

基于 MapReduce 的多机并行 DP 算法与实验分析

张栋海^{1,3}, 黄丽娜^{2,3}, 刘 晖¹, 唐 健¹

(1. 武汉大学卫星导航技术研究中心, 武汉 430079; 2. 武汉大学资源与环境科学学院, 武汉 430079;

3. 武汉大学地理信息系统教育部重点实验室, 武汉 430079)

摘要: 随着网络地图不断发展, 个性化网络地图也得到快速发展。个性化网络地图需要以矢量数据为数据基础, 以满足人们对地图色彩、符号等个性化要求, 所以需要实时、快速进行大量数据化简。本文以经典 Douglas-Peucker 算法作为曲线化简算法, 利用开源云计算平台 Hadoop 建立多机协作的曲线并行化简服务框架, 设计和实现了多机并行 Douglas-Peucker 算法, 并在集群上进行实验分析, 验证算法的效率和适用性。算法核心是设计数据的逻辑分片, 利用 MapReduce 计算原理, 将分片分配到集群中, 实现并行运算。实验分别分为两个方面: (1) 比较在固定阈值不同数据量情况下, 传统 DP 算法与多机并行 DP 算法效率; (2) 比较在相同数据量不同阈值情况下, 传统 DP 算法与多机并行 DP 算法效率。实验表明, 在大数据量和高复杂度情况下, 多机并行 DP 算法的效率更高。

关键词: 多机并行 DP 算法; Douglas-Peucker 算法; 曲线化简; MapReduce

DOI: 10.3724/SP.J.1047.2013.00055

1 引言

近年来, 云计算与 GIS 相结合成为了 GIS 发展的新趋势^[1], 网络环境下充分调用云中各种算力资源可为海量地图数据的实时处理提供有力的技术支撑。目前, 网络地图以数据基础为依据可划分为两种: (1) 以预先生成的地图图片为数据基础的网络地图, 以空间换时间, 图片一旦生成无法实时根据用户需求更改图片中相关地物的色彩, 也无法分层显示, 因而难以提供个性化的地图服务; (2) 网络地图以矢量数据为基础, 可以根据用户需求更改地物颜色, 显示图层等, 可以提供个性化地图服务, 但是时间响应较慢, 特别是在数据量较大情况下, 难以快速、实时地提供可用的地图数据处理结果, 所以难以广泛使用, 而在后台数据处理过程中, 矢量数据化简是直接影响其响应速度的关键步骤。因此, 本文以开源的云计算平台 Hadoop 作为集群运算平台, 着重探索网络环境下如何进行地图线状数据化简服务的快速响应。

在制图综合领域, 线要素化简算法很多^[2-7], 其中, Douglas-Peucker 算法是较为经典的线要素化

简算法^[8], 国内外众多学者针对 DP 算法进行了改进^[9-11], 但在算法效率上进行改进的方案较少。传统 DP 算法处理效率较低, 难以实时对大量矢量数据进行化简, 因此, 本文提出并行 DP 算法, 通过数据分片, 进行并行化简。并行 DP 算法根据并行计算原理可以分为单机多线程并行 DP 算法和多机并行 DP 算法。对于第一种实现策略, 单机多线程并行计算, 利用现在多核 CPU 的运算能力, 使用多线程编程技术, 将任务分为若干子任务, 每个线程运算其中的一个或者多个子任务, 最终达到并行计算目的^[12]; 对于第二种实现策略, 多机并行计算, 通过网络连接不同的计算机, 形成计算集群, 将计算任务分解成若干个子任务, 多台计算机同时进行计算, 每台计算机计算其中的一个或者多个子任务^[13-14]。单机多线程计算虽然能够很好地提高计算效率, 但是, 由于一台电脑的总运算能力有限, 所以这种并行 DP 算法性能拓展受到限制。多机并行运算相比于单机多线程并行运算具有良好的扩展性, 可以通过增加服务器数量提高计算效率。因此, 本文研究设计并实现了多机并行 DP 算法, 同时构建了分布式运行环境, 对算法进行检验。本研究

收稿日期: 2012-09-13; **修回日期:** 2012-12-17.

