

文章编号:1001-9081(2014)07-1852-05

doi:10.11772/j.issn.1001-9081.2014.07.1852

简单要素模型多边形拓扑检查并行算法

任沂斌^{1,2}, 陈振杰^{1,2*}, 李飞雪^{1,2}, 周琛^{1,2}, 杨云丽^{1,2}

(1. 南京大学 地理与海洋科学学院, 南京 210023; 2. 江苏省地理信息技术重点实验室(南京大学), 南京 210023)

(*通信作者电子邮箱 chenzj@nju.edu.cn)

摘要: 将并行计算应用到大数据量简单要素模型多边形拓扑检查中, 设计实现了简单要素模型多边形拓扑检查并行算法。算法针对拓扑检查的计算特点, 改进了主从式并行策略, 在主进程中进一步划分线程以实现任务并行, 从而隐藏拓扑错误提取和结果写入时间。采用 MPI 和 PThread 实现进程与线程的结合。利用苏南五市土地现状调查地类图斑数据对算法进行测试。经测试, 该算法能够对大数据量简单要素模型多边形进行准确、快速的拓扑检查。算法提出的进程与线程结合的任务并行策略相对于传统主从式策略加速比提高约 20%。

关键词: 并行计算; 简单要素模型; 拓扑检查; 消息传递接口; PThread

中图分类号: TP751 **文献标志码:**A

Parallel algorithm of polygon topology validation for simple feature model

REN Yibin^{1,2}, CHEN Zhenjie^{1,2*}, LI Feixue^{1,2}, ZHOU Chen^{1,2}, YANG Yunli^{1,2}

(1. School of Geographic and Oceanographic Sciences, Nanjing University, Nanjing Jiangsu 210023, China;

2. Jiangsu Provincial Key Laboratory of Geographic Information Sciences and Technology (Nanjing University), Nanjing Jiangsu 210023, China)

Abstract: Methods of parallel computation are used in validating topology of polygons stored in simple feature model. This paper designed and implemented a parallel algorithm of validating topology of polygons stored in simple feature model. The algorithm changed the master-slave strategy based on characteristics of topology validation and generated threads in master processor to implement task parallelism. Running time of computing and writing topology errors was hidden in this way. MPI and PThread were used to achieve the combination of processes and threads. The land use data of 5 cities in Jiangsu, China, was used to check the performance of this algorithm. After testing, this parallel algorithm is able to validate topology of massive polygons stored in simple feature model correctly and efficiently. Compared with master-slave strategy, the speedup of this algorithm increases by 20%.

Key words: parallel computation; simple feature model; topology validation; Message Passing Interface (MPI); PThread

0 引言

简单要素模型是一种常用的矢量数据存储形式, 被广泛应用于商用、开源地理信息系统(Geographic Information System, GIS)软件中, 但由于不含拓扑关系数据, 在创建和编辑过程中易产生数据冗余和不一致性等拓扑错误^[1-3]。因此, 在此类数据使用过程中会频繁地进行拓扑检查^[4]。目前, 商业 GIS 软件对此类数据进行拓扑检查前, 一般需要先将其导入空间数据库建立起全局拓扑关系^[5]。对大数据量多边形数据, 数据导入导出和全局拓扑关系建立比较耗时^[6], 严重影响拓扑检查的效率。因此, 如何直接对大数据量简单要素模型多边形进行准确、快速的拓扑检查, 显得尤为重要。并行计算允许多个处理单元同时执行一个操作, 从而起到加速处理的作用^[7-8]。将并行计算引入地理学领域以解决大数据量地理数据的计算问题已成为发展趋势^[9]。当前, 地理计算方面的并行算法主要集中于栅格数据处理、空间分析、数据

