

doi:10.3799/dqkx.2010.040

分布式空间分析运算关键技术

吴亮¹, 谢忠^{1*}, 陈占龙^{1,2}, 马丽娜^{1,2}

1. 中国地质大学信息工程学院, 湖北武汉 430074

2. 地理信息系统软件及其应用教育部工程研究中心, 湖北武汉 430074

摘要: 为了提高分布式环境下海量空间数据的空间运算的效率, 解决空间分析处理模块在设计上与底层的数据库服务协议、空间数据模型等透明一致的设计问题, 分析了分布式空间运算具备的基本特征, 从空间运算任务分解和分布式空间数据的划分方法、共享数据复制策略、基于负载的数据划分策略和空间运算框架的缓存机制等几个方面讨论分布式空间信息的运算技术体系, 并提出现阶段可行的实现框架。基于本框架开发的系统用于实际应用中, 较好地解决了分布式环境下的大规模复杂空间数据运算的效率问题。以分布式环境下的经典空间运算为例进行的试验表明: 该框架设计新颖, 提高了空间运算效率。

关键词: 分布式运算框架; 空间数据划分; 运算负载均衡; 地理信息系统。

中图分类号: TP311

文章编号: 1000-2383(2010)03-0362-07

收稿日期: 2010-01-15

Key Technology of Distributed Geospatial Information Operation

WU Liang¹, XIE Zhong^{1*}, CHEN Zhan-long^{1,2}, MA Li-na^{1,2}

1. Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China

2. Engineering Research Center of GIS Software and Applications, Ministry of Education, Wuhan 430074, China

Abstract: In order to improve spatial operation efficiency of massive data in distributed environment and to solve the interactive design problems of spatial analysis processing module designed to service agreement with the underlying database, spatial data models and so on. The basic characteristics of distributed computing are analyzed in this paper. The authors of this paper discuss the distributed computing spatial information technology system from the following aspects: space computing task decomposition, distributed spatial data classification method, sharing data replication strategy, the data partitioning strategy based on the load and the caching mechanism of space computing framework. Based on this framework, the authors have developed the system for solving the practical problems, and solved the efficiency of large-scale spatial data operations in the complex distributed environment. At last, the distributed computing environment test of the classic space experiments has shown that, the framework is a design novelty, and can greatly reduce the spatial computing time in the distributed environments.

Key words: distributed computing framework; spatial data partition; computing load balancing; geographic information system (GIS).

随着信息获取手段的不断改进和信息需求的日益迫切, 带来空间信息资源的急剧膨胀, 对信息处理的效率提出了更高的要求。传统单机、串行的地理信息系统 (geographic information system, GIS) 软件在进行复杂空间分析时, 其计算速度和计算效率已经无法满足海量数据、复杂分析模型的空间分析需求 (高刚毅, 2004)。而计算机网络技术的迅速发展, 使得海量空间数据分布存储于网络环境下的各个服

务器中, 分布存储和分布式处理计算变得越来越重要。国内很多学者对分布式计算进行了研究。吴信才和吴亮 (2006)、徐世武等 (2006) 以及吴信才 (2009) 提出了面向服务的基于数据中心的分布式空间信息支撑框架; 吴沉寒等 (2005) 研究了基于 SOAP 协议的分布式 GIS 模型; 方裕等 (2006) 提出的针对分布式 GIS 环境的协同计算技术研究, 对新一代协同计算的分存式 GIS 进行了软件原型设计; 蔡砥 (2004)

