

面向 D-TIN 并行构建的动态条带 数据划分方法与实验分析

齐琳^{1,2,3}, 沈婕^{1,2,3*}, 郭立帅^{1,2,3}, 周侗^{1,2,3}

(1. 南京师范大学地理科学学院, 南京 210046; 2. 南京师范大学虚拟地理环境教育部重点实验室, 南京 210046;
3. 地理信息科学江苏省重点实验室, 南京 210046)

摘要: 数据划分是并行算法设计的重要步骤,其结果的均衡性与高效性是提高并行算法性能的重要前提。对于集聚分布的点集数据,传统的 D-TIN(Delaunay Triangulation)并行算法尚未给出划分结果均衡、划分效率高效的理想解决方案。针对上述问题,本文在传统 D-TIN 并行算法规则条带划分方法的基础上,提出采用动态条带实现针对集聚分布点集数据的均衡、高效划分方法。首先,获取点集的最小外接矩形,并使用规则矩形条带按照同一方向进行点集粗分,然后,按顺序进行相邻条带的合并,必要时需动态调整合并区域边界以达到满足负载均衡的要求。为了提高划分效率,尽量减少边界移动次数,采用了对半移动的规则进行边界的动态调整。为了验证动态条带划分方法的适用性,本文使用人工模拟点集数据,进行加速比测试,使用实验区域真实数据进行 D-TIN 并行构建效率的统计,实验证明,采用该数据划分方法可以获得更高、更稳定的并行加速比,并且数据分布形态和数据规模对加速比的影响较小,进行 D-TIN 构建可以获得更好的执行效率,并且加速效果更加明显。

关键词: D-TIN; 并行计算; 数据划分; 负载均衡; 加速比

DOI: 10.3724/SP.J.1047.2012.00055

1 引言

作为重要的几何计算工具, Delaunay 三角网(以下简称 D-TIN)已被广泛应用于 DEM 表面建模、全自动网格生成、地图制图综合等领域^[1-4]。经过几十年的研究, D-TIN 算法已趋于成熟,但在用于大数据量处理时,目前, D-TIN 算法仍不能够满足人们对效率的要求,优化提高算法效率一直是许多研究者努力的方向,其中,使用并行计算技术来提高 D-TIN 算法的效率是一条有效的途径。D-TIN 并行算法的研究开始于 20 世纪 80 年代末^[5],经过 20 多年的研究,国内外学者提出了诸多的 D-TIN 算法并行设计方法。基于数据划分方式的 D-TIN 并行算法是比较常用的方法,采用这种方式设计的并行算法具有较好的数据独立性,因此,更加容易取得理想的加速比,并且具有较强可移植性。然而,此类算法所面临的一个主要问题是数据

划分结果的均衡性问题,负载均衡是提高系统资源利用率和并行计算性能的一个关键技术^[5],数据划分结果的均衡性和划分效率的高效性是提高这类 D-TIN 并行算法加速比并使并行算法效率达到最优的重要前提。

目前, D-TIN 并行算法的数据划分方法主要有 3 种:①坐标排序的划分方法,首先,对全局点集进行排序,按照排序结果来进行数据划分^[6]。②规则矩形划分方法,通过求取点集的最小外接矩形,将最小外接矩形平均分为若干个规则矩形来实现数据划分,这种划分方法思想简单,划分效率高,是最为常用的一种划分方式^[5,7-8]。③基于投影分割的划分方法,通过把二维点集投影在多维空间中,将位于抛物线上的点将作为数据划分的边界点^[9],这种方法主要用于多维空间的 D-TIN 并行构建。上述方法在用于均匀分布或者随机分布的点集 D-TIN 并行构建时,可以获得比较好的划分结果和划分效

收稿日期: 2011-04-27; 修回日期: 2011-12-14.

基金项目: 国家自然科学基金项目(41071288, 41171350); 江苏研究生创新计划(CXZZ1_0876)。

作者简介: 齐琳(1988-), 女, 硕士, 主要研究方向为地图综合并行算法。E-mail: qilin04043224@126.com。

* 通讯作者: 沈婕(1969-), 女, 博士, 副教授。主要研究方向为地图自动综合、电子地图与网络地图。

E-mail: shenjie@njnu.edu.cn.

