

基于空间聚类的矢量空间数据并行计算划分方法^①

邱 强^{②***} 方 雷^{***} 姚 晓^{***} 方金云^{*}

(^{*} 中国科学院计算技术研究所 北京 100190)

(^{**} 中国科学院大学 北京 100190)

(^{***} 香港理工大学 香港)

摘要 为了解决并行矢量空间分析在数据划分阶段的负载均衡问题,研究了矢量空间数据的划分,提出了一种基于空间聚类思想的矢量空间数据划分方法。该方法充分考虑矢量空间数据规模以及空间邻近性特征对并行空间分析算法效率的影响,首先采用空间填充曲线对二维空间数据进行编码,保证空间要素邻近性特征;然后用空间要素集合对空间要素流进行填充,从而确保各个子任务集中的要素数据规模相对均衡。以并行叠加分析中点面、线面、面面叠加操作为例,设计了对比实验。实验结果表明,该方法能够有效提高以线、面要素为操作对象的并行算法负载均衡度和提高并行算法整体运行效率。

关键词 并行计算, Hilbert 曲线, 矢量数据, 负载均衡, 要素集合

0 引言

空间分析是地理信息系统(GIS)的重要组成部分,空间分析对数据的处理能力直接决定了 GIS 的整体服务性能。近年来,随着空间数据获取方式的发展,空间数据呈现出规模海量化、结构复杂化、服务实时化的特征,提高传统空间分析算法效率的并行计算技术成为主流的技术手段。负载均衡问题是影响并行算法效率的关键因素。由于矢量空间数据的数据特征,负载不均问题更为突出,因此需要研究和设计更为合理的矢量空间数据划分策略,以便在保持空间数据邻近性特征的前提下提高并行算法的负载均衡度。目前研究工作中,矢量空间数据的划分(partitioning)主要是针对空间查询操作^[1, 2]。相比于空间查询,空间分析算法具有全局计算的特点,即输入的空间数据需要全部读入内存参与计算,输出数据受输入数据的全部要素影响。在并行空间分

析方面,Zhou 等^[3, 4]针对分布式空间数据处理过程结合空间数据的位置关系,基于 Hilbert 曲线提出了一种数据划分策略,该方法在保证数据规模平衡的前提下保证了一定的空间邻近性。相关研究还包括 Chen 等^[5]提出的基于最近距离原则的空间数据划分策略等。

矢量空间分析的并行调度策略有静态调度和动态调度两种,本文作者曾对这两种调度策略的空间分析并行算法进行了对比研究。实验表明,动态调度的并行矢量空间分析算法没有明显优势^[6]。因此,本研究针对静态调度,分析了矢量数据空间邻近性以及数据规模等特征,设计实现了更为合理的数据划分策略。本研究的主要贡献在于利用空间填充曲线和空间要素集思想,解决了空间数据在数据划分时保持空间邻近性和计算任务均衡性等问题,设计并实现了基于该数据划分思想的并行空间分析算法,从而提高了算法的并行效率和扩展性,为实现高性能并行空间分析算法提供了技术保证。

^① 863 计划(2011AA120302)和国家青年科学基金(41301423)资助项目。

^② 男,1987 年生,博士生;研究方向:GIS 理论与方法,并行矢量空间分析,并行计算技术等;联系人,E-mail: qiuqiang@ict.ac.cn
(收稿日期:2014-10-23)

1 矢量空间数据并行计算划分原则

1.1 计算任务负载均衡

数据划分后,各个子任务需保持良好的负载均衡度。负载均衡度是影响并行算法并行效率的重要指标。以 \overline{time} 表示各个计算任务平均使用时间, $time_{max}$ 表示最大的计算任务使用时间,则负载均衡度 $b(n)$ 可以表示为:

$$b(n) = \frac{\overline{time}}{time_{max}} \quad (1)$$

为提高负载均衡度,应尽量保持各个子任务的计算时间相对均衡。对于点数据,由于每一个点要素的几何数据大小基本相同,因此将点数据按要素数量分配给各个计算任务可以基本保持负载均衡。在线数据和面数据中,由于每个线要素或面要素包含的数据不尽相同,因此按照要素数目划分难以确保计算任务的均衡性。实验结果表明,采用并行空间分析算法,子任务的计算时间与空间数据的大小呈正相关关系,因此在数据划分阶段,应保持各个子任务的数据规模相对均衡。