基金项目: 国家自然科学基金项目 (No. 40771165); 国家重点“863”项目 (No. 2007AA120503)。

作者简介: 吴亮 (1976-), 男, 博士, 副教授, 主要从事地理信息系统教学与研究工作。E-mail: wuliang133@189.cn。

* 通讯作者: 谢忠, E-mail: xiezhong68@gmail.com

提出的一种称为 Rolland 搜索算法在网格上的并行化计算方法;杨峰等(2007)研究了分布式计算环境下的 GIS 资源的混合对等发现机制。

这些研究为分布式空间分析运算的研究提供相关的理论基础,现阶段的研究主要集中在适合分布式环境的空间分析的基本运算算子、空间索引、空间数据存储等方面,但对分布式环境下的空间数据的划分策略、空间数据的高效协同计算、空间分析运算的分发回收等分布式 GIS 的关键技术问题的研究力度还不够,但这些恰恰是分布式 GIS 得以实现的关键技术问题。目前许多理论都需要结合空间数据信息分布式的特点进行进一步深化。本文将从分布式空间数据的划分策略、空间运算任务分解和回收策略、共享数据复制策略、基于负载的数据划分策略和空间运算框架的缓存机制等方面对分布式空间信息的运算技术体系作较深入地研究。

1 空间数据的划分策略

空间分析数据一般具有数据量大、数据分布广等特点,在分布式集群系统下,将这些空间分析数据进行数据分割,并结合相关的并行程序设计模式(如消息传输接口等),进行空间分析软件基础并行化研究。在进行空间软件并行化研究中,空间数据分割是实现空间数据组织管理和处理(查询、检索、统计、计算等)及空间分析功能的前提,是空间分析并行化计算的基础问题。

空间数据的划分策略要针对不同的空间对象(点、线、面)特点,采用动态的分治策略。在保证各分区内部空间数据空间邻近性的同时,各分区之间数据存储量均衡。空间数据划分的目的是为了将海量空间数据分散存储于分布式网络环境中的各 worker 节点上,使得各个 worker 节点的任务处理时间尽量相等。

地理数据,尤其是线、区数据的划分有别于传统数据的划分。一般数据库中存储的数据,无论其数据规模有多大,每个数据元组的规模由其字段结构的大小决定,即不同元组具有相等的存储大小,这就有利于将数据划分为相等大小的簇。空间上的点数据也具有这种等长特征(点的 x, y 坐标表示为 double 类型,则每个点的大小为 32 位),但空间数据库中的线、区数据不具备这样的特征。空间数据的规模取决于其几何特征和属性结构总共的存储大小,虽然同

一图层的数据具有相同的属性结构,但线、区空间数据的几何特征大小是千差万别的。同一图层中,5 个点组成的多边形和 50 000 个点组成的多边形,其存储大小相去甚远(分别是 5×32 位和 $50\,000 \times 32$ 位)。

空间数据划分问题已被证明为非确定多项式(non-deterministic polynomial, NP)问题(方裕等, 2006; 马修军等, 2006),因此,本架构中数据分发模块采用了启发式的空间数据分簇方法。具体分簇方法为本地负载均衡方法(local load-balancing, LLB),即分簇目标是使各个 worker 节点上的数据负载尽可能的接近。

在本地负载均衡空间数据划分方法中,如何表达空间数据的负载是一个关键问题。就多边形裁剪运算来说,空间对象的计算负载应该表示为 T_{sequ} ,但 T_{sequ} 的准确值在裁剪运算开始前是无法获得的。另外,当并行运算中需处理多个裁剪框时,笔者难以定义一种负载计算方法来适应大规模空间运算中的每一个裁剪框处理。根据每个裁剪框的 T_{sequ} 作为负载值来划分数据的方法又无疑会增加系统面对大型空间运算时的工作负担,降低系统的整体性能。因此,为了处理好多个裁剪框的情况,采用准确度较低,但平均性能较好的负载计算方式作为空间数据划分的指标。在数据分发模块中,负载值由空间对象的点数来表示,即:

$$l_i = \text{Load}(i) = \text{pntnum}(r_i). \quad (1)$$

本地负载均衡空间数据划分的基本思想如下:

(1)对多边形集合 R ,将前 N 个多边形分配给 N 个节点;

(2)计算每个节点的负载 W_i ,节点负载为所有待处理数据对象的计算负载总合。选择负载最小的节点 p_w ,最小负载处理器按照下面的公式选择:

$$W_{\min} = \min(W_i) = \sum_{r_j \in p_i} \text{Load}(r_j). \quad (2)$$

(3)对集合中的下一个多边形,将下个多边形分配给节点 p_w ;