基金项目: 国家自然科学基金项目(41101448, 51008138); 中央高校自主科研项目(274737); 中国博士后科学基金项目(2011M501230)。

作者简介: 张栋海(1989-), 男, 河北承德人, 硕士研究生, 研究方向为 DEM 地图制图自动综合。E-mail: zdh_zhangdonghai@163.com

在 Hadoop 平台基础上针对 GIS 数据,结合 MapReduce 编程模式^[15-18],设计多机并行 DP 算法,并将其与串行 DP 算法进行对比分析。

2 多机并行 DP 算法设计

多机并行 DP 算法设计与实现的关键:(1)如何设计适合 MapReduce 计算模式的 DP 算法;(2)所设计的算法能够最优地利用这种分布式计算模式,制定合适的计算步骤,减少没用的计算开销;(3)设计适合分布式运算的文件格式,简化编码;(4)设计文件分片方法,提高集群运算效率。本文针对第一个问题和第二个问题设计了多机并行 DP 算法模型,每次输入 Map 阶段的数据是一条完整线数据,同时放弃 Reduce 计算,在 Map 阶段就完成化简计算;针对第三个问题自定义矢量数据结构,将属性数据和空间数据分开存储,简化数据结构;针对第四个问题设计合理的逻辑分片方法。

2.1 多机并行 DP 算法模型设计

算法将数据进行逻辑分片,每个计算节点获取处理程序,同时对本地数据进行处理,达到线要素并行化简目的。算法在 Map 阶段进行并行运算,完成数据化简。如果节点间的运算效率差别较大,例如,节点 A 在一段时间可以完成 3 片数据的处理,节点 B 在相同时间只能完成 1 片,那么节点 B 的一部分分片可能会交由节点 A 处理,类似这种任务调度由底层平台自动完成。图 1 是在理想状态下多机并行 DP 算法线要素化简模型,节点的运算效率一致,每片数据都在本地计算。Map 阶段是完成数据处理的关键阶段,一次输入一条线数据数据,通

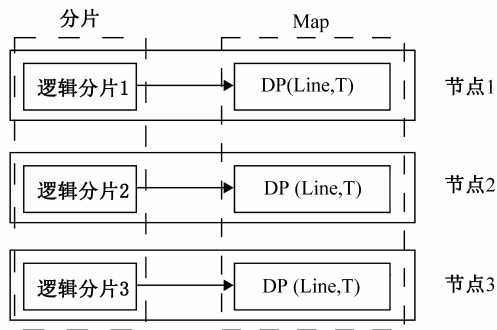


图 1 多机并行 DP 算法线要素化简模型

Fig. 1 Multi-machine parallel DP algorithm simplifies line model

过 DP 算法对这条线数据进行化简,将化简后的点坐标转换成字符串格式,直接输出,如表 1。

表 1 Map 阶段数据
Tab. 1 The data of Map

输入	key(LineID)	value(PointList)
计算	DP(PointList, T), T 为阈值	
转换	结果转换成字符串格式, Text	
输出	key(NullValue)	value(result)

DP(Line, T)函数中的 T,表示阈值,阈值越大,算法递归的次数越少,程序处理的速度越快,数据压缩比越大;反之,处理速度越慢,压缩比越小。阈值的变化与处理时间变化之间并不是线性关系,在实验中会详细比较不同阈值下多机并行 DP 算法和传统 DP 算法处理效率。

2.2 数据逻辑分片

数据本身以一定的分块大小(预定义 64M)存储在若干个 DataNode 节点,在对数据进行 Map 处理之前,可根据自身需要,自定义每个 Map 作业的数据。数据以硬分割方式存储,每个分片并不保证数据的完整性。例如,一条线数据包含的若干个分片,可能被存储在不同的分片上,所以,逻辑分片不仅可以用来自定义每个 Map 作业数据大小,而且可以保证计算数据完整。