1.2 空间邻近性

空间要素在空间分布上具有越邻近越相关的性质。例如,在并行缓冲区分析中,数据划分时保持空间要素的邻近性可以保证相邻要素在同一个计算任

务中,从而减少缓冲区的构建范围。又如叠加分析中,叠加图层的空间要素相对集中在同一计算任务中可以减小该任务内图层数据的外包,进而在与被叠加图层进行外包过滤时大大降低计算范围。因此,在数据划分阶段保持空间邻近性特征可有效提高子任务处理效率。

1.3 空间要素一致性

空间要素在划分过程中要保证各个子集的要素无重复,且计算结束后保持数据完整性和一致性。在空间数据进行划分时,空间要素包含完整的几何信息和属性信息,是数据划分的最小独立单元。划分算法需保证所有空间要素均被划分到各个计算任务中,且同一要素不会被重复划分到不同的计算任务中。否则,被重复分配的空间要素会带来冗余计算时间,影响并行效率。又因为浮点计算等误差因素,对算法正确率造成不良影响。

2 矢量空间数据划分方法

目前针对矢量空间数据的划分方法主要包括顺序划分和随机划分、空间索引、空间聚集和空间统计聚类等。这几类方法在负载均衡度、空间邻近性特征以及算法复杂度方面各有特点,表 1 给出了这几类主流方法的对比。

表 1 四类空间数据划分方法对比

	顺序和随机	空间索引	空间聚集	统计聚类
负载均衡度	不均	均匀	均匀	均匀
空间邻近性	最低	好	较好	好
要素一致性	一致	不一致	一致	一致
算法复杂度	最低	较好	较好	最高
综合性比较	由于空间索引主要应用于空间查询操作,不能保持在并行计算中各个计算任务的要素一致性,而统计聚类的方法算法复杂度较高,对空间数据的预处理过程时间较长,会影响并行算法的效率;顺序和随机划分方法实现最为简单,数据预处理过程对并行算法整体效率的影响最低,是目前采用的主流划分方法,但该方法不能保证空间数据的邻近性特征以及负载均衡问题;空间聚集方法能够保证空间数据较好的邻近性特征和要素一致性,且算法效率较高			

并行空间分析算法采用 fork-join 的并行模式。其中,数据划分操作属于预处理阶段,因此,该操作的时间代价将对并行算法效率造成影响。在选用数据划分方法时需确保该划分策略对并行算法的整体运行时间优化程度远大于引入该方法造成的时间代价。综合考虑各类数据划分方法特征,本文采用空间聚集的思想进行研究。

2.1 空间要素位置的表达

矢量空间要素主要包含点、线、面等类型。如前所述,在并行计算任务中,空间要素要保持一致性特征,因此需要唯一的位置信息来表达空间要素的位置。在不同类型的空间要素中,点要素的表达相对简单,以点要素的(x, y)坐标即可确定该要素的位置。针对线要素以及面要素,精确表达要素位置的代价较大,因此,在并行效率优先的前提下可以采用特征值近似表达线、面要素的位置。常用方法包括最小包围矩形(minimum bounding rectangle, MBR)、凸包等。为尽量减小附加操作对并行算法效率的影响,方便线、面要素进行空间填充曲线编码,本文针对线、面要素,采用MBR中心点近似表达要素的位置信息,如图1所示。

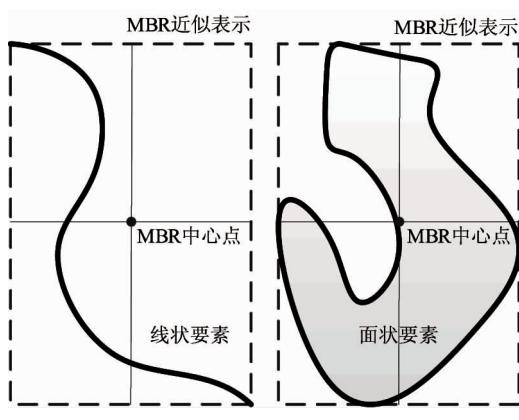


图1 线、面要素位置的近似表达

2.2 空间聚类方法

如前文所述,本文选用空间聚集的方法进行研究。在空间聚集方法中,主要操作为利用空间填充曲线对二维空间数据进行编码,从而在确保空间邻近性原则的前提下,得到一维状态的空间要素流。常见的空间填充曲线包括 Hilbert 曲线、Z 曲线以及 Gray 曲线等。其中,Hilbert 曲线对二维空间数据的