(4)重复步骤(2)、(3)直到所有多边形被分配。

按照如上方法,依据并行环境中每个 worker 节点上负载量完成的空间对象静态分簇,保证了每个并行 worker 节点的负载均衡。

2 分布式空间运算策略

Master-worker 分布式计算架构(钱卫宁,

2003;付志祥,2006;邱彤庆和陈贵海,2007)将一系列空间运算软件库部署在架构内,具体包括范围查询、裁剪运算、缓冲区分析和叠置分析.该架构的数据收集与分发过程中的一个关键技术就是空间运算的任务分解策略,其中空间运算的负载均衡是关键问题.本节将以空间数据的多边形裁剪运算为模型,介绍分布式计算环境下的空间运算任务分解策略.

空间设计强调了并行环境中 GIS 运算的负载均衡,在本框架的资源检测模块中设计了两层负载均衡模型,由基于负载的数据划分(load-based data partitioning, LDP)和动态负载调度(dynamical load schedule, DLS)两阶段构成. LDP 阶段将数据按照其负载大小分布到不同 worker 节点中,使各节点负载基本均衡;再根据运行时的 worker 节点工作情况通过 DLS 机制动态调度待处理的数据.依据如上所述的两层负载均衡机制(图 1),并行处理后的结果将进行汇总并交付分析处理.

这样的数据划分虽然是静态的,但却比动态数据划分方法对整体性能的提升更有帮助.采用接收裁剪框后再数据分簇的动态划分方法,虽能保证数据划分的负载不均较低,但由于裁剪框多边形的个数不止一个,如果每次接收裁剪框后都进行数据重新分簇,无疑会大大增加系统的整体开销.采用的改进方法是先根据多边形数据的负载大小进行数据分簇,保证系统内各节点的数据负载量接近;再通过运行时第二层的负载动态调度,改善负载不均,减少系统内节点空转现象的发生,当相应的数据传送到每个 worker 节点后,进行并行化的空间运算处理.

以裁剪运算为例, master 节点将裁剪框数据广播到各个并行 worker 节点,所有 worker 节点开始并行地处理分派给它的静态本地数据,直到有 worker 节点完成了本地数据的处理,此时启动动态负载调度机制.

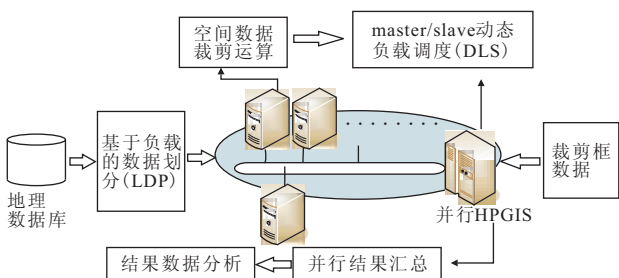


图 1 分布式环境下的空间数据裁剪运算策略

Fig. 1 The method of clipping spatial data in distributed environment

Master-worker 模式的动态负载调度机制旨在通过在分布式的并行 worker 节点间动态调度负载来改善静态数据划分阶段无法避免的负载不均,从而避免 worker 节点的空转,提高整个系统的计算能力利用率,最大程度上实现空间运算并行化带来的性能提升.

动态负载调度机制通过在 master 节点上存储并动态更新共享数据处理结构(share data solve structure, SDSS)来维护整个架构中 worker 节点的任务分派与调度,并避免节点处理任务的冲突.该共享数据处理结构表示如图 2 所示.

buk_id 表示被划分为 T 份的动态共享数据; buk_start 表示每一份动态共享数据的处理状态,可分为已筛选、已裁剪、未处理,分别表示该部分共享数据已结束了裁剪框外接矩形筛选、裁剪框裁剪处理和未作任何处理; slv_node 表示该部分共享数据的筛选或裁剪处理分别由哪个 worker 节点完成,以便 master 节点告知其他 worker 节点待处理数据在哪个 worker 节点上; done_flg 表示动态共享数据是否全部完成了处理,其初始值为 T ,每当一份动态共享数据被处理完成, done_flg 值递减.

