

三维城市模型数据划分及分布式存储方法

李朝奎, 严雯英, 杨武, 陈果

(1. 湖南科技大学 地理空间信息技术国家地方联合工程实验室, 湘潭 411201;

2. 湖南科技大学 地理空间信息湖南省工程实验室, 湘潭 411201)

摘要 随着信息获取技术的快速发展, 地理信息数据每天以TB级的数量增加。三维城市模型数据作为三维GIS的重要内容, 在数字城市和智慧城市建设中发挥重要作用。由于三维城市模型数据结构复杂, 其数据量具有海量性, 因此, 高效地对三维城市模型进行划分及存储, 以满足数据的长效管理及三维GIS系统的快速可视化数据调度和空间辅助决策需求, 成为近年的研究热点。以往的数据划分方法导致划分区域在数据调度中变化频繁, 使数据更新和管理变得困难, 需寻找一种更为稳定且具有普适性的数据划分方法。本文分析了现有三维城市模型数据划分方法的不足, 提出了基于拓扑关系模型的大比例尺图幅划分方法, 并对划分后三维模型数据进行统一命名编码; 借助非关系数据库MongoDB强大的海量数据组织及高效的多并发访问功能, 构建了MongoDB分片集群服务器; 对三维城市模型数据进行了单元划分, 并采用规则建模软件City Engine进行建模, 得到三维城市模型, 借助非关系数据库软件MongoDB进行数据存储实验。结果表明, 基于拓扑关系模型的大比例尺图幅划分方法适用于三维城市模型数据划分, 划分后数据的存储效率明显提高, MongoDB数据库的多并发访问效率具有良好的稳定性。

关键词 三维城市模型; 数据划分; 空间拓扑; 空间数据库; MongoDB

DOI:10.3724/SP.J.1047.2015.01442

1 引言

GIS在智慧城市的数据采集、建模、处理及分析等方面发挥重要作用。传统二维GIS必然向三维GIS发展, 这不仅是简单的二维到三维的转变^[1], 维数的扩展所承载和表达的信息呈几何倍数增长, 数据结构复杂, 数据组织和存储更加困难, 其所带来的复杂性远远超过预期。三维城市模型(Three Dimensional City Model, 3DCM)是指在计算机中以三维形式直观地描述城市地理实体^[2], 是城市直观三维可视化和三维空间分析的基础。其作为数字城市建设的一种重要框架数据, 是3DGIS的重要内容^[3], 近年已成为GIS领域的一个热门研究课题^[4]。对3DCM的研究主要集中在空间数据采集、三维模型重建、数据管理、可视化及空间分析等^[5]。

Stadler采用基于CityGML格式的Oracle Spatial 11g管理存储3DCM数据^[6], 但只支持简单的三维几何, 无法表达具有三维曲线或曲面的复杂三维

空间实体^[7]。Sozer基于面向对象数据库实现了时空数据的管理^[8]。兰小机等提出了GML空间数据在DB4O数据库下的管理和空间索引方法^[9]。上海市建立了基于ArcGIS平台的三维城市模型数据库管理系统, 该系统通过ArcSDE进行数据的格式转换和Oracle数据的I/O操作。冯琰等基于上海市三维城市模型管理系统, 将模型分类合并建立了5种细节模型以便组织整个三维场景^[10]。朱庆等提出顾及语义的三维空间数据库模型, 通过语义层、核心对象层、几何层三层架构实现了空间数据的地上地下一体化管理^[11]。陈超等提出了使用NOSQL存储地图瓦片数据的方法, 解决了传统关系数据应对多并发访问的瓶颈问题^[12]。龚俊等探讨了三维场景的三维R树扩展管理方法^[13-14]。

上述研究虽然取得了一定的成果, 但随着3DCM数据的高速增长和快速访问, 使关系数据库应对多并发访问的瓶颈矛盾日益突出。鉴此, 本文从数据划分出发, 采用大比例尺地形图图幅作为

收稿日期 2014-07-15; 修回日期: 2015-07-26.