编码效果最佳,Z 曲线的编码实现最为简单,但 Z 曲线对空间要素邻近性的保持效果不佳。在考虑编码效率与编码效果的情况下,本文选用 Hilbert 曲线进行研究。

Hilbert 曲线是由德国科学家 David Hilbert 构建发明的,其构建过程是首先确定一个原子,再利用递归思想执行产生式规则,从而实现对二维空间对象进行逐步分解的过程^[7]。其编码规则可以表示为

$$(L \rightarrow +RF -LFL -FR +, R \rightarrow -LF +RFR +FL -) \quad (2)$$

其中,F 为指定原子,“+”表示顺时针旋转一个栅格单位,“-”表示逆时针旋转一个栅格单位。理想状态下的 Hilbert 编码为每一个 Hilbert 划分域中只包含一个空间要素。但是,在真实地理数据中,需要进行数据划分的空间数据的规模较大,且空间要素在位置分布上不均匀。例如,在我国行政区域数据中,东部沿海地区行政区域较多,每个行政区域面积较小;而西部地区行政区域较少,每个行政区域的面积较大。在这种情况下,进行 Hilbert 编码的复杂度较高,其算法效率会影响并行算法的整体效率。为提高编码效率,在数据划分过程中,Hilbert 格网数量不必严格匹配空间要素的数量,只需设定每个格网中包含空间要素的阈值即可。本文采用一种层次分解的 Hilbert 编码规则进行编码^[8],其算法思想是首先将图层范围进行粗网格划分,然后根据划分网格内空间要素的数量如果大于该网格能容纳要素数量的阈值,则对该网格进行四叉树划分,并计算子网格内的 Hilbert 编码,递归实现以上过程,从而实现以空间要素分布为依据的 Hilbert 编码。利用 Hilbert 格网对空间要素进行编码的算法过程可以描述如下:

Hilbert 划分方法主函数

```
DataPartitionByHilbert(input, output, PartsNum, jeNum)
    ReadLayerAtPath(layerpath);
    HilbertMap(I,j,ROW);
    for(i=0;i<FeatureCount;i++)
        //获取空间要素所在网格位置
        Env = getEnvelope();
        middleX = Env.minX() + Env.width()/2;
        middleY = Env.minY() + Env.height()/2;
        Cells[middleX][middleY].push_back(feature(i));
```

3 并行矢量空间分析算法实现

并行空间分析算法可分为数据预处理、并行计算和结果归并三个阶段。数据预处理阶段包括矢量

空间数据的划分和计算任务的分配。如图 2 所示，在对空间数据编码后，会得到一串空间要素的 Hilbert 流。如前文所述，对于点要素而言，每个点要素的几何数据大小基本相同。针对线、面要素，数据大小受该要素中所包含点的数量影响，因此数据大小

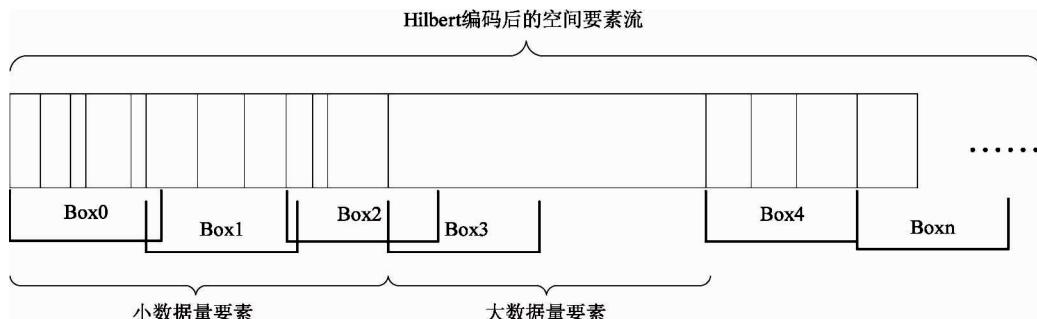


图 2 线、面要素数据划分规则

不尽相同。特别是在真实地理数据中，不同线、面要素的大小差异非常明显。为了以数据规模为均衡划分的依据，本文采用一种空间要素集合的结构体（Box）对按照 Hilbert 编码排序的空间要素进行填充。每个 Box 容器固定大小且容量相同，当要素填满该容器时转入下一个容器的填充。在计算任务分配时，将 Box 容器中的空间要素 ID 分配给各个计算任务，从而实现子任务中空间数据大小相对均衡，其实现过程如下：