以下简要介绍了 master-worker 模式下基于 SDSS 的动态负载调度过程.

当某个 worker 节点 p_w 完成了其上静态本地数据的处理后, p_w 节点发送消息告知 master 节点它正要处于空转状态(如图 2①、(1)).

master 节点检查 SDSS 结构 done_flg 标记(如图 2②、(2)),若 done_flg 不为 0, master 查找到未被处理的一份动态共享数据(如图 2③、(3)),将其 buk_id 传送给 p_w ,并将该份数据的 buk_start 标识为已筛选, slv_node 记录为 w (如图 2(4)), p_w 对已复制存储于其上的该部分动态共享数据进行筛选处理.

若 SDSS 结构中没有未处理的数据,则 master 查找是否存在已筛选的一份数据,其 slv_node 和 w 相同.如果相同, master 则将其 buk_id 传送给 p_w , p_w 在自身节点上找到该部分数据作裁剪处理,同时 master 将该份数据的 buk_start 标识为已裁剪(如图 2(4)).

如果 p_w 访问 master 后,发现 slv_node 和 w 值不同,则 master 将 buk_id 和完成该份数据筛选操作的节点 idslv_node 返回给 p_w ,并将 buk_start 标识为 w , buk_start 标识为已裁剪(如图 2(5)).接下来, p_w 向 slv_node 节点发送消息请求 buk_id 的筛选结果(如图 2(6)), slv_node 节点将筛选后需进一

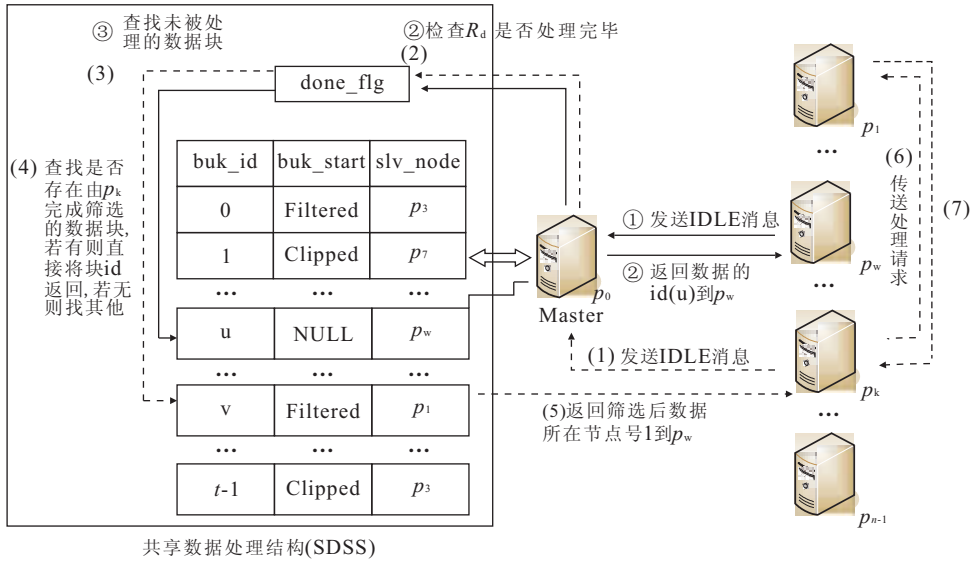


图 2 Master-worker 模式下基于 SDSS 的动态负载调度

Fig. 2 Dynamic load attemper based on SDSS in master-worker model

步裁剪的空间对象 id 返回给 p_w (如图 2(7)). p_w 在自身节点上找到该部分数据完成裁剪处理.

综上所述,以多边形裁剪运算为例,并行化空间运算两层负载均衡模型的算法伪代码表现为以下算法:

输入:

$R_i \in R = \{r_0, r_1, \dots, r_m\}$, 待处理的多边形集合;

$C_j \in C = \{c_0, c_1, \dots, c_k\}$, 裁剪框数据多边形集合;

$P_w \in P = \{p_0, p_1, \dots, p_n\}$, 构成并行环境的节点集合.