率,但用于密度不均匀的点集时,只能以牺牲划分效率为代价来获取划分结果的均衡性。针对面向空间数据的并行算法的数据划分问题,有学者提出四叉树^[10]、Hilbert 曲线^[11]、K-Means 聚类^[12]等方法,进行空间数据划分,实验证明这些方法能够保证空间数据划分结果的平衡。但是,此类算法主要是用于面向空间数据的分布式存储与并行查询,只需要遍历数据,算法执行前后并没有涉及对数据进行转换操作,也不改变数据的几何形状。而 D-TIN 算法输入的数据为点状数据,输出的则为三角网结构的面状数据,如果直接使用这些方法来进行数据划分会造成一些新的问题。因此,上述空间数据划分方法并不适用于 D-TIN 并行构建。

综上所述,研究一种适用于不同分布类型点集数据,能够保证划分结果均衡性和高效性的数据划分方法是提高 D-TIN 并行算法性能的一种重要需求。针对以上问题,本文提出动态条带数据划分方法,笔者使用不同分布形态的点集数据进行数据划分实验,结果证明该算法可以获得较为理想的数据划分结果。另外,通过将划分结果用于 D-TIN 并行构建实验验证了该划分方法有助于提升 D-TIN 并行算法的性能。

2 动态条带数据划分的策略

2.1 面向 D-TIN 并行构建的数据划分原则

为了保证 D-TIN 并行构建的正确性和高效性,在进行数据划分时要考虑 D-TIN 并行算法的一些特殊要求,本文将面向 D-TIN 并行构建的数据划分原则总结为:

(1)划分区域必须为凸多边形。由于 D-TIN 具有外边界为凸壳的特殊性质,为避免 D-TIN 子网三角形边之间出现跨越的现象,给子网合并带来更大的困难甚至产生错误结果,需要在数据划分时保证每一子集数据都是凸多边形包围的,并且不发生重合。

(2)划分结果均衡性要求。保证负载均衡,即使每一个计算节点的计算任务相对平衡,是并行算法设计的重要要求。D-TIN 算法的时间复杂度和点集包含的点数直接相关^[13],因此,数据划分结果应该保证每一个划分区域内所包含点的数量大致相等。

(3)高效性要求。数据划分过程是并行算法的

一部分,数据划分的执行时间也包括在并行算法执行时间中。为了能够使 D-TIN 并行算法能够获得较理想的加速比,数据划分方法应该具有比较高的效率。因此,在满足上述两项原则的前提下,尽量减少因数据划分所消耗的时间。

2.2 数据划分的算法与过程

按照上文总结的面向 D-TIN 并行构建的数据划分方法原则,本文在传统 D-TIN 并行算法中所使用的规则条带划分方法^[5](Equally Strip Partitioning,简称 ESP)基础上,提出动态条带方法的数据划分方法(Dynamic Strip Partitioning,简称 DSP)。该方法基于负载均衡的需求,使用条带粗分、合并的方式并通过动态调整合并区域边界来实现对点集数据的均衡划分。

DSP 方法基本思想描述如下:根据点集的分布范围,将整个空间范围粗略地划分为 m 个初始条带。按照顺序给每一个初始条带编号,并统计每个条带中所包含的点数。从编号最小的条带开始累加条带内点的数量,如果累加到第 i 个条带时点数量超过了单个计算节点(或线程)的负载阈值上限,则需要移动这个条带的边界范围,直到有适量的点被归入当前合并区域为止。重复上述过程,可以使每个计算节点(或线程)获得具有良好划分效果的子集。通常情况下,计算节点(或线程)的数目远远小于点的个数,因此,利用条带划分的方法可以避免对整个点集的遍历,提高点集数据的划分效率。

D-TIN 并行计算的数据划分为避免出现凹多边形而使 D-TIN 子网之间发生跨越,将使用矩形条带来控制数据的划分区域。为了使问题简单化,首先使用规则矩形条带将点集数据进行初始划分。

点集进行条带粗分的个数 m 将直接影响着数据划分的效率, m 分的过细会导致一些不必要的计算过程, m 划分的过粗则会增加边界移动调整的次数。因此,应该选择一个适当的粗分粒度,根据经验,建议选择计算节点数的 10—20 倍,点集规模越大粗分粒度应越细。

为了方便描述,将 DSP 算法所涉及的变量及定义列表,如表 1 所示。

2.3 算法过程

(1)DSP 方法主要过程(见图 1)