空间要素集合填充函数

```
for(i = 0; i <= totalcells; ++i)
    if( curSize + totSize[rol][col] <= box.size || curPartNum == numParts-1)
        curSize += totSize[rol][col];
        //当未填满 box 容器时插入要素
        feat.insert(cells.begin(), cells.end());
    else
        SendGIDdata(IDarray, size); //发送 ID 数组
```

随后，各个子进程获取主进程的调度任务，分别为需要操作的要素 ID 集合，通过集合中的要素 ID 分别读取所需要素，进行独立的计算过程。在运算结束后，进程间将以先计算结束先归并的方式进行两两归并^[9]，最终输出得到并行计算结果。

4 空间数据划分实验与分析

实验分为两组，实验一采用真实地理数据测试，验证本文数据划分方法对不同类型、数据规模和操作类型的并行空间分析算法性能的提升效果。实验二采用大规模构造数据验证基于本文数据划分方法所实现的并行空间分析算法在高性能集群中的并行计算可扩展性。

4.1 实验环境

硬件环境：CPU 为 Intel Xeon，主频为 3.4 GHz，共 8 核；内存为 4GB DDR3 1333MHz ECC；外存为 500GB 7200RPM SATA 硬盘；采用千兆以太网连接各个节点。该配置服务器共 8 台，共计 64 核。

软件环境：操作系统为 64 位 CentOS 专业版；MPI 采用 OpenMPI 版本；集成开发环境为 Eclipse + CDT；采用 GCC 作为编译器。

4.2 实验数据

实验数据如表 2 所示，将采用真实地理数据和真实数据经过抽吸、复制、位移构造的构造数据组成，分别验证不同类型和数据规模的操作下本文数据划分方法对并行算法的优化效果。

表 2 实验数据说明

数据名称	数据类型	要素个数	数据说明	操作说明
ChinaResidences25m_50000.shp	点	565772	中国城镇居民点抽吸数据	实验一输入图层
rivers_400000.shp	线	415130	全国河流抽吸数据	实验一输入图层
ict_landuse.shp	面	122552	全国土地利用	实验一输入图层
ict_counties.shp	面	2449	中国县界行政图	实验一叠加图层
landuseoutput_8_8.shp	面	1048575	以土地利用数据为源数据的构造数据	实验二输入图层
Courtryoutput_8_8.shp	面	50816	以行政图为源数据的构造数据	实验二叠加图层

4.3 实验结果和分析

实验一分为 3 组,分别选用点面叠加求交、线面叠加过滤和面面叠加裁剪 3 种并行算法进行对比测试。该类并行算法原有空间数据划分策略为顺序划

分,在采用本文空间数据划分方法后,在 1 到 8 核的测试环境中进行对比,算法整体计算时间对比如图 3 所示。

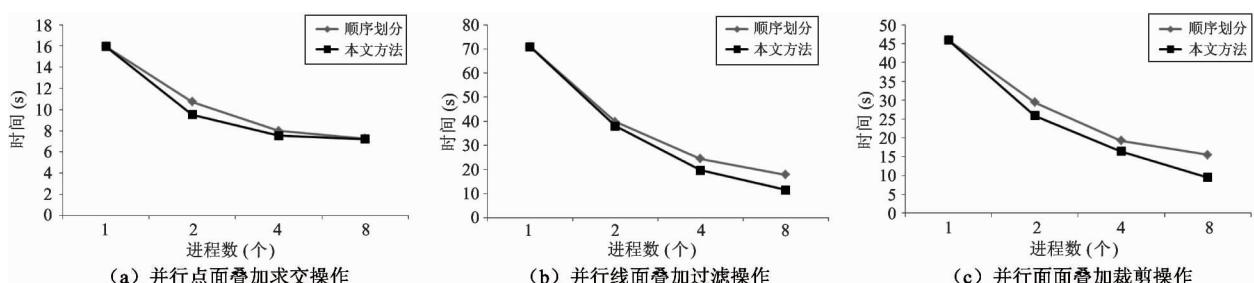


图 3 点、线、面不同数据类型在并行空间分析算法中运行时间对比

在点面叠加实验中(图 3(a)),随着进程数目的增多,采用本文数据划分策略的算法体现出了一定的加速效果,但是在 8 进程状态下,本文方法与原算法相比没有明显时间优势。由此可知,针对点数据,本文数据划分方法能够提高数据的空间邻近性,提高子进程运算速度。但是,如前文所述,点要素的几何大小基本相同,本文方法相比于顺序划分没有取得负载均衡度上的优势。在 8 进程状态下,本文数据划分的预处理代价和对算法时间优化程度基本持平,因此没有体现出较好的整体加速效果。

