

时空大数据背景下并行数据处理分析挖掘的进展及趋势

关雪峰,曾宇媚*

(武汉大学测绘遥感信息工程国家重点实验室,武汉 430079)

摘要:随着互联网、物联网和云计算的高速发展,与时间、空间相关的数据呈现出“爆炸式”增长的趋势,时空大数据时代已经来临。时空大数据除具备大数据典型的“4V”特性外,还具备丰富的语义特征和时空动态关联特性,已经成为地理学者分析自然地理环境、感知人类社会活动规律的重要资源。然而在具体研究应用中,传统数据处理和分析方法已无法满足时空大数据高效存取、实时处理、智能挖掘的性能需求。因此,时空大数据与高性能计算/云计算融合是必然的发展趋势。在此背景下,本文首先从大数据的起源出发,回顾了大数据概念的发展历程,以及时空大数据的特有特征;然后分析了时空大数据研究应用产生的性能需求,总结了底层平台软硬件的发展现状;进而重点从时空大数据的存储管理、时空分析和领域挖掘3个角度对并行化现状进行了总结,阐述了其中存在的问题;最后指出了时空大数据研究发展趋势。

关键词:时空大数据;高性能软硬件;并行空间分析;数据挖掘;进展及趋势

1 引言

随着互联网、物联网和云计算的高速发展,数据获取手段向多元化方向发展,数据种类不断多样化,促使时空相关的数据呈现出“爆炸式”增长的趋势,时空信息与大数据的融合标志着正式进入时空大数据时代。时空大数据除具备大数据典型的“4V”特性外,还具备对象/事件丰富的语义特征和时空维度动态关联特性。对时空大数据进行处理、分析和挖掘得到蕴含的复杂特征是其核心价值所在(李德仁,马军等,2015)。

在时空大数据时代中机遇与挑战并存。一方面是时空数据量和类型的丰富,弥补了数据缺乏的不足,能够在最大程度上满足各类研究的需求,进一步推动交叉研究的不断深入;另一方面是面对时空大数据时空特征的特殊性,时空对象、事件等要素的动态演化以及相互间的动态关联关系对数据

管理和分析带来了极大的挑战。在存储管理方面,以往集中式存储严重依赖单机性能,极大限制了存储能力的可扩展性,无法支撑海量非结构化数据低延迟存取高并发访问。在处理分析方面,以往串行分析算法已无法满足海量时空数据的实时处理需求,不能充分发挥当前新型硬件构架和并行模型/框架的优势。在数据挖掘方面,传统的数据挖掘算法大多是基于常规数据集实现,推广到TB级别甚至是PB级别数据时,其计算效率低、扩展性能差的不足就会显现。因此时空大数据与高性能计算/云计算融合是必然的发展趋势,通过两者融合从而进一步提升时空大数据的利用效率,能更好地为研究应用服务。

针对上述问题,本文基于时空大数据背景,对现有时空大数据存储管理、时空分析和领域挖掘进行了全面的总结和阐述。首先,从时空大数据的概念和起源出发,介绍了大数据的分类和特点,分析

收稿日期:2018-08-31;修订日期:2018-10-13。

基金项目:国家自然科学基金项目(41301411) [Foundation: National Natural Science Foundation of China, No.41301411]。

作者简介:关雪峰(1980-),男,湖北松滋人,副教授,研究方向为高性能地理计算,E-mail: guanxuefeng@whu.edu.cn。

通讯作者:曾宇媚(1994-),女,广东茂名人,硕士研究生,研究方向为地理信息建模,E-mail: zengyumei@whu.edu.cn。

引用格式:关雪峰,曾宇媚. 2018. 时空大数据背景下并行数据处理分析挖掘的进展及趋势[J]. 地理科学进展, 37(10): 1314-1327. [Guan X F, Zeng Y M. 2018. Research progress and trends of parallel processing, analysis, and mining of big spatiotemporal data[J]. Progress in Geography, 37(10): 1314-1327.]. DOI: 10.18306/dlkxjz.2018.10.002

了时空大数据的固有特征。在此基础上总结了现有的高性能计算平台软硬件的发展现状,包括硬件架构、并行计算模型/框架以及各自优势对比。然后,全面总结了现阶段时空大数据的存储管理模式、并行分析策略和数据挖掘算法的并行化实现,并认为并行化是支撑时空大数据进行高效分析处理的重要手段。最后,探讨了时空大数据时代下分布式存储管理与并行处理分析当前发展趋势。