基金项目 国家自然科学基金项目(41271390、41571374); 卫星测绘技术与应用国家测绘地理信息局重点实验室开放基金项目(KLAMTA-201406)。

作者简介 李朝奎(1967-), 男, 博士, 教授, 研究方向为三维GIS建模及应用。E-mail: chkl_hn@163.com

划分单元,以应对数据的频繁访问;通过空间拓扑关系模型解决三维模型的跨图幅分割问题;采用非关系数据库 MongoDB 对经上述处理的数据进行存储实验。实验表明:基于拓扑关系模型的大比例尺图幅划分方法,对三维城市模型数据划分及存储具有明显的效果,极大地提高了数据调度与访问效率。

2 三维城市模型数据划分

2.1 三维城市模型数据内容

3DCM 作为城市逼真的三维数字表示,具有虚拟现实技术的高度真实感、3D GIS 数据库管理技术和空间分析功能,许多大学和公司都积极开展了 3DCM 的研究和开发,因为 3DCM 具有复杂的数据类型和数据结构,在相当长一段时间内保持其特有的生命力。

2.1.1 三维数据模型

3DCM 数据较二维点、线、面更加复杂,需要三维数据模型来描述地理空间实体。目前比较常用三维数据模型主要包括:(1)面表示的数据模型;(2)体表示的数据模型;(3)面与体混合表示的数据模型。面模型侧重于地理空间实体表面信息的描述,优点是便于数据更新维护,但是空间分析功能较差,如边界表示模型(图1);体模型侧重于空间实体的体表示,适用于空间分析,缺点是数据量较大,对计算能力要求较高,如八叉树模型(图2);面与体的混合模型兼具二者的优点,改进了二者的不足,得到了最广泛的应用^[15]。

2.1.2 数据内容

三维城市模型数据内容包括城市地下管线、隧

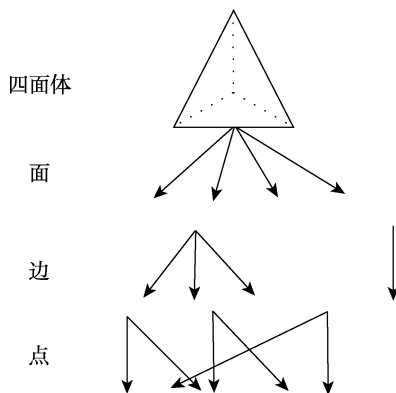


图1 B-REP模型分级图

Fig. 1 The hierarchical diagram of B-REP model

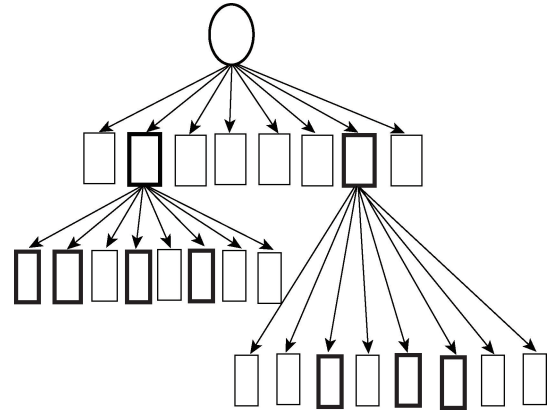


图2 八叉树(Octree)模型

Fig. 2 The Octree model

道、地表地形、水系、植被、建筑物、地上桥梁等。3DCM 数据结构可分为3层:几何模型层、应用模型层和组合关系模型层。而几何模型层又可抽象为:点实体、线实体、面实体和DEM(图3),并以此为基础,定义多边形、函数构造面、几何构造体(CSG)、TIN 面片等元素^[16]。