转换等方面^[10-12], 拓扑方面的并行算法研究较少。文献[13]从空间数据划分的角度, 提出了一种实现拓扑关系高效并行计算的矢量数据划分方法, 但没有针对拓扑检查提出完整的并行算法流程。针对上述问题, 本文以实现对大数据量简单要素多边形的快速拓扑检查为目的, 综合考虑数据划分、任务调度等方面, 提出了适用于简单要素模型多边形拓扑检查的并行策略。以检查多边形要素类的“多边形之间不存在重叠”拓扑规则为例, 实现了多核环境下的多边形拓扑检查并行算法。

1 简单要素模型拓扑检查方法

对简单要素模型拓扑检查的目标是判断所有要素之间的拓扑关系是否满足拓扑规则, 并提取出不满足拓扑规则的拓扑错误区域。如: “多边形之间不存在重叠”规则要求同一多边形要素类中多边形之间不能存在重叠^[14], 对此规则检查就是查找要素间重叠区域。本文采用基于点集理论维度扩展的

收稿日期:2013-12-26;修回日期:2014-02-21。 基金项目:国家 863 计划项目(2011AA120301)。

作者简介:任沂斌(1990-),男,山东青岛人,硕士研究生,主要研究方向:地理信息系统、高性能地理计算; 陈振杰(1974-),男,陕西宝鸡人,副教授,博士,主要研究方向:GIS 算法、高性能地理计算; 李飞雪(1983-),女,黑龙江齐齐哈尔人,副教授,博士,主要研究方向:城市增长建模; 周琛(1990-),男,江苏宿迁人,硕士研究生,主要研究方向:高性能地理计算; 杨云丽(1989-),女,江苏淮安人,硕士研究生,主要研究方向:高性能地理计算。

9-交模型(Dimensionally Extended Nine-Intersection Model, DE-9IM)作为描述要素间拓扑关系的模型^[13]。拓扑关系计算需先判断要素间是否相交,如果相交则在相交关系的基础上进行其他拓扑关系计算^[13]。为了提高拓扑检查的效率,本文分三步进行拓扑检查:一维空间索引构建,相交关系判断,拓扑错误区域提取和写入。其中,第三步是在一维空间索引构建、相交关系判断的基础上,针对存在相交关系的多边形提取拓扑错误区域,并将这些错误区域写入结果文件。

1.1 一维空间索引构建

为提高相交关系判断效率,需要对所有多边形建立一维空间索引。基本思路为:根据多边形最小外包矩形(Minimum Bounding Rectangle, MBR)最小X坐标对多边形进行排序,形成按X值由小到大的多边形序列(图1)。建好索引后,每个多边形只需与排在其后面的多边形进行拓扑关系计算。

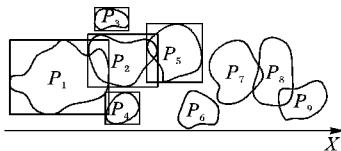


图1 一维空间索引示意图(以按X坐标建立索引为例)

1.2 相交关系判断

相交关系判断是判断多边形间是否相交。以图1中建好索引的多边形序列为例,从中取出第一个多边形 P_1 作为当前多边形,依次与后续多边形 $P_i(i=2,3,\dots,9)$ 进行相交关系判断。先利用MBR进行粗判断,若MBR相交再利用空间关系判断函数进行精确判断。 P_1 完成判断后,继续取出第二个多边形 P_2 作为当前多边形,重复上述过程,直至序列中所有多边形都完成判断。当序列中某个多边形MBR的最小X坐标大于等于当前多边形MBR最大X坐标时,后续多边形均不可能与当前多边形相交,即可终止本次循环,继续对下一个多边形的循环判断。以图1的多边形序列为例, P_1 作为当前多变形时,当判断到 P_5 时终止 P_1 多边形的判断循环。

2 并行算法设计与实现

2.1 矢量数据划分

2.1.1 矢量数据划分方法

简单要素模型拓扑检查的基本特征是要素间拓扑关系的计算,因而数据划分策略要保证多边形要素在空间上的邻近性。综合考虑要素空间邻近性、进程间负载均衡和划分预处理时间,本文采用矢量数据行、列划分方法:如果数据在X方向的跨度大于其在Y方向的跨度,则采用列划分法;否则采用行划分法,如图2所示。