分片的原则是:保持每片数据中包含的线数据完整;保证每个分片最接近 64M。前者保证数据完整性;后者是为了尽可能使每个逻辑分片尽量只包含一个物理分片数据,每个 Map 尽可能的处理本地分片数据,减少网络间传输的数据量,提高运行效率。所以,算法中每个逻辑分片最接近 64M,每个分片内部包含若干条完整线数据,如图 2 所示。

2.3 算法实现流程

图 3 为算法流程图,整个算法有两个阶段最为关键:(1)数据的分片;(2)Map 阶段数据化简,这两个阶段前面已经详细介绍。A 阶段动态分配 Job,保证每个节点均参与运算,都在处理 Job,有多个 Map Job 同时在多个节点运行。每个 Map Job 处理一个逻辑分片,每次化简一条线数据,作业会不断有线数据输入,直到这个 Map Job 完成。直到所有的 Map Job 全部完成计算,整个运算过程结束。

算法在 Map 阶段完成并行运算,由于并不需要

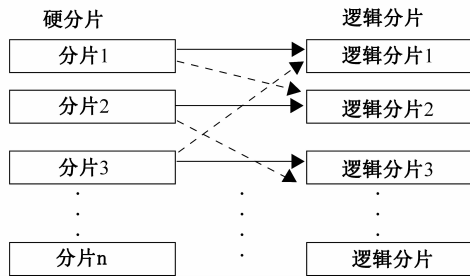


图 2 数据从物理分片到逻辑分片示意图

Fig. 2 Instruction of the process of data slicing in logic

MapReduce 计算模式中的 Reduce 阶段,所以,算法中并不涉及 Reduce 阶段,这样可以减少数据在网络间传输所用时间和排序分组所产生的时间开销,提高计算效率。

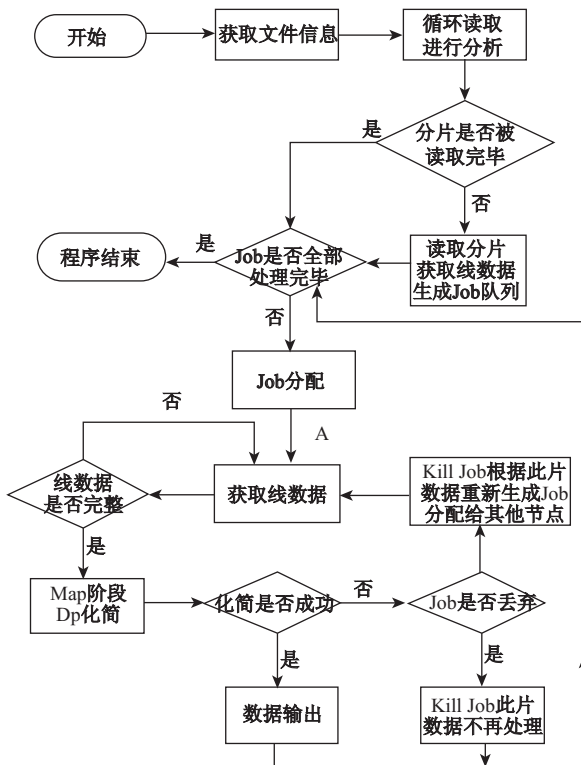


图 3 多机并行 DP 算法流程图

Fig. 3 Flow chart of multi-machine parallel DP algorithm

3 算法的实验分析

3.1 实验硬件环境配置

由于条件限制,本实验环境的三台服务器性能差别较大,基本可以满足实验要求。第一台服务器相比于其他服务器的能力强大的多,所以,这台服

务器不仅作系统运行主服务器,还要虚拟 3 个节点,1 个主节点(NameNode),2 个从属节点(Slave-Node)。其他 2 台机器配置较低,所以,只虚拟一个从属节点,在下面分析中会涉及到硬件配置问题。实验中单机 DP 运算的预算环境是第一台 CPU 为 i7 的电脑,因此,单机硬件环境要优于多机并行运算环境。