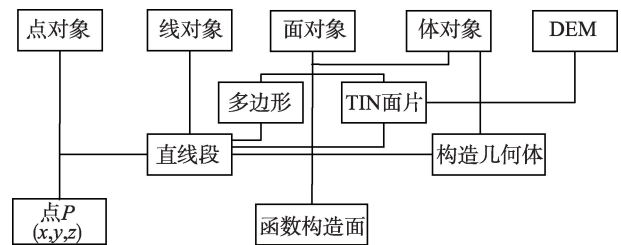


图3 3DCM几何空间数据模型

Fig. 3 The geometry spatial data model of 3DCM

2.2 三维城市模型建模单元划分

三维城市模型在空间上具有一定跨度。宏观上,三维城市模型可分为地下、地表、地上3个层次。现实生活中,三维城市模型分类非常复杂,考虑数据生产要求,根据《城市三维建模技术规范》(CJJ/T157-2010)中的模型分类标准,将模型分为:建筑模型、地形模型、水体模型、植被模型、管线模型、交通设施模型及其他模型7类,并对模型采用“建模单元编码-模型类型-模型序号”的3级12位编码规则进行编码。

对于3DCM的建模单元的划分,常用的方法有:以社区为基本建模单元;以规划地块为基本建模单元;以自行划分的多边形区域为基本建模单元等,以上几种划分方法都存在数据长效管理不便的缺点,主要原因是建模单元划分方法不统一,不能满足不同行业部门对数据的应用需求,导致数据调

度频繁变动建模单元。地形图具有统一的分幅规则,具有全世界范围内的普及性,并且长时间内不会有大的变动,所以,采用标准地形图作为建模单元组织三维模型数据有利于数据的长效管理。

通过对某地的1:1000图幅内的三维模型数据量进行统计分析(图4),结果显示:一幅1:1000的大比例尺地形图,实地面积为0.25 km²,其对应最大数据量的建筑模型平均数据量为34.68 M,对其划分4个子块比较合适,平均三角面片数为31 852个,最小数据量的水系模型平均数据量为0.07 M,三角面片数仅有315个,因此,该比例尺图幅最适合作为建模单元。

2.3 空间拓扑关系模型

以图幅为基本建模单元,对于二维三维数据的同步更新、三维模型数据的长效管理、大规模数据的共享操作都有很高的效率,但同时也带来了一些问题:图幅的划分,不可避免地带来了地物跨图幅现象。由于三维模型是在二维数据的基础上建模构成,所以会造成建筑、地形等三维模型在图幅中被切割,给后期数据的管理和应用带来极大的困难。

拓扑关系是指地理实体之间在缩放、旋转、平移等操作下维持不变的一种关系,在空间数据建模、数据更新及组织管理中具有重要作用。常用的拓扑关系模型有四交模型、九交模型、基于Voronoi图的九交模型、基于维数扩展的九交模型、Intersection模型等。在三维模型的建模过程中,建筑物模型要求分开建模,地形模型采用分块建模,小区域的广场、公园等作为整体建模;水系、道路等采用分段建模。针对模型跨图幅现象,根据建模流程及三维模型特点,建筑物模型、道路交叉口等代表性地



图4 基于2 km×2 km图幅的建模单元划分

Fig. 4 The modeling unit division based on 2 km×2 km map sheet

物不分割或重复建模。对于模型的跨图幅分割解决方案(图5),具体判断步骤如下:

(1)计算不可分割地物实体在图幅上的最小圆形包围域A;

(2)计算A所处的图幅号,并计算各图幅最小外界矩形域B_n(n=1、2、3、4),如果A与多个图幅具有拓扑关系,则各图幅根据自东向西,自南至北优先原则选择建模图幅,n的编号同样根据该原则;

(3)判断A与B_n拓扑关系,基本关系只有邻接、相交、包含和内切4种;

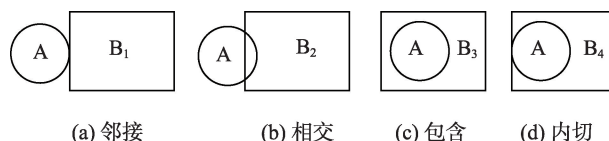


图5 地物与图幅的4种基本关系

Fig. 5 Four kinds of basic relationship between object and map

(4)第1种邻接关系,不需建模,重复步骤(3),判断A与B₂(B₃、B₄)拓扑关系,直到进入步骤(5)为止;