输出:

$S_h \in S = \{s_0, s_1, \dots, s_q\}$; 裁剪结果多边形集合.

节点 p_0 作为 master 节点, p_1, p_2, \dots, p_n 为 worker 节点; arrRd 数组记录了每一份 R_d 的处理情况,包括是否完成裁剪框筛选、是否完成裁剪、分别由哪个节点完成.

- 1: 数据划分为静态本地数据 R_s 和动态共享数据 R_d
- 2: LDP 划分 R_s 为 N 份 $R_s^w, 0 \leq w \leq N$
- 3: 同样方法划分 R_d 为 T 份 $R_d^u, 0 \leq u \leq M$
- 4: 数据装载入 $p_w \leftarrow R_s^w, p_w \leftarrow R_d$
- 5: while 裁剪框数据集合不为空 do
- 6: master 节点 $p_0 \leftarrow C_j$
- 7: one-to-all 广播 $p_w \leftarrow C_j$
- 8: p_w 并行处理分配到的 R_s^w
- 9: if p_w 处理完毕 R_s^w then
- 10: 回收 p_w 处理完毕的裁剪结果

- 11: while 有未作筛选处理的 R_d do
- 12: p_0 传送待筛选的 buk_id 到 p_w , 修改 SDSS
- 13: p_w 筛选处理 R_d^u
- 14: end while
- 15: while 有未作裁剪处理的 R_d do
- 16: if R_d^v 的 slv_node 为 p_w then
- 17: p_w 裁剪处理 R_d^v
- 18: else if R_d^v 的 slv_node 为 p_1 then
- 19: p_0 告知 p_w 向 p_1 申请待处理数据, 修改 SDSS
- 20: p_1 将 R_d^v 中待处理数据 id 传送给 p_w
- 21: p_w 裁剪处理 R_d^v 中待处理数据
- 22: if p_w 裁剪完毕 R_d^v then
- 23: 回收 p_w 处理完毕的裁剪结果
- 24: end while
- 25: end while

该算法详细介绍了采 master-worker 模式的负载动态调度机制,实现了 N 个 worker 节点并行处理 k 个裁剪框对 $R_i \in R = \{r_0, r_1, \dots, r_m\}$ 的空间裁剪运算.

在该算法中,消息传递避免了空间对象本身的传送,而使用 id 传送来降低通信开销,因为 id 号的传送规模比数据本身的传送规模要小的多.由于动态共享数据在各个 worker 节点上都复制存储,因此将共享数据分配给 worker 节点处理的过程,只需传递共享数据块号到指定 worker 节点.如果 worker 节点 p_w 待裁剪的数据是由 p_1 完成筛选的, p_1 将需裁剪的那部分数据的 id 号传送给 p_w .

3 分布式空间分析缓存调度策略

分布式空间分析的缓存调度采用类似最近最少使用算法(least recently used, LRU)的策略来对空间分析框架缓存进行管理. 当一个缓存中的某一页由非激活态转变为激活态, 即内存中的某一页内容需要从外存文件中载入到内存中时, 缓存调度器会在自己维护的处于激活态的内存块中选择一个将它“淘汰”, 使它转为非激活态, 将“淘汰”掉的内存页的内容写入到外存文件中, 从而释放该页内存(图 3).

4 实验分析

以空间裁剪为例对分布式环境中的空间分析运算进行了测试: 使用 8 台机器组成分布式计算环境, 具体配置内容如表 1 所示. 其中 IP 地址为 192. 168. 83. 1 为任务管理节点, 各分布式节点在进入该体系之前要向任务管理节点中的节点资源注册中心注册, 经允许后各分布式节点才可以加入分布式环境下的矢量空间分析运算服务平台中(图 4).