在线面、面面叠加实验中(图 3(b)(c)),随着进程数目的增加,采用本文数据划分策略的并行算法取得了更快的计算速度。并且进程数目越多,算法优化效果越明显。在面面叠加(图 3(c))8 进程状态下,原算法运行时间为 15.54s,采用本文方法后的计算时间为 9.52s。为验证本文方法的优化效果,本文详细对比了 8 进程状态下该算法空间数据划分效果(图 4,图 5),并对各个子进程计算时间进

行了对比分析(图 6)。

图 4 和图 5 为全国土地利用数据采取顺序划分和本文划分方法的划分效果对比。其中,不同颜色的数据区域表示被分配到不同进程中的空间要素。相比于顺序划分方法,在使用基于空间填充曲线的划分方法后,同一计算子任务中的空间要素在位置关系上更为集中,因此子任务在进行空间分析计算



图 4 顺序划分的效果



图 5 利用 Hilbert 曲线划分的效果

时需要处理的空间要素范围更小,子任务效率更高。另外,经对比,本文方法中各个子任务所包含要素的数据规模更加均衡,有利于提高并行算法的负载均衡度。

由图 6 实验结果可见,叠加分析多边形裁剪运算的并行算法在使用顺序划分策略时,各个进程运算时间从 4.59s 到 9.51s 不等。在使用本文基于 Hilbert 的划分策略后,各个进程平均计算时间由 7.6s 下降到 6.5s。并行算法的负载均衡度由 77.06% 上升到 82.71%。由此可见采用本文方法不仅各个计算进程的运算时间更加均衡,且平均计算时间有了明显优化。

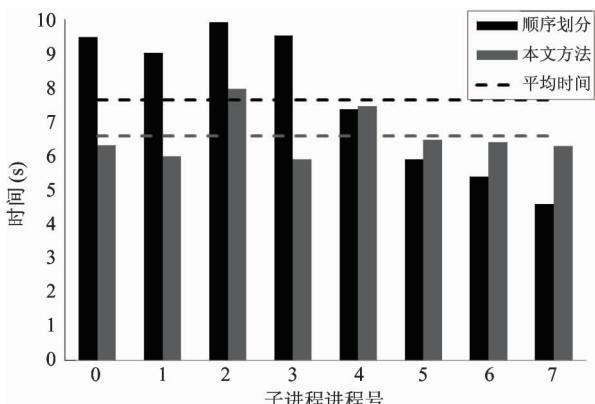


图 6 顺序划分与本文划分方法子进程计算时间对比

实验二通过真实地理数据进行图层平移、复制,构造了数据规模在百万级别以上的面数据,选用叠加分析中面面叠加求交操作进行实验,验证算法的扩展性。该数据集在已有商业 GIS 软件中无法完成分析任务。测试结果如图 7 所示,串行状态下该算

法需要执行近 800s,随着进程数目倍数增长,并行算法呈现出良好的并行加速效果,最多可以扩展到 128 核状态。但是,算法的并行效率逐步下降,计算资源的利用率逐步降低。在计算内核数目多于 16 核时,算法 IO 时间明显高于并行算法计算时间,逐渐成为性能瓶颈。这体现出矢量空间分析算法具有计算密集型和 IO 密集型的双重特征,本文研究内容体现在对并行算法计算阶段的优化,针对空间数据 IO 性能的进一步优化将作为未来研究工作的重点。