① 计算点集的最小外接矩形,如果 $|X_{\max} -$

$|X_{\min}| \geq |Y_{\max} - Y_{\min}|$, 则在 X 轴方向将点集最小外接矩形粗分为 m (m 是 K 的整倍数) 个面积相等的条带, 否则进行 Y 轴方向的最小外接矩形粗分。

表 1 DSP 算法变量描述
Tab. 1 Description of variables in DSP

变量名	变量描述
N	点集 V 所包含点的总数
K	计算节点个数(或者线程个数)
N_A	$N_A = N/K$, 为理想负载均衡状态下, 每个计算节点(线程)所接纳的点数目。通常情况下, 数据划分并不能保证每个节点数据量严格保持一致, 因此需要围绕 N_A 设定一个负载阈值的上下限, N_D, N_U
N_D	负载阈值下限, 即单个计算节点负载量下限, 本文中 $N_D = (1 - 0.01)N_A$
N_U	负载阈值上限, 即单个计算节点负载量上限, 本文中 $N_U = (1 + 0.01)N_A$
N_F	合并区域内的点数目

- ② 设当前处理矩形块编号为 i , 则其点数为 N_i , 如果 $N_i < N_D$, 执行步骤③; 如果 $N_i > N_U$, 执行步骤④; 如果 $N_D < N_i$, 且 $N_i \leq N_U$, 执行步骤⑤。
- ③ 将 $i+1$ 条带合并到 i 条带中, 执行步骤②。
- ④ 将当前合并条带块的尾条带进行划分方向上边界调整, 边界调整规则如下节所述, 将邻接合并区域的矩形条带计入当前合并区域, 继续执行步骤②。
- ⑤ 将这些条带合并, 并给予新的编号, 转到步骤②执行下一次合并, 直到所有条带都被分配为止。

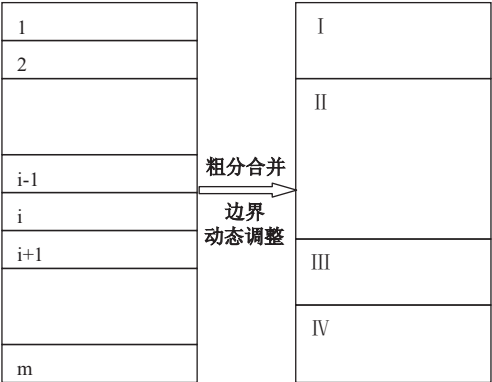


图 1 动态条带数据划分方法过程示意图
Fig. 1 Instruction of the process of dynamic data partitioning

(2)边界调整规则

对点集进行条带粗分后, 在合并过程中粗分条带内点数的累加值有可能不能够满足负载阈值的条件, 如步骤④所描述, 这种情况下需要移动条带边界进行微调。为了减少边界移动的次数, 提高算法效率, 需要制定一个有效的边界移动规则。这里, 我们使用对半移动的方式进行合并区域边界的微调, 如果加入第 i 个条带时, 出现超出负载均值上限的现象, 则将第 i 个条带的下边界向内移动(缩小条带范围)原始宽度值的 $1/2$ 。然后重新统计第 i 个条带内的点数, 将其加入到该合并区域的累积点数中, 并与负载阈值进行比较, 若仍然超过负载阈值上限 N_U 则继续向内移动边界; 若区域内点数小于负载阈值下限 N_D 则需要将条带边界向外移动, 每次边界移动的宽度值为上一次移动值的 $1/2$ 。这是一个迭代执行的过程, 经过若干次的边界移动调整, 可以到达满足负载阈值条件。考虑到一些特殊的情况, 为了避免迭代次数过多, 甚至出现死循环的现象, 影响数据划分效率。实验将迭代次数的上限设为 8, 即每个合并区域通过最多 8 次调整, 此时条带动态调整的细化程度为原始条带范围的 $1/2^8$, 一般情况下可以达到比较好的效果。边界调整完成后, 还需要重新统计新的第 $i+1$ 个条带内的点数, 并开始执行下一个计算节点的区域合并任务。通过使用这种对半移动的边界微调规则, 可以使合并区域通过尽量少的边界调整次数达到优良划分效果的目的。

3 数据划分算法的实验与结果分析

3.1 实验环境与实验数据

硬件参数: CPU: Intel(R) Core(TM) 2 Quad CPU Q8400 @2.66GHz 四核; 内存: 2 GB;