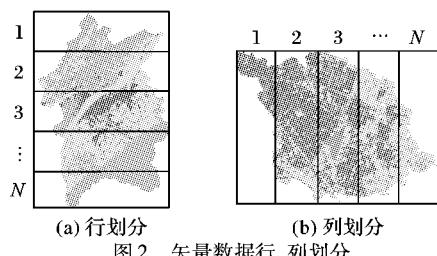


图2 矢量数据行、列划分

划分的行、列数要大于进程数,以便在任务调度时采用动态分配策略平衡进程间的负载。若按行划分,则按多边形MBR最小X坐标建立索引;若按列划分,则按多边形MBR最

小Y坐标建立索引。此划分方法的预处理时间较少,能保证多边形要素在X或Y方向的空间邻近性,与上文的一维空间索引结合,通过多边形MBR过滤,可以有效减少相交关系判断的计算次数。

2.1.2 跨任务块边界多边形处理

跨任务块边界(以下简称跨边界)多边形是指在数据划分时跨数据块边界的多边形。为了不遗漏拓扑错误,跨边界多边形必须被所跨进程均读入。

图3中 a,b,c 为多边形,上下两任务块分别被进程I和进程K处理。图3(a),当前处理的多边形对中只有一个一个多边形 a 跨边界时,进程I处理 a,b ,进程K处理 a,c ,没有遗漏和冗余。图3(b),当前处理的多边形 a,b 均跨边界,进程I,K均读入多边形 a,b ,若不做特殊处理,I,K均会对 a,b 进行处理,造成冗余处理。

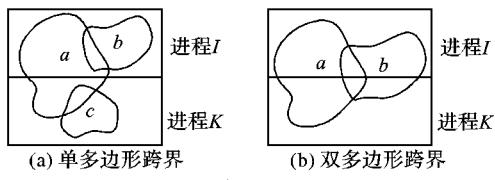


图3 跨边界多边形

为解决上述问题,本算法将两多边形均跨边界的情况交由主进程处理:主进程中建立跨边界多边形集合 D , D 的元素为多边形FID(Feature Identification)对 (FID_1, FID_2) ,初始状态下 D 为空集。从进程判断出两多边形相交且均为跨边界多边形时,将两多边形FID对发送给主进程,主进程首先查找集合 D 中是否存在多边形 a,b 的FID对,若存在则不需要进行处理。否则,对 a,b 进行处理,并将 a,b 的FID对加入到集合 $D\{(FID_a, FID_b)\}$ 。图3(b)中,进程I,K均将多边形 a,b 的FID发给主进程,但主进程只进行一次处理。

2.2 任务调度策略

2.2.1 改进的主从并行策略

简单要素模型多边形拓扑检查主要计算集中在相交关系判断和拓扑错误提取写入两部分。传统主从式并行策略下,主进程负责任务分发,从进程中串行执行相交关系判断和拓扑错误提取写入这两个任务。然而,由于并行写入的限制,同一时间段只能有一个进程将拓扑错误写入文件,此时其他需要写入结果的进程将等待。这种受并行写入限制造成的进程间等待增加了算法的写入时间,同时阻塞相交关系判断的执行,影响并行算法效率。而且,主进程中只分发任务,不能充分利用主进程中的计算资源。

针对上述问题,本文作了以下改进:1)考虑到拓扑检查时,判断相交要针对所有多边形分析,任务较大;相比而言,提取相交多边形的错误区域只对确定相交的多边形进行,拓扑错误提取函数的执行次数远远小于相交关系判断函数的执行次数,计算量小很多。因此,本文将拓扑错误提取写入任务交由主进程执行,各从进程只执行相交关系判断,对判断出存在相交关系的多边形由主进程统一提取错误区域并写入文件,两任务并行执行,这样便可消除各从进程进行拓扑错误写入时的等待时间;2)主进程中还需执行任务分发,即主进程中需同时执行任务分发和拓扑错误提取写入两任务。因此,为了提高主进程的处理效率,本文在传统主从式并行模式基础上,对主进程进一步做线程级的并行。即在主进程内部开辟