(5)第2种相交、第3种包含、第4种内切关系,则在图幅B₁中建模,其他图幅不再建模;

(6)结束。

通过以上拓扑关系判断,能有效地避免模型的分割建模和重复建模,这对3DCM的存储及应用都有极大的帮助。

3 基于MongoDB的三维城市模型存储方法

3.1 MongoDB数据库

MongoDB是10gen公司2009年研发的一个基于文件的分布式数据库,它具有Key-values存储方式的高性能和高扩展性。Mongo DB使用数据结构松散的Binary JSON格式,同时它支持复杂的数据结构,利用自动分片实现数据的存储,具有强大的查询功能,几乎支持关系数据库中所有单表查询功能,还可针对存储字段对数据进行索引,具有很好的高并发访问效率,非常适合解决海量数据的存储和多并发访问效率问题。主要特点:存储便捷、功能丰富、可扩展性强及高可靠性。

3.2 3DCM数据组织

本文对数据建模区域进行了1:1000大比例尺图幅的划分,对建模区域内的三维模型按照不同种

类建模。对于三维城市模型中的地形数据和影像数据,采用金字塔结构进行存储,即对地形和影像根据 1:1000 比例尺图幅划分作为叶子结点,然后用四叉树结构组织,树的每一层数据用单独文件夹存储。对于三维模型数据,采用分层分块的方法组织,即整个场景分为若干个子场景,每个子场景下面又有 7 种数据类型,每个数据类型下面对应多个三维模型。这样的组织方法使大规模的数据能够分片组织,避免了单个文件夹下面的超大数据量,使得数据的索引效率增高。本文经过对比分析,选择了非关系数据库产品 MongoDB 进行数据的存储,这种数据组织方法能很好地适应 MongoDB 数据库的分布式存储和数据分片特性(图 6)。

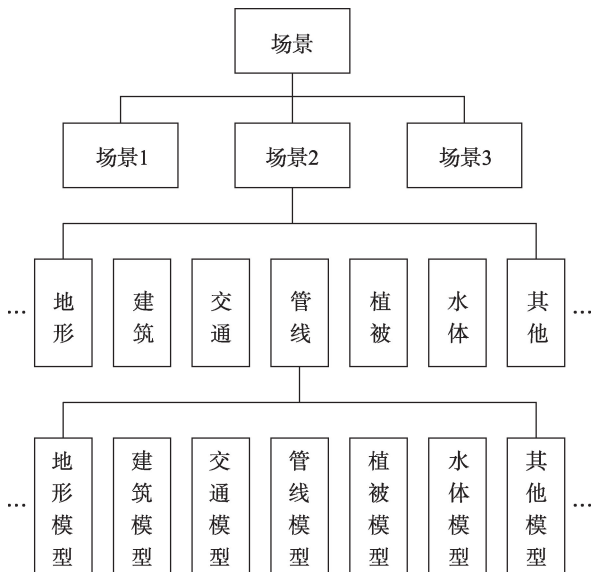


图6 三维模型组织结构

Fig. 6 The organization chart of 3D model

3.3 基于 MongoDB 的 3DCM 存储组织方法

三维城市模型对于数据库的选择,要求其能存储大规模海量数据、能快速地入库和高效地检索、能应对高并发访问和实时更新,而 MongoDB 数据库能出色地满足这些要求。MongoDB 支持的数据结构是松散的 BSON 格式,与二进制 JSON 类似,数据以 Key-Values 键值对进行存储,键作为索引,可以是文件名字或是容易区分的编号,其值对应的就是模型的二进制数据,即存储数据格式为<NameID,Data>。数据存储时既要避免大数据整块存储导致的数据检索和读取困难,也要避免小数据单独存储产生的数据压缩负担。对于影像数据、地形数据,以四叉树结构组织,每一个单元对应四叉树的一个层,实现数据的分层表达。对于三维模型数