软件平台: 操作系统: Windows 7; 编程环境: Visual Studio 2008, ArcGIS Engine 9.3; 并行编程工具: OpenMP; 编程语言: C++;

点群的空间分布通常分为均匀分布, 随机分布和集聚分布 3 种类型^[14]。均匀分布和随机分布的数据, 在每个区域的分布数量相同或相似, 使用传统的 ESP 方法可以满足此类点集的均衡性划分的需求。集聚分布主要是点群围绕单核或多核呈簇状分布, 主要表现为点群集中分布于整个区域的个别区域, 这种类型数据均衡划分问题使用传统方法比较难获得理想的效果。为了验证 DSP 算法的合

理性与适用性,本文选取了聚集类型的点集数据,包括单核类型和多核类型的人工模拟数据(如图 2(a)、(b)),数据规模:1 000 – 100 000),以及混合了单核和多核集聚类型点集的真实数据(图 2(c),数据规模:500 000)进行了数据划分的实验,数据格式均为 shapefile 格式点数据。

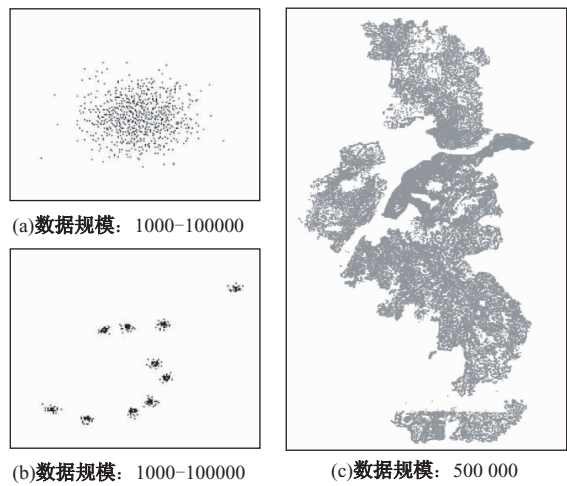


图 2 实验数据示意

Fig. 2 Instruction of experimental data

(a) aggregated distributed point sets of Single-core and Multi-core; (b) thematic point sets of Nanjing

3.2 数据划分结果分析

实验采用南京市地图专题数据作为实验测试的数据集,该数据集包含了 500 000 个点对象,本实验数据为多核集聚分布状态,集中在南京的城区及区县的城镇中心周围。本实验按照 4 个处理节点的划分需求对数据集划分的均衡性作了分析,图 3 为数据划分效果,表 2 是数据划分结果均衡性测试结果。

图 3(a)为使用 ESP 方法进行数据划分 4 节点划分的效果图。采用这种方式进行划分的点集数据每一个区域所占的面积相等,但对于集聚分布点集而言,由于点的分布密度并不相同,因此使用相等的规则矩形划分会使每个区域内的点数具有一定的差异,尤其在点集密度差别较大的情况下,容易造成数据倾斜。

图 3(b)为使用 DSP 方法分布进行 4 节点划分的效果图。这种方法通过对点集进行条带细分并使用动态调整合并区域边界的方式可以将密度不均匀的点集划分成区域面积不等但每个区域内所包括点对象个数相对平衡的子集。

4 节点 DSP 划分过程为,先合并数据块 1 和 2,其未达到阈值下限,但如果加入数据块 3,则超过阈值上限,因此对数据块 3 进行边界调整,通过 3 次调整后,满足划分阈值。对数据块 3 剩余部分继续进行划分,由于其大于阈值上限,所以再对其进行边界调整,通过 2 次调整后满足划分要求,继续执行划分操作直到完成所有数据块分配。