表 2 实验硬件环境

Tab. 2 Experimental hardware environment

用途	使用情况	处理器	内存	硬盘空间	网卡
2 个从属节点					
Hadoop+ 服务器	1 个主节点 系统运行服务 器节点	Intel Core i7-920	3G	100G	千兆网络 适配卡
Hadoop	1 个从属节点	Intel Pentium 双核	1G	20G	千兆网络 适配卡
Hadoop	1 个从属节点	Intel Pentium 单核	1G	20G	千兆网络 适配卡

3.2 算法实验结果

实验 DEM 格网数据来自于“国际科学数据服务平台”,数据名称 ASTGTM_N26E098Y_DEM,使用 ArcGIS 3DAnalyst 利用格网 DEM 生成不同密度等高线,等高线的密度由等高距控制,如表 3 所示。

表 3 实验数据

Tab. 3 Experimental data

数据名称	等高距(m)	数据大小
10Data	10	2.35G
20Data	20	1.17G
40Data	40	603M
100Data	100	241M
200Data	200	120M

设置化简阈值为 10m,在不同数据量情况下多机并行 DP 算法效率与非并行 DP 算法化简效率的实验结果如图 5 所示。以数据量大小为横轴,以化简时间为纵轴,虚线代表并行 DP 算法处理效率,实线代表非并行 DP 算法处理效率。对比多机并行 DP 算法效率与非并行 DP 算法效率可知,本实验环境下,临界点在 240M 至 500M 之间:小于临界点,多机并行 DP 算法的效率要低于非并行 DP 算法效

率;大于临界点,多机并行 DP 计算的效率明显提升。图中多机并行 DP 计算时间曲线曲率随着数据量的增加逐渐小于单机非并行 DP 计算时间曲线曲率,这体现了在处理大数据量时,集群并行计算的优点。

分别设曲线化简阈值为(m):10、15、20、30、40、60、100,化简的部分结果如图 4(b)、图 4(c)所示。

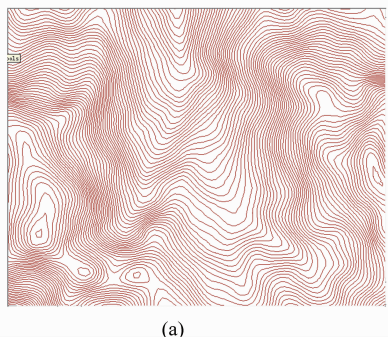


图 4(a) 未化简数据
Fig. 4(a) Raw data

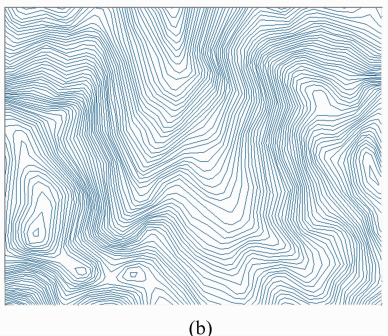


图 4(b) 10m 阈值化简结果
Fig. 4(b) DP algorithm simplified result by 10m threshold

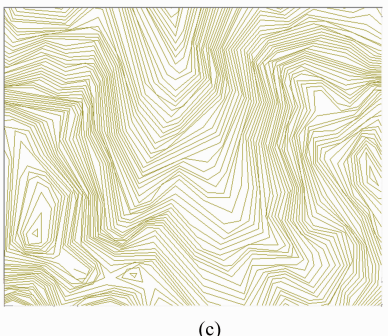


图 4(c) 30m 阈值化简结果
Fig. 4(c) DP algorithm simplified result by 30m threshold
因 DP 是对单根线进行化简,所以,有可能出现相邻等高线相交和同根等高线自相交的问题^[19],但本文