本文所介绍的时空大数据并行处理分析的软硬件环境及研究应用现状见图1。

2 时空大数据

2008年,Nature杂志在其发表的一篇文章“big data: Wikiomics”中首次提出了“大数据”这一名词(Waldrop, 2008)。2011年,Science杂志出版专刊“dealing with data”,探讨了如何借助宝贵的数据资产推动人类社会向前发展(Hong et al, 2011)。2012年,美国针对大数据的发展热潮正式启动了一项“大数据研究和发展计划”,以期在从大数据中获取知识方面有所突破。2015年,中国国务院也印发了

《促进大数据发展行动纲要》,纲要中基于全球大数据发展迅速和大数据广泛应用于各个领域的现状,提出了中国未来在大数据的发展规划中要加快数据共享、提高管理水平等任务。

迄今为止,大数据科学已经发展为一门新兴的综合性学科。对于“大数据”,普遍认为它是数据体量(volume)大、数据类型(variety)多、产生速度(velocity)快和价值(value)含量高的数据集合。而时空大数据,则是指与时空位置相关的一类大数据,是时空信息与大数据的融合。日常生活中带有时间与位置标签的数据十分常见,人类生活中所产生的数据约有80%和时空位置有关(Xu, 1999)。2011年,麦肯锡环球研究院Manyika等(2011)发布了报告“big data: the next frontier for innovation, competition, and productivity”,报告提出医疗保健、零售、公共领域、制造业和个人位置这五大类数据组成了当前主要的大数据流,而这些数据都具有显著的地理编码和时间标签。因此,如何高效处理分析时空大数据是当前学术界研究的热点问题之一。

从感知对象角度,时空大数据可以划分为以下2类:

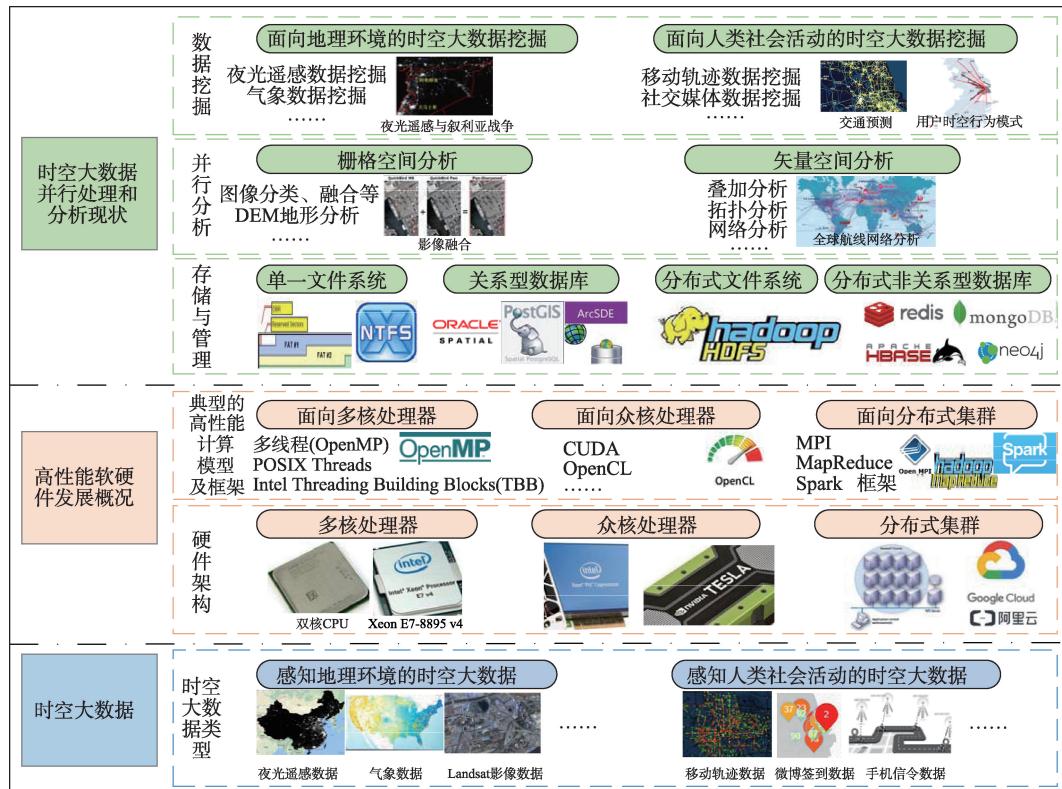


图1 时空大数据并行处理分析的软硬件环境及研究应用现状

Fig.1 Software/hardware environment and research status quo of parallel processing and analysis on big spatiotemporal data