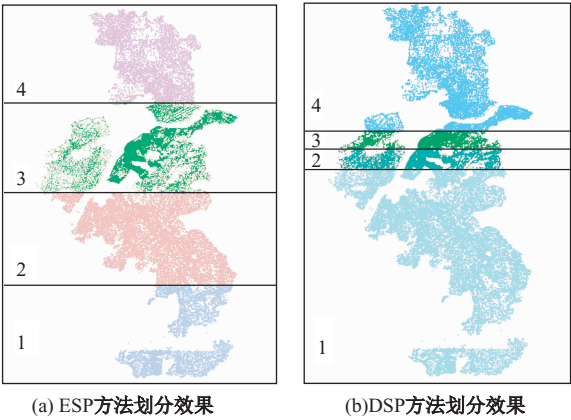


图 3 数据划分效果示意

Fig. 3 Implemation of data partitioning by ESP (a) and DSP (b) method

从表 2 可以看出,使用 ESP 方法产生的划分结果中出现了比较严重的数据倾斜现象。如果使用这种划分结果进行 D-TIN 的并行构建,将会造成计算任务集中在某一个或几个计算节点上,另一些计算节点任务快速完成而变成空节点。使用 DSP 方法可以获得相对平衡的数据划分结果,从而使每个计算节点的所分配得的计算任务相对平衡,保证每一个节点计算任务同时执行与结束,为提高 D-TIN 的并行算法性能提供较好的基础。

表 2 按照 4 个节点划分需求进行数据划分结果

Tab. 2 Result of data partitioning using ESP and DSP for 4 CPU

	1	2	3	4
ESP 方法	20 930	396 736	61 289	21 045
DSP 方法	124 497	124 629	125 676	125 198

从图 3 可以看出,使用 ESP 方法划分点集数据所产生的子集都是分布在矩形区域之内,可以避免子网之间出现边界跨越的现象。通过粗分、合并及动态调整的过程可以获得点数比较均衡的数据子集,这个过程也避免了对点集的全局排序等复杂操作,时间复杂度为 $O(n)$,并且数据划分所耗费的时间

间远远小于 D-TIN 构建的时间,符合面向 D-TIN 构建的数据划分方法的三项原则。

3.3 D-TIN 并行算法性能对比

加速比 $S_p(n)^{[15]}$ 是评价并行算法的一个重要指标,加速比可以定义为:

$$S_p(n) = t_s(n)/t_p(n) \quad (1)$$

$t_s(n)$ 为求解问题的最快串行算法在最坏情形下所需要的运行时间, $t_p(n)$ 为求解同一问题的并行算法在最坏情形下的运行时间。

为了进一步验证数据划分结果对 D-TIN 并行

算法性能的影响,本文使用集聚分布点集数据进行了 D-TIN 并行算法的加速比的测试。实验将分别使用 ESP 方法和 DSP 方法的数据划分结果在多核计算机系统上进行 D-TIN 并行构建,并进行并行算法效率与加速比等性能的评估。

D-TIN 串行算法是使用分块建网的方式进行的,过程如图 4 所示。实验所统计的 $t_s(n)$ 包括了数据划分时间, D-TIN 分块构建时间和子网合并时间。 $t_p(n)$ 为使用相应串行算法的数据划分方法进行 D-TIN 并行构建的执行时间。

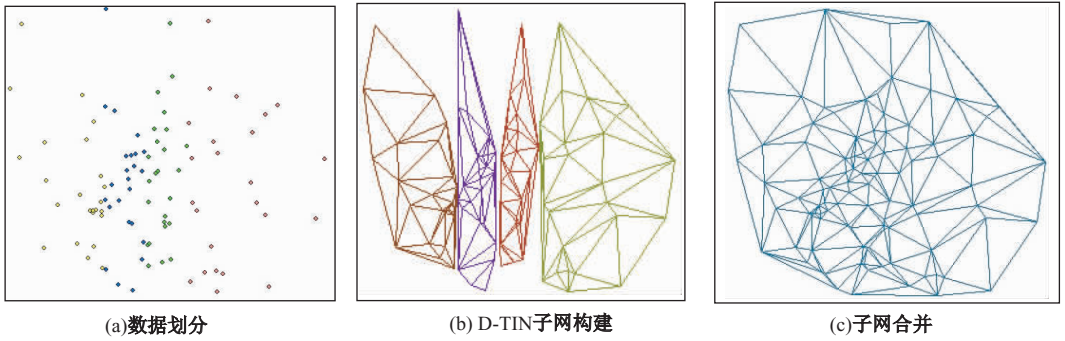


图 4 D-TIN 构建过程示意

Fig. 4 Instruction of the process of Delaunay triangulation constructing

(a) data partitioning; (b) subordinate Delaunay triangulation constructing; (c) subordinate Delaunay triangulation merging

为了验证 DSP 方法的适用性,本文分别使用了单核集聚型和多核集聚型的人工模拟点集数据和真实数据进行实验。人工模拟数据分别选取了几种不同规模点集(数据规模:1 000 - 100 000)进行加速比测试,真实数据则进行 D-TIN 并行构建效率的统计。实验平台为四核 PC 机(实验环境具体参数如 3.1 节所描述),分别使用 2 线程和 4 线程进行