两个线程,分别负责任务分发和拓扑错误提取写入。任务分发线程负责动态地向从进程分发任务,拓扑错误处理线程则根据从进程相交关系判断结果,只对存在相交关系的多边形提取拓扑错误并写入文件。进程控制采用 MPI^[15],线程控制采用 PThread^[16]。

在改进的主从式并行策略中,主进程中同时进行任务分发和拓扑错误提取写入,从进程进行相交关系判断,任务分发、相交关系判断、拓扑错误提取写入这三个任务并行执行。如图 4 所示。

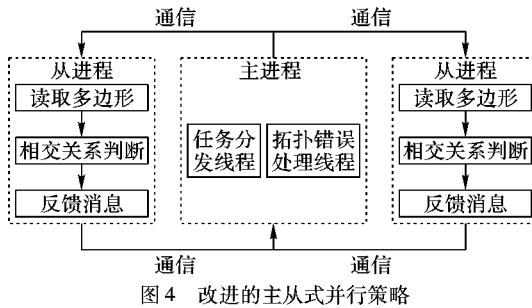


图 4 改进的主从式并行策略

2.2.2 基于任务量的动态任务分发

1) 创建任务队列。

任务分发线程完成任务划分后(假设采用按行划分),获取每个任务块中多边形个数,即任务量。对所有任务块按其任务量由大到小排序,建立待处理任务队列存储任务块对应的行号。

2) 初次任务分配。

任务分发线程从任务队列中取出前 n 项(n 为从进程个数),发送给所有从进程。从进程根据行号计算任务块对应的矩形区域,读取位于区域内的多边形进行处理。

3) 循环分发任务。

从进程完成当前任务后,向主进程请求新任务。任务分发线程检索任务队列,若任务队列不为空,则发送下一任务给从进程;若任务队列为空,说明任务已分配完,则向从进程发送结束标记。

基于任务量的动态任务分发保证了任务按任务量由大到小执行。在任务队列中所剩任务数大于等于从进程数之前,不会出现空闲的从进程,从而较好地保证进程间负载均衡。

基于任务量的动态任务分发伪代码如下:

主进程中任务分发线程

```

int blockNum = N; // N 为任务块数量
/* ----- 创建任务队列 ----- */
int * pNumber = new int[ blockNum ];
/* 定义数组 pNumber, 存储每个任务块多边形数量 */
for ( int i = 0; i < blockNum; i++ )
{
    // 循环获取每个任务块对应矩形区域内的多边形个数,
    // 存入 pNumber 数组;
}
根据任务块所含的多边形数量由大到小创建任务队列 qTask;
/* ----- 初次分发任务 ----- */
从任务队列 qTask 中取出前 processes 个元素, 通过 MPI_Send 函数发送给从进程; //processes 为从进程个数
/* ----- 循环分发任务 ----- */
循环接收从进程的任务请求, 向从进程发送新任务直至任务队列 qTask 中元素为空;

```

2.2.3 相交判断和拓扑错误处理并行执行

1) 从进程判断出某两个多边形之间存在相交关系时,将这两个多边形的 FID 发送给主进程,从进程完成消息发送后继续进行后续多边形的相交关系判断(如图 5)。

2) 主进程中拓扑错误处理线程接收从进程所发的 FID,读取对应的多边形,提取两多边形的拓扑错误区域并写入结果文件。

3) 拓扑错误处理线程在提取当前两多边形拓扑错误的同时,从进程已经继续后续多边形的相交关系判断。多边形拓扑错误提取和写入时间被隐藏在了后续多边形的相交关系判断过程中。