据,按照不同子场景分开存储,同时每一级下面再按照模型类型分级存储,这样便于数据的分片存储及数据节点的扩充,为了方便管理,把不同类型的数据存储为不同的集合,这样每个集合下面的数据类型保持一致,便于数据的索引及管理。

MongoDB 数据库自带分片技术,将数据集分割为多个子集,每个子集存储在一个分片服务器上,在客户端和数据服务器中间通过 Mongos 路由进程通信响应数据的发布请求,接收并返回客户端,当数据增加时只需增加分片服务器。分片集群包括:分片服务器、配置服务器和路由服务器,分片服务器包括数据复制集;配置服务器记录分片数据定位记录,数据查询时直接定位数据所在分片,提高检索效率;路由服务器负责接收客户端请求并接收返回结果。

为快速定位数据所在数据库中的位置,对数据建立索引机制,MongoDB 的索引机制同关系型数据,支持多级索引,本研究将数据类型定义为一级索引,主要是确定请求数据所在集合,对于集合内部,使用 ID 索引即文件名字索引,通过该索引结构,实现数据的快速定位。创建索引代码如下:db.products.ensureIndex({ "集合:collection": 1, "名字:nameID": 1, })。

4 实验及结果分析

4.1 实验方法及数据

本文实验数据采用某地二维 GIS 矢量数据,利用 City Engine 建立三维城市模型,建模流程为:(1)数据准备阶段,包括二维 GIS 矢量数据、高分辨率影像数据;(2)对二维数据进行图幅划分、转换为三维格式预处理,影像数据主要用于建筑物屋顶的纹理贴图;(3)编写建筑物生成的规则文件;(4)导入经过处理的数据,利用相关规则生成模型,并利用遥感影像进行屋顶贴图;(5)最终生成模型,导出模型。建模流程如图 7 所示,建模结果如图 8 所示。

针对同一区域,分别进行原始建模和采用本文划分方法建模,2 种方法建立模型数据量分别为:11.7 GB 和 12.1 GB。

4.2 实验环境

采用普通 PC 机作为数据服务器,由于 MongoDB 数据库在 32 位系统上最多支持 2.5 GB 的数据量,且计算机最大内存为 4 GB,而 MongoDB 数据

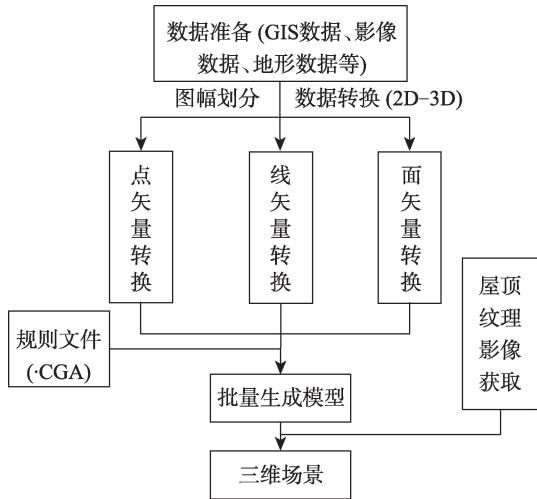


图7 基于规则的CityEngine建模流程

Fig. 7 Rule-based modeling procedure of CityEngine



图8 构建的三维城市模型

Fig. 8 The construction achievements of 3DCM

库又依赖于内存映射实现高性能数据读写,64位系统则没有这些限制,因此,本文选择64位操作系统来构建MongoDB集群服务器。

硬件环境:1台华硕笔记本,处理器: Intel(R) Core(TM) i5-2410 CPU 2.3GHZ,内存2.0 GB,硬盘500 GB,显卡: Nvidia GeForce GT 520M (1GB),操作系统: windows7 64位。2台联想台式机,处理器: Intel(R) Core(TM) i3-2120 CPU 3.3GHZ,内存2 GB,硬盘500 GB,操作系统: windows7 64位。