D-TIN 并行构建实验,点集分块个数与线程数一致。

图 5 分别使用 ESP 和 DSP 方法的数据划分结果进行 D-TIN 并行构建的加速比统计图,图 5(a) 为使用单核集聚分布数据实验结果,图 5(b) 为使用多核集聚分布数据实验结果。图中 ESP 2 线程、ESP 4 线程分别代表使用 ESP 方法进行双线程和

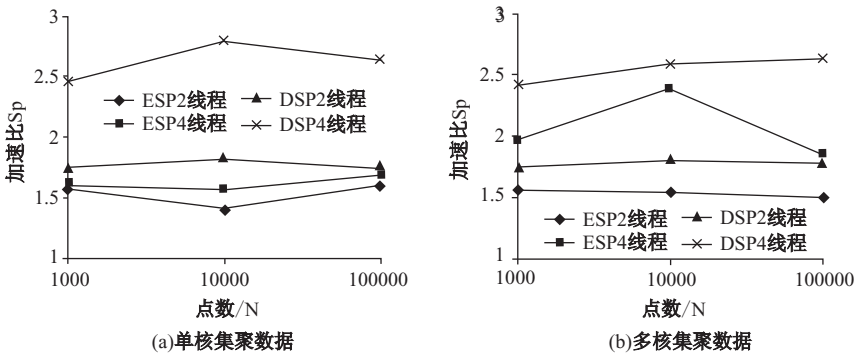


图 5 两种数据划分方法 D-TIN 并行构建加速比

Fig. 5 Contrast of the two Delaunay triangulation parallel algorithms speed-up

(a) single-core aggregated data; (b) multi-core aggregated data

四线程的 D-TIN 并行构建的加速比。DSP2 线程、DSP 4 线程含义同上。图 4 表明使用 DSP 方法的划分结果进行 D-TIN 并行构建可以获得更高、更稳定的并行加速比,并且数据分布形态和数据规模对加速比的影响较小。

图 6 为使用真实数据(图 3(c),数据规模:500 000)在 4 核计算机上分别进行 2 区域和 4 区域划分后进行 D-TIN 构建的执行时间统计图。从图 6 看出,使用 DSP 方法进行数据划分结果进行 D-TIN 构建可以获得更好的执行效率,并且加速效果更加明显。

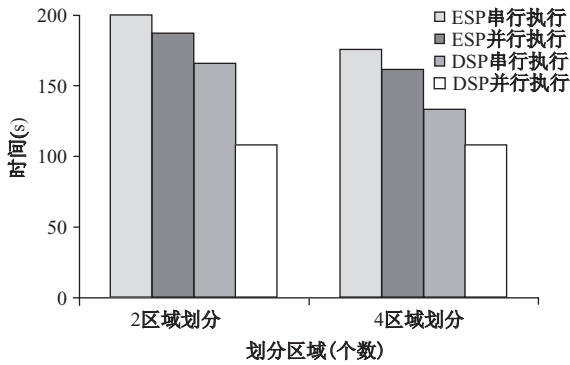


图 6 两种数据划分方法 D-TIN 构建执行时间比较
Fig. 6 The time of D-TIN construction by ESP and DSP

4 结论与展望

本文提出了使用动态条带划分方法来实现针对集聚型点集数据的均衡划分,实验证明该算法可以产生较为均衡的划分结果,保持计算节点(或线程)之间的负载平衡,从而并有助于 D-TIN 并行算法的算法效率和加速比等性能的提高。

受条件限制,仅探讨了在单机多核环境下通过使用 OpenMP 实现以数据划分的 D-TIN 并行算法,对于机群系统下 DSP 算法对 D-TIN 并行算法性能的影响,还有待进一步研究。另外,还应该深入研究其他以 D-TIN 算法为基础的地图制图综合算法的并行设计方法,在此基础上有望使制图自动综合在网络地图、应急环境地图生成方面发挥重要的作用。

参考文献:

[1] 汤国安,刘学军,闫国年. 数字高程模型及地学分析的原理与方法[M]. 北京:科学出版社,2005.