的研究重点是如何实现多机协同的曲线化简服务,针对这一问题,可以约束 Delaunay 剖分和自适应单调链,将等高线拆分为若干子曲线,得到的每条子曲线能用 DP 算法进行化简,并保证不相交^[20]。在实际应用中,也可通过对化简后的曲线进行拓扑检验,解决这个问题。从计算复杂度角度对多机并行 DP 算法效率与非并行 DP 算法效率进行比较分析(如图 6 所示),计算复杂度增加与阈值变化是非线性关系,对于这套数据,阈值小于 20m 以后,计算量迅速增加。相同数据量,阈值越小,计算复杂度越高,多机并行 DP 算法效率相比于单机非并行 DP 算法效率越高,时间曲线的斜率小于单机非并行运算。分析原因:阈值较大情况下,并行运算节点处理每一片数据所用的时间很短,数据的网络传输时间开销影响明显,所以,运算效率难以提高;阈值较小情况下,每一片的处理时间较长,网络传输时间开销所占比重较小,节点的运算效率得以充分发挥,效率得到显著提高,在阈值为 10m 处效率提高 29%。

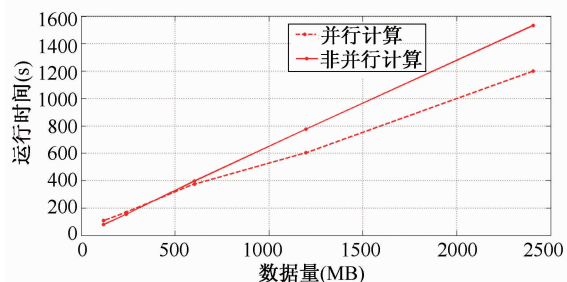


图 5 并行和非并行算法在阈值为 10m 时
不同数据量数据运算效率对比图
Fig. 5 Efficiency comparison chart of parallel and non-parallel simplifying on different amounts of data by 10m threshold

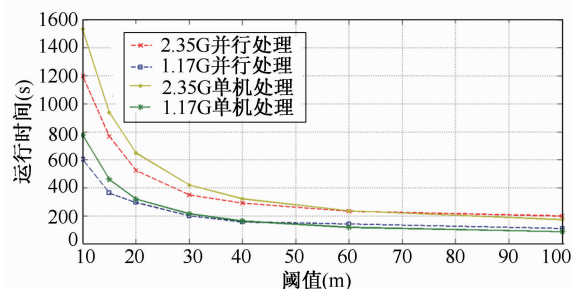


图 6 并行和非并行算法在较大数据量
不同计算复杂度时运算效率对比图
Fig. 6 Efficiency comparison chart of parallel and non-parallel simplifying on large scale data by different thresholds

图 7 为多机并行 DP 算法在处理 2.35G 未分片数据时的节点运算占比图,由图 7 可见 SlaveNode1 和 SlaveNode2 承担了 80% 的运算量,SlaveNode3 和 SlaveNode4 只承担了 20% 的运算量。如表 2 所示,SlaveNode1 和 SlaveNode2 被安装在第一台电脑上,其他 2 个节点分别安装在第二台和第三台电脑上,整个集群环境差异很大。又因为原始非并行 DP 算法运算在第一台电脑上,环境非常优越,造成多机并行 DP 算法的效率相对原始 DP 算法的效率在提升最大处,也只提高 29%。所以,在配置集群时要合理配置硬件,才能发挥集群的效率。