软件环境: MongoDB 版本: mongodb-win32-x86_64-2008plus-2.4.8。

网络环境: 使用千兆以太网交换机连接各台服务器。每台服务器固定唯一的IP,分别有一个路由服务器,端口为10 000;一个配置服务器,端口为20 000;并配置2个分片。集群服务器配置如表1所示。

4.3 结果及分析

根据本文研究方法分别设计了2个实验验证:
(1) 遥感影像数据的MongoDB数据库入库时间和

表1 MongoDB 集群环境

Tab. 1 The cluster environment of MongoDB

服务器编号/IP	路由服务器端口	配置服务器端口	分片/端口
1:192.168.0.1	Mongos1:10000	Configdb1:20 000	Shard1:27 001 Shard2:27 002
2:192.168.0.2	Mongos2:10000	Configdb2:20 000	Shard1:27 001 Shard2:27 002
3:192.168.0.3	Mongos3:10000	Configdb3:20 000	Shard1:27 001 Shard2:27 002

SQL Server 数据库的数据入库时间和模拟多并发访问效率的对比实验;(2)采用本文方法划分的三维模型数据入库时间和不划分的三维模型数据入库时间与多并发访问效率对比实验。实验环境中用一台华硕笔记本和2台联想电脑,华硕笔记本作为数据服务器,使用2台联想台式机作为客户机,通过自己编制的模拟软件模拟多并发访问,来测试2种数据组织方式的访问效率。2台客户机配置及环境一致,所以实验结果可靠。本文所指的查询时间,是指2台客户机在模拟多并发访问时加载实验区域所有数据的访问时间。因为实验区域数据的空间分布并不均匀,导致加载部分区域数据难免因为数据分布不均而有所偏差,所以本文采用加载实验区全部数据的方案,对实验结果进行分析,验证本文方法的可行性和效果。

(1) 遥感影像存储实验

本实验采用实验区内1 m分辨率的影像数据,其坐标对按照1:1000比例尺图幅进行划分,获得256幅影像,总数据量大小为4217 MB。分别对这些数据使用MongoDB存储方案和SQL Server 2005进行存储,并对其效果进行对比分析。

在SQL Server 2005数据库中,同一级的影像分块存储为一张表,每一块影像单独建立索引,使用java连接代码如下: String conUrl="jdbc:microsoft:SQL Server://localhost:1433;DatabaseName=dbname" //连接字符串; Connection con==DriverManager. getConnection(conUrl); //生成连接。

数据入库时间对比如表2所示,从中可看出: MongoDB数据库入库时间明显低于SQL Server 2005,且随着数据量的增大,入库时间差距更加明显,因此本文存储方案具有一定的优越性。

2种数据库多并发访问平均时间对比如图8所示,随着并发数的增加,SQL Server 2005数据库平均访问时间急剧增加,而MongoDB数据库由于具有数据缓存功能,随着并发次数的增加平均时间基

表2 影像数据入库时间对比

Tab. 2 The comparison of image data storage time

数据量(MB)	时间(s)	
	Mongo DB	SQL Server 2005
500	98.231	173.962
2000	327.917	631.674
4217	745.483	1752.790

本没有变化,甚至命中缓存数据较多时,平均访问时间还有可能降低,因此,MongoDB数据库对多并发访问具有更高的效率。

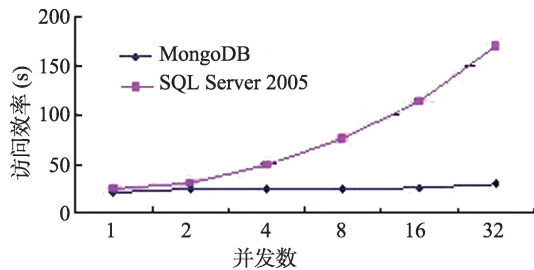


图9 模拟多并发访问时间效率对比

Fig. 9 The time efficiency comparison chart of simulating multiple concurrent access