[2] 李水乡,陈斌,赵亮,等. 快速 Delaunay 逐点插入网格生成算法[J]. 北京大学学报(自然科学版),2007,43(3): 302 - 306.

[3] 艾廷华,刘耀林. 保持空间分布特征的群点化简方法[J]. 测绘学报,2002,31(002):175 - 181.

[4] Yan H and Weibel R. An algorithm for point cluster generalization based on the Voronoi diagram[J]. Computers & Geosciences, 2008, 34(8): 939 - 954.

[5] Davy J R, Dew P M. A note on improving the performance of Delaunay triangulation[C]. // Proc. of Computer Graphics International '89, 1989.

[6] Jiang J, Zhang M, Liao X. Study of load balancing algorithms based on multiple resources[J]. Acta Electronica Sinica, 2002, 30(8):1148 - 1152.

[7] Cignoni P, Montani C, Pereo R, et al. Parallel 3D Delaunay triangulation [C]. Wiley Online Library, 1993.

[8] Chen M B, Chuang T R, Wu J J. Efficient parallel implementations of 2D Delaunay triangulation with high performance FORTRAN [J]. 2000.

[9] Hardwick J C. Implementation and evaluation of an efficient parallel Delaunay triangulation algorithm [C]. ACM, 1997.

[10] Wang S and Armstrong M P, A quadtree approach to domain decomposition for spatial interpolation in grid computing environments [J]. Parallel Computing, 2003,29(10): 1481 - 1504.

[11] 赵春宇,孟令奎,林志勇. 一种面向并行空间数据库的数据划分算法研究[J]. 武汉大学学报:信息科学版, 2006,31(11):962 - 965.

[12] 贾婷,魏祖宽,唐曙光,等. 一种面向并行空间查询的数据划分方法[J]. 计算机科学,2010,37(8):198 - 200.

[13] 武晓波,王世新,肖春生. Delaunay 三角网的生成算法研究[J]. 测绘学报,1999,28(1):28 - 35

[14] 周侗,龙毅,汤国安,等. 面向集聚分布空间数据的混合式索引方法研究[J]. 地理与地理信息科学,2010 (001):7 - 10.

[15] 陈国良. 并行算法的设计与分析[M]. 北京:高等教育出版社,1994.

Dynamic Strip Partitioning Method Oriented Parallel Computing for Construction of Delaunay Triangulation

QI Lin^{1, 2, 3}, SHEN Jie^{1, 2, 3}, GUO Lishuai^{1, 2, 3} and ZHOU Tong^{1, 2, 3}

(1. *School of Geographic Science, Nanjing Normal University, Nanjing 210046, China;*

2. *Key Laboratory of Virtual Geographic Environment, MOE, Nanjing 210046, China;*

3. *Key Laboratory of Geographic Information Science of Jiangsu Province, Nanjing Normal University, Nanjing 210046, China*)

Abstract: Data partitioning is an important step of parallel algorithm design. The load balance and efficiency of data partitioning is the precondition for improvement of parallel algorithm efficiency. For aggregated distributed point sets, the traditional Delaunay triangulation parallel algorithm can't ensure the balance and the execution's efficiency of the partitioning result. In view of the problems above, this paper we proposed a partitioning method using dynamic strips based on the idea of equally strips partitioning method in traditional Delaunay Triangulation construction and we titled it Dynamic Strip Partitioning Method. The detailed steps of this algorithm are as follows. First, the minimum bounding rectangle of the point data set should be obtained and the point set is roughly split using regular slim strips in the same direction. Then the number of points in every strip would be counted and the neighbor strips are merged into a partition region from the first strip in the sequence following a certain regulation. The boundaries of some strips should be moved dynamically if the total amount of points in these strips reached the load threshold value. In order to promote the efficiency of partitioning and reduce the boundaries movement, a rule of "move half points a time" has been used. We tested the speed-up of the Delaunay Triangulation parallel algorithm using the artificial point sets and tested the performance of the Delaunay triangulation parallel algorithm using the real test area point sets in the multi-kernel parallel computing systems. The results of the experiments showed that the method of dynamic strips partitioning can help to get high and stable speed-up of the Delaunay triangulation parallel algorithm and the data distributional pattern and size has less influence to it. Delaunay triangulation parallel algorithm based on dynamic strips partitioning method can get high efficiency and the speed-up effect is superior to the traditional method.

Key words: D-TIN; parallel computing; data partition; load balance; speed-up ratio