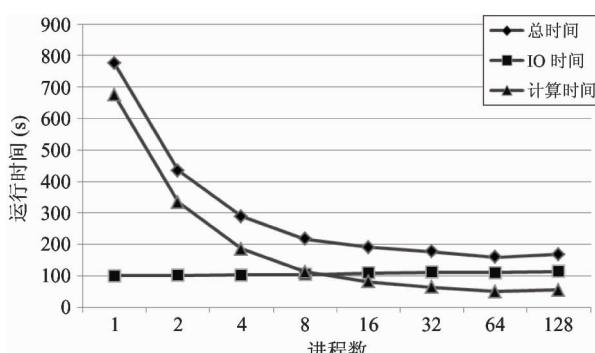


图 7 大规模数据算法扩展性测试

5 结 论

在矢量空间数据划分中,影响并行算法负载均衡的两个因素分别为空间数据邻近性特征和空间数据规模。本文提出了一种基于 Hilbert 填充曲线的数据划分方法,在并行算法的数据预处理阶段对数据进行划分,保持空间数据邻近性特征和数据规模相对均衡。实验证明,本文空间数据划分方法对以线、面类型的空间数据为操作对象的并行算法能够有效降低并行任务中各个进程的计算时间,且有效提高负载均衡度,进而得到算法优化的效果。

在大规模数据测试中,随着计算内核数目的增多,该并行算法能够获得持续的加速效果。但是,该算法对计算资源的利用率较低,IO 瓶颈较为明显。在此状态下,以内存计算等技术手段提高空间数据访问性能,降低并行算法 IO 时间将是进一步提高并行空间分析算法效率和扩展性的重要工作。

参 考 文 献

- [1] Meng L, Huang C, Zhao C, et al. An improved Hilbert

- curve for parallel spatial data partitioning. *Geo-spatial Information Science*, 2007, 10(4): 282-6
- [2] Yan B Q, Rhodes P J. IDEA——An API for parallel computing with large spatial datasets. In: Proceedings of the 2011 International Conference on Parallel Processing, Taipei, China, 2011. 355-364
- [3] Zhou Y, Zhu Q, Zhang Y. GIS spatial data partitioning method for distributed data processing. In: Proceedings of the International Symposium on Multispectral Image Processing and Pattern Recognition, F, 2007. International Society for Optics and Photonics. doi: 10.1117/12.739790
- [4] Zhou Y, Jiang L. Hilbert curve based spatial data declustering method for parallel spatial database. In: Proceedings of the 2nd International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE), Nanjing, China, 2012. 1-4
- [5] Chen Z, Wu L, Zhang D. Spatial data partitioning based on the clustering of minimum distance criterion. In: Proceedings of the International Conference on Computer Science and Service System (CSSS), Nanjing, China, 2011. 2583-2586
- [6] 邱强, 曹磊, 方金云. 并行点面叠加算法在动态调度和静态调度中的对比研究. 地理与地理信息科学, 2013, 29(4): 35-8
- [7] Chen H L, Chang Y I. All-nearest-neighbors finding based on the Hilbert curve. *Expert Systems with Applications*, 2011, 38(6): 7462-75
- [8] 陆锋, 周成虎. 一种基于空间层交分解的 Hilbert 码生成算法. 中国图象图形学报: A 辑, 2001, 6(5): 465-9
- [9] Qiu Q, Yuan M, He F, et al. A parallel algorithm for line segment intersection. In: Proceedings of the 21st International Conference on Geoinformatics (GEOINFORMATICS), Kaifeng, China, 2013. 1-4

A spatial clustering based Method for Partitioning of Vector Spatial data for parallel computation

Qiu Qiang * ** , Fang Lei *** , Yao Xiao * ** , Fang Jinyun *

(* Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(** University of Chinese Academy of Sciences, Beijing 100190)

(*** The Hong Kong Polytechnic University, Hong Kong)

Abstract

The partitioning of vector spatial data was studied, and a new data partitioning method based on spatial clustering was proposed to deal with the load balancing problem in the data partitioning stage of parallel vector spatial analysis. This method fully considers the influence of the vector spatial data size and spatial proximity on the efficiency of the algorithm for parallel vector spatial analysis. Firstly, it uses space filling curves to encode the two-dimensional spatial data to keep the characteristic of spatial proximity. Secondly, it fills the features to the spatial feature box to ensure the balance of the feature sizes in each slaver processing. The operations of point-to-surface, curve-to-surface and surface-to-surface overlay were used as the examples to design the contrast test. The experimental result proved that this proposed method improved the load balancing degree and the whole efficiency of the parallel algorithm on the curve and surface spatial data.

Key words: parallel computing, Hilbert curve, vector data, load balancing, feature box