(1) 感知地理环境的时空大数据

随着对地观测技术的发展,各类遥感数据成指指数级增长并逐步积累,成为一类典型的时空大数据,即“遥感大数据”。随着遥感云平台的建设,各类遥感数据服务与处理服务逐渐被发布出来,地理服务从专业走向大众,用户无需搭建专用环境就可以方便地应用遥感大数据。2018年2月,中国科学院正式启动A类战略性先导科技专项“地球大数据科学工程(CASEarth)”。它的目标是建成具有全球影响力的、开放性的国际地球大数据科学中心,逐渐突破技术瓶颈,形成资源、环境、生态等多学科领域融合、独具特色的地球大数据云服务平台,肩负起国家宏观决策与重大科学发现的重任。

(2) 感知人类社会活动的时空大数据

随着互联网技术、社交媒体平台的不断发展和进步,人类活动每时每刻都会产生大量的时空数据,具有位置坐标和时间标签,具体包括移动轨迹数据、社交媒体数据、购物订单数据、手机信令数据等。这些数据记录着人类的日常生活,蕴含着人类活动的潜在规律,且它们正以前所未有的速度和规模增长和累积,亟待被合理、高效、充分地挖掘应用。近年来,面向人类活动的时空大数据逐渐被挖掘、利用、生成各类智慧服务,并渗透到人们生活的各方面。在智慧经济方面,企业利用数据挖掘技术,从客户消费的时空大数据中获取人们消费习惯,并划分成不同的消费群体,从而有针对性地投放产品,实现精准营销;在智慧交通方面,通过分析人流和车辆移动轨迹的时空大数据,可以预测路段的人流密度与交通状况,从而有效改善交通拥堵现象;在智慧医疗方面,通过对海量病历数据进行分析建模,可以了解人群疾病的时空分布规律,从而及时进行疾病的预防和控制。

时空大数据除了具备大数据本身所具有的海量、多维、价值高等特征之外,还具备对象/事件的丰富语义特征和时空维度动态关联特性(李德仁,马军等,2015),具体包括以下4点:

(1) 时空大数据的要素包括对象、过程、事件等,且这些要素在空间、时间、语义等方面具有关联约束关系;

(2) 时空大数据在空间和时间上具有动态演化特性,这些基于时空大数据要素的时空变化是可被度量的;

(3) 时空大数据具有尺度特性,根据比例尺大

小、采样粒度以及数据单元划分的详细程度可以建立时空大数据的多尺度表达与分析模型;

(4) 时空大数据具备多维动态可视化特性,实时获取对象在不同阶段的行为特征,通过参照并映射到三维地理空间中,可以实现动态可视化渲染效果,从而更好地觉察、理解和预测对象的发展。

时空大数据不断被应用于各个领域,促进了新的研究模式生成,然而传统的数据存取、分析和挖掘方法却难以支撑新的研究模式形成。日趋庞大的数据量容易导致算法性能陷入瓶颈,用户对响应的实时性要求越来越高,传统集中式的数据存储管理策略和串行时空分析算法已越来越不能满足时空大数据高效存储和实时处理分析的需求。因此,在分布式计算、并行计算以及云计算技术飞速发展的背景下,针对时空大数据的特殊性,将高性能计算技术应用到时空大数据的处理分析中,实现数据快速高效处理,快速准确提取其中的价值信息,是当前时空大数据的一大研究热点。

3 高性能软硬件发展概况

近年来,计算机技术迅速发展,在硬件方面,计算能力成倍增长,硬件架构发生了巨大变化;在软件方面,云计算技术的兴起,面向大数据的高性能计算模型和处理框架不断涌现。

3.1 硬件架构

(1) 多核处理器

多核处理器(multi-core processor)是在单个芯片(die)中封装了两个或以上的独立中央处理器核心(core),核心间通过高速总线(bus)互联。多核处理器通过提供不同层次的指令级并行和线程级并行从而提高计算性能。2005年,Intel公司和AMD公司率先开发出双核处理器(彭晓明等,2012)。2017年,Intel公司发布了基于14 nm工艺制程的24核处理器(Xeon E7-8894 v4),支持48个线程,频率为2.4~3.4 GHz。

(2) 众核处理器

众核处理器(many-core processor)是专为高度并行处理而设计的专业多核处理器,包含大量简单、独立的处理器内核,并广泛用于嵌入式计算和高性能计算中。众核处理器相对于多核处理器的区别在于它设计的出发点是对更高程度的显式并行进行优化,以密集的计算线程提高吞吐量,使得