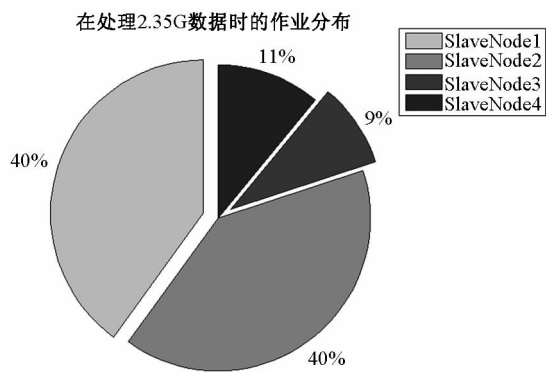


图 7 节点运算量对比图

Fig. 7 Instruction of nodes operation comparison

3.3 结果分析

(1)MapReduce 计算模式的核心是数据并行,针对这种并行运算模式,每个 Map 在一台机器上运行,一个 Map 最佳的情况是处理本地数据,可以减少数据网络传输和磁盘 IO,所以,算法的关键是制定合适的数据处理策略,与原来算法有效结合。鉴此,数据逻辑分片应接近物理分片大小,保证数据处理本地化。

(2)MapReduce 计算模式注重数据并行,在每个 Map 计算过程中之间是无法通信的,只有在 Map 阶段计算结束后可以把不同 Map 结果合并。所以,MapReduce 计算模式适用于计算时单个数据之间弱相关或者不相关,这样的并行算法处理效率会更高。

(3)针对空间数据的计算往往是数据集之间的相关运算,一些运算是否适合采用这种并行运算的方式,需要根据算法对数据处理时的实现方式,如果算法需要对全局数据统一进行处理,显然很难做到数据并行。针对大规模、大范围的数据,可以将数据范围分解,每个区域作为一个数据集,在一个

Map 中针对这个区域数据进行空间运算。

(4)从实验可知,本文设计的多机并行 DP 算法适合数据量大、计算复杂度高的数据处理,对于小数据量和低复杂度的计算优势并不明显,这也是 MapReduce 计算模式的特点。

(5)试验中的计算机集群运算效率与单机情况下相比运算效率提高并不明显,原因在于集群中计算机之间性能差异较大,整体性能无法充分发挥,所以,在配置集群时要尽可能减少机器间性能的差异,选择性能接近的计算机构建集群。

4 结语

以矢量数据为基础的多机地图服务,面临大量数据实时化简问题,本文提出的多机并行 DP 算法可有效地减少数据化简时间,有良好拓展性。实验结果表明,本算法适合处理大数据量和计算复杂度高的数据,针对小数据量或计算复杂度低的数据化简效率提升并不明显。本算法的并行实现模式,可为设计其他空间数据并行运算算法提供参考。本研究没有涉及自行控制底层数据调度,选择最优计算节点,没有根据计算量选择并行还是非并行运算,今后将从这两个方面对算法进行完善。

参考文献:

- [1] 蔡晓桦. 云计算及其在地理信息系统中的应用[J]. 江西测绘, 2012(1): 39 - 41.
- [2] 郭庆胜. 线状要素图形综合的渐进方法研究[J]. 武汉测绘科技大学学报, 1998, 23(1): 52 - 56.
- [3] Li Z L, Openshaw S. 基于客观综合自然规律的线状要素自动综合的算法[J]. 武测译文, 1994(1): 49 - 58.
- [4] Buttenfield B P. Digital definitions of scale-dependent structure[J]. Auto-Carto, 1986, 497 - 506.
- [5] 艾廷华, 郭仁忠, 刘耀林. 曲线弯曲深度层次结构的二叉树表达[J]. 测绘学报, 2001(4): 343 - 348.
- [6] 应申, 李霖. 基于约束点的曲线一致性化简[J]. 武汉大学学报·信息科学版, 2003(4): 488 - 491.
- [7] 任海艳, 陈飞翔. 一种基于遗传算法的曲线化简方法[J]. 测绘通报, 2012(10): 32 - 35.
- [8] Douglas D H, Peucker T K. Algorithms for the reduction of the number of points required to represent a digitized line or its character[J]. The Canadian Cartographer, 1973, 10(2): 112 - 123.
- [9] 毋河海. 基于多叉树结构的曲线综合算法[J]. 武汉大学学报·信息科学版, 2004(6): 479 - 483.
- [10] 张胜, 朱才连, 钟世明. Douglas-Peucker 算法的改进及