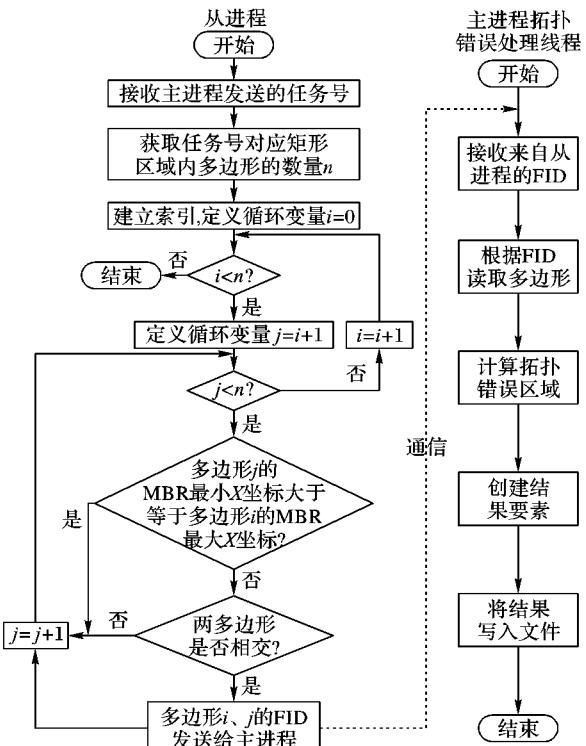


图 5 相交关系判断、拓扑错误提取写入并行执行流程

本部分伪代码如下:

从进程

```

/* ----- 接收主进程发送来的任务号 ----- */
int taskID = 0; // 存储主进程发来的任务号
int featureNum = 0; // 存储任务块内多边形个数
MPI_Status statusP; // 定义 MPI_Status 型变量
MPI_Recv( &taskID, 1, MPI_INT, 0, 4, MPI_COMM_WORLD, &statusP ); // 接收主进程所发消息
/* ----- 建立一维索引(按 X 方向建立索引) ----- */
获取任务块内所有多边形的 MBR 和多边形个数;
利用快速排序算法按 MBR 的最小 X 坐标对多边形排序;
/* ----- 多边形相交关系判断 ----- */
for ( int i = 0; i < featureNum; i++ )
{
    if ( 两多边形相交 )
        // 通过 MPI_Send 向主进程发送两多边形的 FID;
}

```

主进程拓扑错误处理线程

```

/* ----- 读取从进程发来的多边形 ----- */
int * pFID = new int[ 2 ]; // 定义数组, 存储接收到的 FID 对
MPI_Recv( pFID, 2, MPI_INT, MPI_ANY_SOURCE, mFlag, MPI_COMM_WORLD, &status );

```

```

//接收相交多边形 FID,存入数组 pFID 中
//利用 OGR 库读取两多边形, pLayer 为源矢量文件
OGRFeature * pFeature1 = pLayer -> GetFeature(pFID[0]);
OGRFeature * pFeature2 = pLayer -> GetFeature(pFID[1]);
/*-----拓扑错误提取写入-----*/
OGRFeature * pResultFeature = ComputeFault(pFeature1,
                                             pFeature2); //计算拓扑错误区域
WriteFault(pResultFeature);
/*该函数调用 OGR 库,将拓扑错误区域写入 Shapefile 文件*/

```

2.2.4 主从进程通信

算法执行过程中主从进程通信如图 6。从进程向拓扑错误处理线程发送多边形 FID,当前任务完成时发送完成标记并向任务分发线程发送任务请求;任务分发线程向从进程发送任务号,若所有任务均已分配完,则发送结束符。

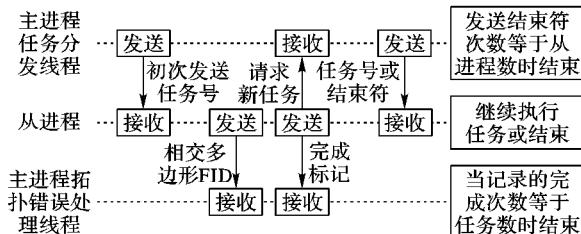


图 6 算法执行过程中主从进程通信

3 并行算法测试

3.1 软硬件环境与测试数据

本算法采用标准 C++ 在 Visual Studio 2010 平台下开发,采用 MPI 实现进程控制、PThread 实现线程控制,采用开源地理数据格式转换类库 OGR 库读写矢量数据。算法测试环境为 IBM System x3500-M3X 服务器,配置 2 颗 Intel Xeon Quad Core E5620 四核超线程 CPU、8 GB 内存、2 TB 硬盘、双口千兆以太网。软件配置:操作系统为 Centos Linux 6.0,其中 MPI 的实现产品选择 Openmpi 1.6, GDAL/OGR 库为 gdal-1.9.2,Linux 系统自带对 PThread 的支持。