它在处理海量结构化数据时更有优势。典型的众核处理器有NVIDIA GPGPU(general purpose GPU)、Intel MIC(many integrated core)等。其中用户群最广的是GPGPU,其具有高带宽、高并行性的特点,因此在处理单指令流多数据流(SIMD)时,面对数据运算量远大于数据调配和传输的场景时,其计算性能相对于传统多核CPU来说有了极大的提升(Nickolls et al, 2010)。2017年NVIDIA发布了Tesla Volta100 GPU架构,能提供30TFLOPS的FP16半精度性能,15TFLOPS的FP32单精度性能,7.5FLOPS的FP64双精度性能,以及120TLOPS独立Tensor操作量。Tesla Volta100中新型流式多处理器架构针对深度学习进行了专门优化,在深度学习性能上显著提升,架构比上一代节能50%,且使用HBM2内存,操作更快、效率更高(NVIDIA, 2017)。

(3) 分布式集群

分布式集群是指将相互独立的计算机节点(node)通过高速互联网络连接起来而形成的计算集群(cluster)。集群面对的高并发请求或计算密集任务可以分发到所有工作节点协同完成。同时由于节点之间通信开销成本和数据传输延迟,因此集群多采用粗粒度的并行任务划分策略。

随着虚拟化技术的发展与成熟,基于云平台的新兴计算模式——云计算应运而生。云平台借助虚拟化技术的伸缩性和灵活性,提高了计算/存储/网络资源利用率;通过信息服务自动化技术,将各类资源封装为服务交付给用户,减少了用户使用成本。云平台提供的服务通常可分为:基础设施即服务(infrastructure as a service, IaaS)、平台即服务(platform as a service, PaaS)、软件即服务(software as a service, SaaS)这3大类。与传统的集群相比,云平台可以将物理资源虚拟化为资源池,灵活提供软硬件资源,实现对用户的按需供给,具有资源池化、按需服务、服务可计量等特点,因此云平台在并行计算中发挥着重要作用。自云计算的概念被IBM提出后,得到了Google、Microsoft、Amazon等各大

IT公司的重视,纷纷推出了各自的云平台,如Google Cloud、微软Azure平台、Amazon Cloud、阿里云等。

表1对以上提出的硬件架构从硬件类型、计算能力、可扩展性等角度进行了比较。

3.2 典型的高性能计算模型及框架

同时,对应的高性能计算模型及框架也呈现出多元化特点,不同的硬件结构逐步发展出与该硬件架构相符合的高性能计算模型,如面向多核处理器的多线程模型,包括OpenMP、POSIX Threads和Intel Threading Building Blocks(TBB)等;面向众核处理器的CUDA、OpenCL等;面向分布式集群的MPI、MapReduce和Spark框架等。

3.2.1 面向多核处理器的并行计算模型

多线程模型是一种基于多线程并发执行(multi-threading)来提升算法处理能力或效率的并行程序开发模型。在一个多线程程序中,一个主进程通常创建多个工作线程,每个工作线程并行执行不同的任务,共享主进程中的全部系统资源。其中具有代表性的多线程模型是OpenMP,它支持C、C++和Fortran语言,可以实现任务并行和数据并行(Dagum et al, 1998)。OpenMP不仅可以利用编译指令自动进行任务分解,允许渐进式并行化,而且对原串行代码不需要进行重大改变,具有良好的可移植性。然而OpenMP只能在共享内存的多核/多处理器平台上高效运行,可扩展性受到存储器架构的限制。

3.2.2 面向众核处理器的并行计算模型

(1) CUDA(compute unified device architecture, 统一计算设备架构)。CUDA是2007年NVIDIA公司推出的运行在本公司各种型号GPU上的一种通用并行计算架构(Garland et al, 2008),最初基于C语言环境,如今可支持C、C++和Fortran等编程语言。CUDA能够为用户提供统一的开发框架和编程模型,辅助用户快速构建高性能应用程序并充分发挥GPU的优势,从而极大提高了通用计算能力。2017年发布了专为Volta GPU而构建的CUDA 9,更快的GPU函数库,基于协作组(cooperative groups)新的

表1 硬件架构对比

Tab.1 Comparison of different hardware architectures

硬件架构	类型	计算能力	可扩展性	优点
多核处理器	Intel Xeon CPU等	中等	中等	多个内核同时处理,可实现多任务处理和计算
众核处理器	GPGPU、MIC等	较强	中等	高度并行性,计算速度快,数据吞吐能力高
分布式集群	云平台等	强	高	节点松散耦合,节点内部可以集成多/众核处理器,扩展性非常强