从物理结构角度来讲, 该分布式运算框架由多个任务管理节点和多个分布式节点构成, 节点之间通过高速局域网连通(可以方便地对本系统进行扩充, 通过广域网来连接各个分布式节点), 如图4所

表 1 服务节点配置情况

Table 1 The configure of service nodes

节点名称	IP 地址	作用
node1	192. 168. 83. 1	任务管理节点
node2	192. 168. 83. 2	分布式节点
node3	192. 168. 83. 3	分布式节点
node4	192. 168. 83. 4	分布式节点
node5	192. 168. 83. 5	分布式节点
node6	192. 168. 83. 6	分布式节点
node7	192. 168. 83. 7	分布式节点
node8	192. 168. 83. 8	分布式节点

示. 分布式节点既可以是单个的计算机也可以是一个专用的集群. 各个分布式节点可以在任务管理节点的控制下自由选择加入或退出系统, 使得系统具有物理上的可扩展性, 从而保证了系统逻辑上存储空间无限性.

每个存储节点上都安装了分布式矢量空间分析运算服务软件, 任务管理节点上安装数据库, 保存着整个体系的用户信息的记录. 在中心管理节点上安装了存储管理系统软件, 管理员可以通过该系统软件在任务管理节点主机上对整个系统以及用户进行管理. 同时任务管理节点上安装用于运行基于 Web 的图形管理界面; 集成了存储调度器, 用于做出合适的存储调度; 配置了节点资源注册模块, 用于方便分布式节点的接入. 用户可以使用浏览器通过Internet

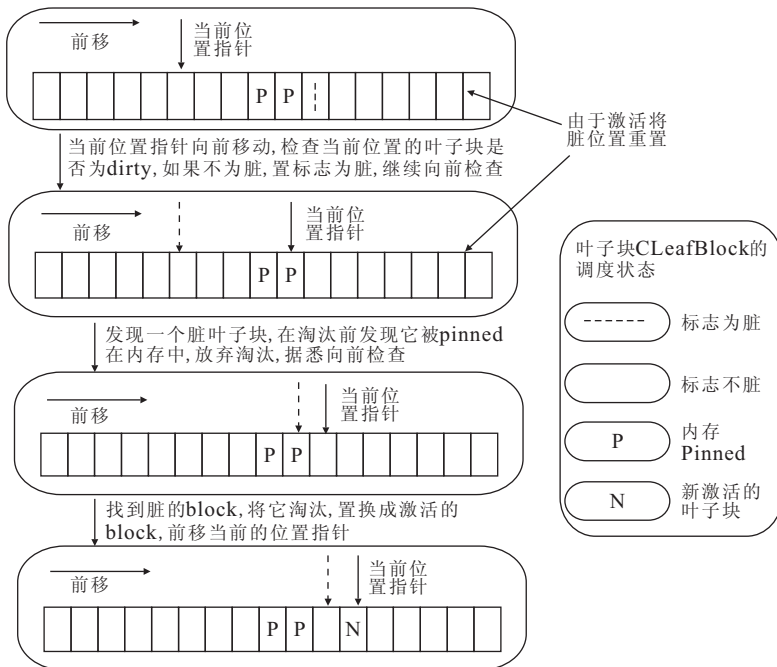


图 3 淘汰算法

Fig. 3 Washed out algorithm

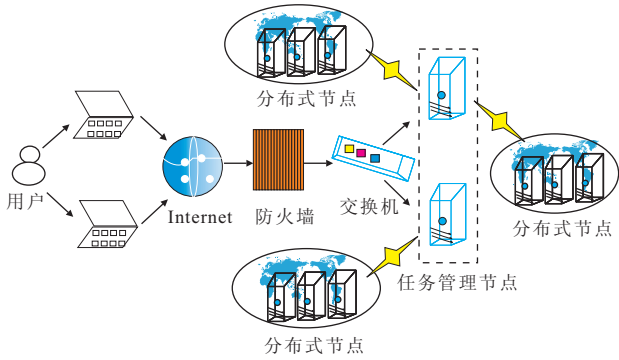


图 4 分布式矢量空间分析运算服务平台结构

Fig. 4 Structure map of workstation for distributed spatial vector data analysis

表 2 测试数据

Table 2 The data for testing

序号	内容	类型	数量
1	居民点	点	2×10^4
2	居民点	点	6×10^5
3	国道	线	5×10^4
4	省道	线	8×10^4
5	县乡道	线	3×10^5
6	等高线	线	4×10^6
7	道路	线	7×10^3
8	省域	面	3×10^2
9	县域	面	3×10^3
10	土地覆盖	面	3×10^4
11	土地利用	面	2×10^5
12	土地利用	面	3×10^6