(2) 三维模型存储实验

对于本文所构建的三维城市模型数据,为了验证数据划分方法和模型简化算法的有效性,本文对:模型①没有采用划分方法划分的三维城市模型;模型②以本文方法划分的三维城市模型;模型③简化以后的三维城市模型,做对比实验,主要从数据入库时间和应对多并发访问时间效率2个方面进行对比分析。

数据入库时间对比如表3所示,由表可看出模型①和②入库时间几无差别,由于划分产生多个数据分块,在入库时会额外产生一定的时间消耗,模型②的入库时间比①稍高;模型③由于构建4个分辨率的LOD模型,导致数据量成倍增加,数据入库时间明显高于①和②。

多并发访问效率对比图如图9所示,由图可看出,模型①、②、③应对多并发访问效率逐步提高,这是由于模型②的数据划分使得访问数据不必加载整块数据,只需要加载所访问的分片数据即可,所以模型②的效率明显高于①;而模型③由于对模型建立了多分辨率LOD模型,可以根据实际情况选择合适分辨率的简化模型,这样所需要加载数据量就会明显降低,所以其访问效率显著高于模型①和②。

从实验1结果以看出,由于MongoDB数据库是

表3 不同处理方法数据入库时间对比

Tab. 3 The data storage time comparison of different methods

模型种类	数据大小(GB)	入库时间(s)
①	11.7	1257.695
②	12.1	1309.553
③	20.5	2478.391

分布式存储,且数据库内部没有严格的表结构,其数据入库时间和应对多并发访问效率方面均远远优于SQL Server 2005数据库,证明非关系数据库的优势。对实验2,由于本文对模型进行划分并建立多分辨率模型,入库时间有一定的增加,但区别不明显,但是应对多并发访问时效率远远高于上面2种方法,也就是说通过合理的数据冗余,能换来高效的数据访问效率。同时,从图9曲线变化可看出,随着并发数的增加,MongoDB数据库访问效率基本保持一致,展示多并发访问效率具有良好的稳定性。

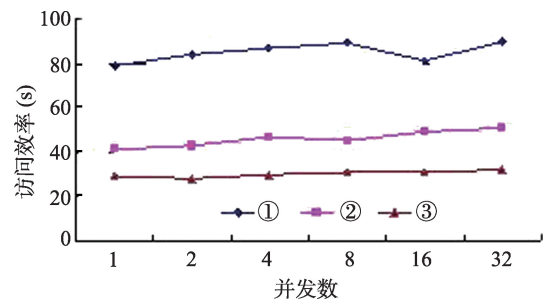


图10 不同处理方法的数据多并发访问效率对比

Fig. 10 The comparison of efficiencies for multiple concurrent access with different processing methods

5 结论

通过对三维城市模型数据特性和数据内容的分析,总结了以往数据组织方法存在的不足,在此基础上提出了基于拓扑关系的大比例尺地形图单元划分及建模方法。利用该方法对实验数据进行单元划分,对划分单元采用规则建模软件City Engine进行建模得到三维城市模型,借助非关系数据库软件MongoDB进行数据存储实验。实验表明,本文提出的方法具有较高的存取效率。提出的空间拓扑关系模型仅局限于解决模型分割建模问题,而数据库内部的数据结构是今后需要深入研究的课题。

参考文献:

[1] Zhu Q, Li D, Zhang Y, et al. Integration of DEMs, imag-

- es, and 3D models[J]. *Photogrammetric Engineering and Remote Sensing*, 2002,68(4):361-367.
- [2] 陈超益. 基于图幅的城市三维模型(3DCM)数据管理模式研究[D]. 杭州: 浙江大学, 2012.
- [3] 冯琰, 郭容寰, 汪旻琦, 等. 三维城市模型数据组织与管理方法研究[J]. *测绘科学*, 2011(1):215-217.
- [4] 孙敏, 马嵩乃, 陈军. 三维城市模型的研究现状评述[J]. *遥感学报*, 2002(2):155-160, 168.
- [5] Gruen A, Xinhua W. Creation of a 3D city model of Zurich with CC-M odeler[M]. //In: Deren Li. *Spatial Information Science Technology and its application RS, GPS, GIS, Their Integration and Application*. Wuhan: Wuhan University Press, 1998.
- [6] Stadler A, Nagel C, KLnig G, *et al.* Making inter-operability persistent: A 3D Geo- Database based on CityGML [C]. *The 3rd International Workshop on 3D Geo-Information*, Seoul, Korea, 2008.
- [7] Kothuri R, Godfrind A, Beinat E. *Pro Oracle spatial for Oracle database 11g*[M]. New York: Apress, 2007.
- [8] Soger A. Design and implementation of spatiotemporal databases[D]. Ankara: Middle East Technical University, 2010.
- [9] 兰小机, 苏健强, 张卫国. DB4O引擎下的GML空间数据存储研究[J]. *测绘科学*, 2010,35(3):146-148.
- [10] 冯琰, 郭容寰, 汪星琦, 等. 三维城市模型数据组织与管理方法研究[J]. *测绘科学*, 2011,36(1):145-149.
- [11] 朱庆, 李晓明, 张叶廷. 一种高效的三维GIS数据库引擎设计与实现[J]. *武汉大学学报(信息科学版)*, 2011(2):127-132, 139.
- [12] 陈超, 王亮, 闫浩文. 一种基于NoSQL的地图瓦片数据存储技术[J]. *测绘科学*, 2013(1):142-143, 159.
- [13] 龚俊, 朱庆, 张叶廷, 等. 顾及多细节层次的三维R树索引扩展方法[J]. *测绘学报*, 2011,40(2):249-255.
- [14] 龚俊, 朱庆, 章汉武, 等. 基于R树索引的三维场景细节层次自适应控制方法[J]. *测绘学报*, 2011,40(4):531-534.
- [15] 李拥. 基于OpenMP的三维城市模型并行绘制研究[D]. 湘潭: 湖南科技大学, 2013.
- [16] 李朝奎, 张云珍. 基于几何3DCM的日照分析模型及其辅助空间决策支持[J]. *湖南科技大学学报(自然科学版)*, 2005(1):52-55.

Research on Three Dimensional City Model Data Partitioning and Distributed Storage

LI Chaokui*, YAN Wenying, YIN Zhihui and CHEN Guo

(1. *National-Local Joint Engineering Laboratory of Geo-Spatial Information Technology, Hunan University of Science and Technology, Xiangtan 411201, China*; 2. *Hunan Province Engineering Laboratory of Geospatial Information, Hunan University of Science and technology, Xiangtan 411201, China*)

Abstract: With the rapid development of information acquisition technology, the geographic information data is increasing at the magnitude of terabyte every day. As an important content of 3D GIS, 3D city model data plays an important role in the construction of digital city and smart city. Because the data structure of 3D city model is complex and the data volume is huge, how to efficiently divide and store large amount of 3D city model data in order to meet the long-term management of data, the rapid visualization of data scheduling and the requirement of spatial assistant decision-making of 3D GIS system, has become a research hotspot in recent years. Previous data partitioning methods have caused the changes of zoning frequently in the data scheduling, which makes the update and management of data become more difficult. So, it is necessary to find out a more stable and universal data partitioning method. In this paper, based on the research of the shortcomings for the existing 3D city model data partitioning methods, we proposed the large scale map partition method based on topology relation model, and then we designed a unified name encoding scheme for the 3D models data after splitting. With the help of the powerful massive data organization and efficient multiple concurrent access function of the non-relational database MongoDB, a MongoDB sharded cluster server is constructed. The 3D city model data was used in unit division, and the rules modeling software City Engine was applied to processing the divided units, thus producing the 3D city model. Afterwards, MongoDB was used for data storage experiments. The results show that the large scale map partition method based on topology relation model is capable and suitable for the data partition of 3D city model, and the storage efficiency of the divided data is obviously improved. Moreover, the MongoDB database has a good stability on multiple concurrent access.

Key words: 3DCM; data partition; spatial topological relations; spatial database; MongoDB

*Corresponding author: LI Chaokui, E-mail: chkl_hn@163.com