- 应用[J]. 武汉理工大学学报·交通运输与工程版, 2005(5):671-674.
- [11] 陈轶,彭认灿,郑义东,等. 基于 Douglas 双侧多叉树的曲线综合算法研究[J]. 测绘学报, 2010(3):310-315.
- [12] 马劲松,沈婕,徐寿成. 利用 Douglas-Peucker 并行算法在多核处理器上实时综合地图线要素[J]. 武汉大学学报·信息科学版, 2011(12):1423-1426.
- [13] Lin C, Snyder L. Principles of parallel programming [M]. Addison Wesley Longman, 2008.
- [14] Andrade D, Fraguera B, Brodman J, et al. Task-parallel versus data-parallel library-based programming in multicore systems[C]. Euromicro International Conference on Parallel, 2009.
- [15] White T. Hadoop: The definitive guide (Second Edition)[M]. O'REILLY®, 2011.
- [16] Lam C. Hadoop in action[M]. Manning Publications, 2010.
- [17] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters[C]. OSDI, 2004.
- [18] 王鹏. 走近云计算[M]. 北京:人民邮电出版社, 2009.
- [19] Wu S T, Marquez M R G. A non-self-intersection Douglas-Peucker algorithm[C]. Proceedings of the Sixteenth Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAP), Brazil, 2003, 60-66.
- [20] 张传明,潘懋,吴焕萍,等. 保持拓扑一致性的等高线化简算法研究[J]. 北京大学学报(自然科学版), 2007(2):216-222.

Research on Multi-machine Parallel DP Algorithm Based on MapReduce

ZHANG Donghai^{1,3} and HUANG Lina^{2,3}

(1. Research Center of GNSS, Wuhan University, Wuhan 430079, China;

2. School of Resource and Environmental Science, Wuhan University, Wuhan 430079, China;

3. Key Laboratory of Geographic Information System, Ministry of Education, Wuhan University, Wuhan 430079, China)

Abstract: Real time and rapid simplification of large-scale data, required by personalized WebGIS service which is based on vector data, becomes more and more important. The study was based on Douglas-Peucker, one of classical curve simplification algorithms, but in the view of its low performance, it can hardly simplify large-scale data in real time and rapidly. At the same time, the development of cloud-computing offers new storage technologies and computational methods for real time and rapid simplification of large-scale data. So this study made use of hadoop, one of the open source cloud computing platforms, to design and realize multi-machine parallel Douglas-Peucker algorithm. In the algorithm, we deigned the logic slices of data, and assigned the slices to the clusters by MapReduce computing model, achieved parallel simplification. In order to verify the efficiency of the algorithm, we designed the experiments and compared the efficiency of traditional DP algorithm and multi-machine parallel DP algorithm in tow aspects: 1) the same threshold and different amount of data; and 2) the fixed amount of data and different thresholds. The result of the experiments showed: the multi-machine parallel DP algorithm was more efficient than tradition DP algorithm for large-scale data and high-complexity computing. In this case, the data processing time was much longer than the data allocated in the inter-cluster and the transmission time, and every node was involved in a certain operation, improved the efficiency of operations. But for small scale data and low-complexity computing, the advantage of multi-machine parallel DP algorithm was non-obvious. Mainly due to a part of the nodes didn't participate in the operation, the computing potential of the cluster was not full play, while the data processing required time was very short, so the data allocation and transmission time impacted obviously. And, in order to meet the real time and rapid simplification of large-scale data, the multi-machine parallel DP algorithm should choose the appropriate simplification method for different amount of data and complexity computing in future.

Key words: multi-machine parallel DP algorithm; Douglas-Peucker algorithm; curve simplification; MapReduce