测试数据为苏南五市(南京、镇江、常州、无锡、苏州)土地现状调查地类图斑数据。数据格式为 Shapefile 格式,图斑总数为 130 万个,数据量为 1.6 GB。

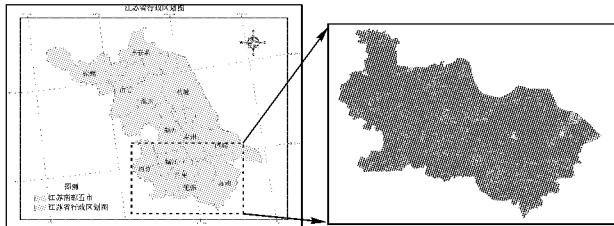


图 7 苏南五市地理区位及土地现状调查地类图斑

3.2 测试结果与并行效率分析

3.2.1 结果正确性分析

对苏南五市土地现状调查地类图斑进行“多边形之间不存在重叠”拓扑检查,共检查出 71005 条拓扑错误。图 8 展示了本算法所求拓扑错误与 ArcGIS 所求拓扑错误的细节情况对比。从图中可以看出,本算法所求的拓扑错误区域与 ArcGIS 所求一致,即本算法所求的拓扑错误区域是正确的。

3.2.2 并行效率分析

该测试数据下串行算法运行时间为 4293.41 s,由于算法采用主从式并行模式,所以并行算法从两个进程开始运行,运

行时间随着进程的增加而减少,当进程数达到 8 时,算法运行时间达到最小,为 645.22 s(如图 9)。之后,随着进程数的增加运行时间不再明显减小,稳定在 650 s 左右。算法加速比随着进程的增加呈现明显上升趋势,当进程数为 8 时达到峰值 6.65,随后不再增加,稳定在 6.5 左右(如图 10)。可见,当进程数为 8 时,本算法对计算资源的利用率达到最好,当进程数超过 8 时,由于进程对计算资源的竞争导致算法运行时间略有增加,加速比略有下降,最终趋于稳定。



(a) 本文算法所求拓扑错误 (b) ArcGIS 算法所求拓扑错误
图 8 本算法与 ArcGIS 所求拓扑错误细节处对比

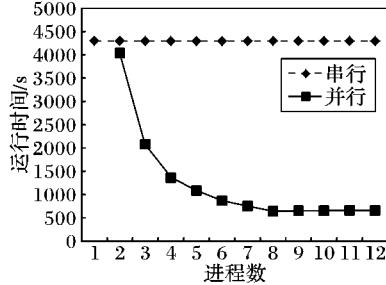


图 9 算法运行时间

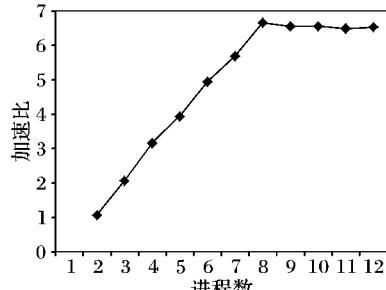


图 10 算法加速比

3.2.3 本算法策略与主从式并行策略对比分析

本文还采用 MPI 实现了传统主从式并行策略下多边形拓扑检查并行方法,并与本文 MPI + PThread 实现的进程与线程结合的任务并行策略进行了对比,两者对比如图 11、图 12。