访问 Portal 图形界面,实现对分布式矢量空间分析运算服务系统的访问,而不需安装额外的客户端软件。测试数据如表 2 所示。

以利用多边形进行空间数据裁剪为例的测试结

果如下。该测试分析利用了分布式环境中的 3 个节点,其中裁剪框数据放在 node2(192.168.83.2)中,需要裁剪的点、线、面数据分别放在 node4(192.168.83.4)和 node5(192.168.83.5)中,通过对上面的测试分析(图 5a)可以看出:裁剪的点空间实体为 20 000 时,单节点和双节点的运算时间都差不多,因为此时的数据量较小,并且点和多边形的判断相对线和多边形对多边形的判断简单,而在裁剪线实体(图 5b)和裁剪面实体(图 5c)过程中,即使线和面的个数只有 10 000 个,单节点和双节点运算所得的时间也是有差距的。例如在图 5c 中,裁剪 1 000 个多边形单节点运算时间为 20 s,而双节点的时间为 12 s。当空间实体的规模变大时,利用双节点所运算的时间将大大减小,例如在图 5c 中,空间面实体的个数达到 10 000 时,单节点运算时间需要 100 s,双节点运算时间需要 56 s。利用双节点并行计算可以提高将近 1 倍的运算效率。

6 结语

该研究进行了原型设计,主要目标是在现有的矢量空间数据运算的基础上,研究分布式环境下的矢量空间运算模型,充分利用分布式环境中的计算资源对空间数据库中的矢量数据进行深加工。

目前基于国产 GIS 软件—MapGIS K9 已经实现了分布式空间运算 GIS 软件系统原型,包括分布式结构化查询语言、空间资源发现模块及其分布式并行空间索引机制等。本文在已有的研究基础上提出了适合于分布式空间分析运算的框架,深入地研究了分布式环境下空间运算任务分解和空间数据划

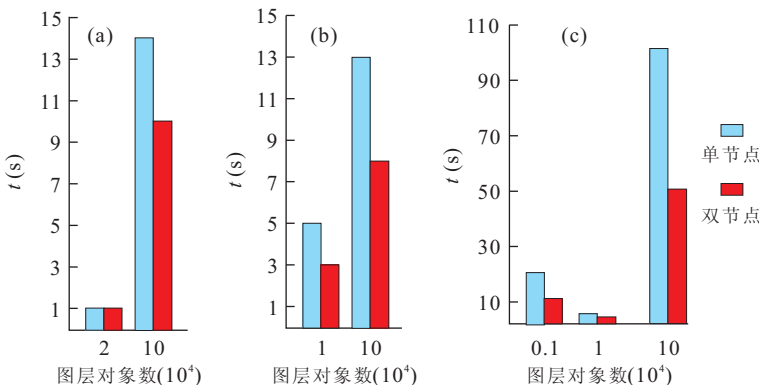


图 5 裁剪空间实体

Fig. 5 Clipping spatial data

a. 裁剪点空间实体; b. 裁剪线空间实体; c. 裁剪面空间实体

分的方法、共享数据复制策略、基于负载的数据划分策略和空间运算框架的缓存机制等,并以分布式环境下的空间裁剪运算为例进行了试验.初步解决了分布式空间分析运算框架的关键技术问题,但软件原型的实现与实用还有一定的距离,需进一步完善空间数据分布式空间分析运算框架的细节.

