者的响应时间曲线增长幅度趋于平缓, 数据库瓦片地图服务的性能与文件式相当。

综上所述, 本文将瓦片金字塔模型技术应用于海量空间数据的快速访问中, 并通过实验分析比较了文件方式和数据库方式的存储和响应性能。结果表明, 在分布式架构下, 针对海量空间数据的快速访问机制, 采用基于 vtb 数据文件格式的数据库方式进行管理, 除获得与文件方式相当的快速响应性能外, 瓦片数据的存储空间和数据备份时间显著减少, 可迁移性得到提高。

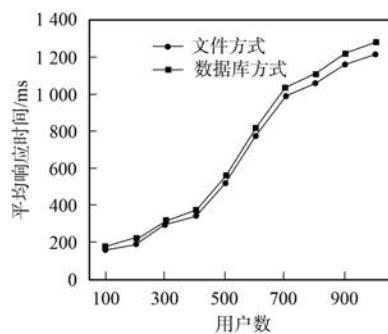


图 7 性能并发测试响应时间

Fig. 7 Response time of concurrency performance test

参 考 文 献

- [1] 李浩松, 朱欣焰, 李京伟, 等. WebGIS 空间数据分布式缓存技术研究 [J]. 武汉大学学报: 信息科学版, 2005, 30(12): 1092-1095. (LI Haosong, ZHU Xinyan, LI Jingwei, et al. Research on Spatial Data Distributed Cache Technology in WebGIS [J]. Geomatics and Information Science of Wuhan University, 2005, 30(12): 1092-1095.)
- [2] 孙伟, 侯瑞霞, 庞丽峰, 等. 荒漠化和沙化监测数据库框架设计与应用 [J]. 西北林学院学报, 2012, 27(6): 139-145. (SUN Wei, HOU Ruixia, PANG Lifeng, et al. Framework Design and Application of Desertification and Sandification Monitoring Database [J]. Journal of Northwest Forestry University, 2012, 27(6): 139-145.)
- [3] 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. (XU Hu, NIE Yunfeng, SHU Jian. Design and Implementation of a TileMap Service Based on the Middleware [J]. Journal of Geo-Information Science, 2010, 12(4): 562-567.)
- [4] Kang Y K, Kim K C, Kim Y S. Probability-Based Tile Prefetching and Cache Replacement Algorithms for Web GeoGraphical Information Systems [C]//The 5th East European Conference on Advances in DataBases and Information Systems. Berlin: Springer, 2001: 127-140.
- [5] Yang C W, Wang D W, Yang R X, et al. Performance-Improving Techniques in Web-Based GIS [J]. International Journal of Geographical Information Science, 2005, 19(3): 319-342.
- [6] 罗智勇, 黎小东. 基于数据库存储方案的高性能瓦片地图服务研究 [J]. 地理与地理信息科学, 2013, 23(3): 48-51. (LUO Zhiyong, LI Xiaodong. High Performance Tile Map Service Based on Database Storage Scheme [J]. Geography and Geo-Information Science, 2013, 23(3): 48-51.)
- [7] 刘义, 陈革, 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. (LIU Yi, CHEN Luo, JING Ning, et al. Parallel Batch-Building Remote Sensing Images Tile Pyramid with MapReduce [J]. Geomatics and Information Science of Wuhan University, 2013, 38(3): 278-282.)
- [8] 李学俊, 王尊亮, 李晓欢, 等. 基于地图切片预取的 WebGIS 系统 [J]. 计算机应用研究, 2009, 26(10): 3849-3851. (LI Xuejun, WANG Zunliang, LI Xiaohuan, et al. WebGIS System Based on Map Tiles Pre-fetching [J]. Application Research of Computers, 2009, 26(10): 3849-3851.)
- [9] 李海亭, 费立凡, 彭青山, 等. 预生成思想在地理信息服务中的应用研究 [J]. 测绘信息与工程, 2009, 34(1): 31-32. (LI Haiting, FEI Lifan, PENG Qingshan, et al. Application of Pre-generation Thinking to Geographical Information Services [J]. Journal of Geomatics Feb, 2009, 34(1): 31-32.)
- [10] 赵大龙, 孙恒宇. 地图切片技术分析与简单实现 [J]. 测绘与空间地理信息, 2010, 33(1): 116-118. (ZHAO Dalong, SUN Hengyu. The Simply Analysis and Implement of Map Tile Technology [J]. Geomatics & Spatial Information Technology, 2010, 33(1): 116-118.)