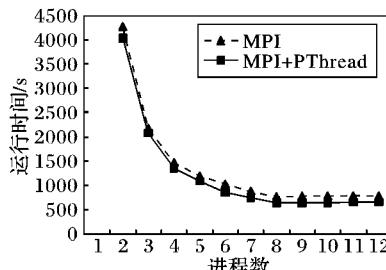


图 11 MPI + PThread 与 MPI 并行算法运行时间对比

由图 11、图 12 可以看出,本文并行策略相对于传统主从式并行策略可以显著地缩短算法运行时间,展现出更好的加速比。原因是进程与线程结合的任务并行策略下,拓扑错误

提取写入的时间隐藏在相交关系判断过程中,消除了各从进程中拓扑错误提取时间和写入文件时的等待时间。

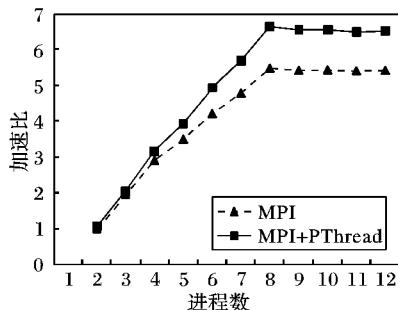


图12 MPI+PThread与MPI并行算法加速比对比

由表1看出,本文并行策略与主从式并行策略的加速比差值随着进程数的增加而增大,当进程数为8时达到最大值1.18,之后稳定在1.10左右。原因是在主从式并行策略下,随着进程数的增加,各从进程在拓扑错误写入时的等待时间增加,写入时间在从进程的整个处理过程中所占比重增加;而且,等待时间增加又会阻塞后续相交关系判断的执行。以上两方面原因使得加速比的增速逐渐缓慢。而在本文并行策略下,不会出现各从进程等待写入和阻塞问题,加速比呈现近似线性的增加趋势。因此二者的加速比差值逐渐增大,并随着对计算资源的使用达到饱和而趋于平稳。

表1 MPI+PThread与MPI并行模式运行时间、加速比差值

进程数	运行时间差/s	加速比差	进程数	运行时间差/s	加速比差
2	-236.41	0.06	8	-131.50	1.18
3	-104.13	0.10	9	-136.34	1.13
4	-118.14	0.25	10	-136.00	1.13
5	-112.44	0.44	11	-132.20	1.08
6	-169.43	0.72	12	-134.10	1.11
7	-138.28	0.90			

4 结语

本文研究了并行技术的基本方法,针对拓扑检查的特点,设计了适用于大数据量简单要素模型多边形数据拓扑检查的矢量数据划分方法和任务调度策略,提出了简单要素模型多边形拓扑检查并行方法完整流程,实现了多核环境下多边形拓扑检查并行算法。以验证“多边形之间不存在重叠”规则为例,对苏南五市土地现状调查地类图斑进行拓扑检查,并将本文并行策略与传统主从式并行策略进行对比分析。通过实验分析,得出本算法可以对简单要素模型多边形数据进行准确、快速的拓扑检查,算法提出的进程与线程结合的任务并行策略相对于传统主从式并行策略展现出了更好的加速比,更适用于大数据量简单要素模型的拓扑检查。然而,本文只实现了对“多边形之间不存在重叠”规则的拓扑检查,在后续工作中,将会以本文提出的并行模式为基础,完善拓扑错误提取方法,以实现其他拓扑规则的拓扑检查。

参考文献:

- [1] CHEN Z, WU X, WU L. Polygon overlay analysis algorithm based on monotone chain and STR tree in the simple feature mode[J]. Acta Geodaetica et Cartographica Sinica, 2010, 39(1): 102–107. (陈占龙, 吴信才, 吴亮. 基于单调链和STR树的简单要素模型多边形叠置分析算法[J]. 测绘学报, 2010, 39(1): 102–107.)
- [2] WANG X. Research on optimisation application of data structure processing of geographic information[D]. Zhengzhou: Information Engineering University, 2010. (王晓理. 地理信息数据结构处理优化应用研究[D]. 郑州: 信息工程大学, 2010.)
- [3] HUANG X, MA J. Geographical information system[M]. Beijing: Higher Education Press, 2008. (黄杏元, 马劲松. 地理信息系统概论[M]. 北京: 高等教育出版社, 2008.)
- [4] PAN C, ZHOU S, YANG L, et al. The research of spatial topology rules managing mechanism based on MapGIS7.0[J]. Science of Surveying and Mapping, 2009, 34(1): 222–224. (潘婵玲, 周顺平, 杨林, 等. 基于MapGIS7.0的空间拓扑规则管理机制研究[J]. 测绘科学, 2009, 34(1): 222–224.)
- [5] FENG H, MA T, LIU W, et al. Topology relationship analysis of cadastral spatial database and validity method based on rule[J]. Journal of Computer Applications, 2006, 26(17): 2522–2524. (冯杭建, 麻土华, 刘伟宏, 等. 地籍空间数据库拓扑关系分析及基于规则的验证方法[J]. 计算机应用, 2006, 26(17): 2522–2524.)
- [6] XIE Z, YE Z, WU L. Polygon overlay analysis algorithm using the simple data model[J]. Geography and Geo-Information Science, 2007, 23(3): 19–23. (谢忠, 叶梓, 吴亮. 简单要素模型下多边形叠置分析算法[J]. 地理与地理信息科学, 2007, 23(3): 19–23.)
- [7] MINETER M J. A software framework to create vector-topology in parallel GIS operations[J]. International Journal of Geographical Information Science, 2003, 17(3): 203–222.
- [8] SUI J, LIU H, WANG J, et al. Overview of multicore multithread technologies[J]. Journal of Computer Applications, 2013, 33(S1): 239–242. (眭俊华, 刘慧娜, 王建鑫, 等. 多核多线程技术综述[J]. 计算机应用, 2013, 33(S1): 239–242.)
- [9] ZHANG S. Study on execution time prediction for parallel geocomputation in multi-core cluster environment[D]. Changsha: National University of Defense Technology, 2011. (张思乾. 多核集群环境下并行地理计算执行时间预测技术研究[D]. 长沙: 国防科学技术大学, 2011.)
- [10] GUAN Q, CLARKE K C. A general-purpose parallel raster processing programming library test application using a geographic cellular automata model[J]. International Journal of Geographical Information Science, 2010, 24(5): 695–722.
- [11] PURI S, AGARWAL D, HE X, et al. MapReduce algorithms for GIS polygonal overlay processing[C]// Proceedings of the 27th International Parallel and Distributed Processing Symposium Workshops and PhD Forum. Piscataway: IEEE, 2013: 1009–1016.
- [12] WANG Y, CHEN Z, CHENG L, et al. Parallel scanline algorithm for rapid rasterization of vector geographic data[J]. Computers and Geosciences, 2013, 59: 31–40.
- [13] YANG Y, WU L, GUO J, et al. A method of spatial data partition for efficient parallel computing of topological relations[J]. Geography and Geo-Information Science, 2013, 29(4): 25–29. (杨宜舟, 吴立新, 郭甲腾, 等. 一种实现拓扑关系高效并行计算的矢量数据划分方法[J]. 地理与地理信息科学, 2013, 29(4): 25–29.)
- [14] WANG B. Rules-based topological relationships in Geodatabase[J]. Geomatics and Spatial Information Technology, 2004(3): 17–19. (王宝利. Geodatabase中基于规则的拓扑关系[J]. 测绘与空间地理信息, 2004(3): 17–19.)
- [15] RABENSEIFNER R, HAGER G, JOST G. Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes[C]// Proceedings of the 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing. Piscataway: IEEE, 2009: 427–436.
- [16] SACHDEV G S, SUDAN K, HALL M W, et al. Understanding the behavior of PThread applications on non-uniform cache architectures[C]// Proceedings of the 2011 International Conference on Parallel Architectures and Compilation Techniques. Piscataway: IEEE, 2011: 175–176.