References

- Cai, D., 2004. A study on the computing models for spatial analysis in the GRID computing environment (Dissertation). East China Normal University, Shanghai (in Chinese).
- Fang, Y., Wu, L., Xie, K. Q., et al., 2006. Research on distributed and cooperating GIS. *Geography and Geo-Information Science*, 22(3): 9—12, 54 (in Chinese with English abstract).
- Fu, Z. X., 2006. Research and implement of master-worker computation in grid (Dissertation). Fuzhou University, Fuzhou (in Chinese).
- Gao, G. Y., 2004. Distributed geographic information system (Dissertation). Zhejiang University, Hangzhou (in Chinese).
- Ma, X. J., Liu, C., Xie, K. Q., et al., 2006. A research on global spatial data directory in peer-to-peer networks. *Geography and Geo-Information Science*, 22(3): 22—25 (in Chinese with English abstract).
- Qian, W. N., 2003. Data management in peer-to-peer systems (Dissertation). Fudan University, Shanghai (in Chinese).
- Qiu, T. Q., Chen, G. H., 2007. A generic approach to making P2P overlay network topology-aware. *Journal of Software*, 18(2): 381—390 (in Chinese with English abstract).
- Wu, C. H., Meng, L. K., Deng, S. J., 2005. A new type of distributed GIS model. *Computer Engineering and Applications*, 8: 207—209 (in Chinese with English abstract).
- Wu, X. C., 2009. Datacenter integration development technology: the next generation GIS architecture and development model. *Earth Science—Journal of China University of Geosciences*, 31(5): 624—630 (in Chinese with English abstract).
- Wu, X. C., Wu, L., 2006. Service-oriented distributed spatial information supporting system. *Earth Science—Journal of China University of Geosciences*, 31(5): 585—589 (in Chinese with English abstract).
- Xu, S. W., Xie, Z., Huang, Z. C., 2006. Research and design of isomerism distributed multilevel spatial data center. *Earth Science—Journal of China University of Geosciences*, 31(5): 624—630 (in Chinese with English abstract).
- Yang, F., Li, F. X., Yu, H. L., et al., 2007. A hybrid peer-to-peer lookup service algorithm on distributed hash table. *Journal of Software*, 18(3): 714—721 (in Chinese with English abstract).

附中文参考文献

- 蔡砥, 2004. 网络计算环境下空间分析的计算模式研究(博士论文). 上海: 华东师范大学.
- 方裕, 邬伦, 谢昆青, 等, 2006. 分布式协同计算的 GIS 技术研究. *地理与地理信息科学*, 22(3): 9—12, 54.
- 付志祥, 2006. 网络环境下 Master—Worker 计算的研究和实现(硕士论文). 福州: 福州大学.
- 高刚毅, 2004. 分布式地理信息系统研究(博士论文). 杭州: 浙江大学.
- 马修军, 刘晨, 谢昆青, 等, 2006. P2P 环境中的全局空间数据目录研究. *地理与地理信息科学*, 22(3): 22—25.
- 钱卫宁, 2003. 对等计算系统中的数据管理(博士论文). 上海: 复旦大学.
- 邱彤庆, 陈贵海, 2007. 一种令 P2P 覆盖网络拓扑相关的通用方法. *软件学报*, 18(2): 381—390.
- 吴沉寒, 孟令奎, 邓世军, 2005. 一种新型的分布式 GIS 模型. *计算机工程与应用*, 8: 207—209.
- 吴信才, 2009. 数据中心集成开发技术: 新一代 GIS 架构技术与开发模式. *地球科学——中国地质大学学报*, 34(3): 540—546.
- 吴信才, 吴亮, 2006. 面向服务的分布式空间信息支撑平台. *地球科学——中国地质大学学报*, 31(5): 585—589.
- 徐世武, 谢忠, 黄志超, 2006. 分布式异构多级空间数据中心的研究与设计. *地球科学——中国地质大学学报*, 31(5): 624—630.
- 杨峰, 李凤霞, 余宏亮, 等, 2007. 一种基于分布式哈希表的混合对等发现算法. *软件学报*, 18(3): 714—721.