编程模型,进一步加快了应用程序的编译速度。2018年发布的CUDA 9.2具有更低的内核启动延迟,启动CUDA内核的速度比CUDA 9快2倍。

(2) OpenCL。OpenCL为异构计算提出的统一编程标准,在由CPU、GPU、FPGA或其他处理器等组成的异构平台中,OpenCL可以提供基于任务划分和基于数据划分的并行计算机制(Javier et al, 2012)。OpenCL支持跨平台和硬件体系结构编程的特点,使得它在面对异构计算时具有强大的优势。目前OpenCL 2.2版本通过将OpenCL C++内核语言引入到核心规范中,从而显著提高并行计算的效率。

3.2.3 面向分布式集群的并行计算模型及框架

(1) MPI。MPI是一个基于消息传递的并行计算应用程序接口(Dinan et al, 2016),主要应用于分布式集群上,支持点对点和广播2种通信方式,典型的开源实现有OpenMPI、MPICH等。MPI可移植性强,能同时应用于分布式内存/共享内存的处理平台。在分布式集群上通常采用混合编程模型(Hybrid),同时结合了OpenMP和MPI两者的优势,基于OpenMP实现线程级并行,同时在节点间基于MPI实现任务分配和消息传递,以实现线程和进程2个层次的并行计算(赵永华等, 2005)。目前MPI仍然是当大型计算密集型应用主要使用的并行计算模型。

(2) MapReduce。Google公布的关于Google

File System(Ghemawat et al, 2003)、MapReduce(Dean et al, 2004)和BigTable(Chang et al, 2008)的3篇技术论文,奠定了当前云计算发展的重要基础。其中MapReduce并行开发模型面向大规模数据集的并行处理,能够实现计算任务的自动并行和调度,因其具有简单适用的特点而被广泛应用(李建江等, 2011)。MapReduce模型把计算过程抽象为2个阶段,即Map和Reduce,用户通过实现map(映射)和reduce(规约)2个函数,从而实现分布式计算。结合MapReduce并行框架可实现海量的并行计算任务自动并发执行,同时隐藏底层实现细节,大大降低编程难度(杜江等, 2015)。

(3) Spark。Spark是UC Berkeley大学AMP实验室在2009年提出的一个开源通用并行计算框架,以支持海量数据集的并行处理(Zaharia et al, 2016)。Spark提供了一个基于集群的分布式内存抽象,即弹性分布式数据集(RDD),RDD作为一个可并行操作、有容错机制的数据集合,提供了统一的分布式共享内存。Spark使用了内存计算技术,减少磁盘I/O,允许多次循环访问内存数据集,有助于实现迭代算法,从而进行交互性或探索性数据分析;同时容错性高,保证了分布式应用的正确执行。因此Spark在大数据处理领域中发挥着越来越重要的作用,但也同时存在内存消耗大的问题。

表2 从出现时间、支持硬件、并行粒度、内存访

表2 典型的高性能计算模型及框架的对比

Tab.2 Comparison of high-performance computing models and frameworks

并行计算模型分类	并行计算模型/框架	出现时间	支持硬件	并行粒度	内存访问模型	性能优势	不足	适用场景
面向多核处理器的并行计算模型	多线程(以OpenMP为例)	1997	CPU、MIC	细粒度	共享内存	实现简单,并行效率高,可移植性强	只适用SMP等并行环境,不适合集群,扩展性较差	简单处理算法级的并行与优化
面向众核处理器的并行计算模型	CUDA	2007	GPU	细粒度	共享内存	基于C语言,有强大的并行浮点计算能力	仅能在GPU硬件上使用,通用性较弱	适合大规模数据密集型并行计算
	OpenCL	2008	CPU、GPU、FPGA等	细粒度	共享内存	支持跨平台和硬件体系结构编程,可移植性强	开发环境完善性较差,API设计缺乏一致性	适合异构平台计算
面向分布式集群的并行计算模型	MPI	1992	CPU、MIC	细/粗粒度	分布式内存	适用于集群环境,并行算法扩展性强	编程模型复杂,容错性差	计算密集型应用
	MapReduce	2004	CPU	粗粒度	分布式内存	编程实现简单,容错性好,计算能力较强	模型单一,适合批处理模式,但中间结果要写入磁盘,不适合迭代运算	数据密集型应用、分布式数据处理
	Spark	2010	CPU	粗粒度	分布式内存	丰富的API,减少磁盘I/O,适合迭代算法	内存消耗较大	适合图计算、迭代计算和交互式数据分析

问模型、性能优势、不足以及适用场景等方面对上面列举的高性能计算模型及框架进行了分析比较。

4 时空大数据并行处理与分析进展

4.1 时空大数据存储与管理

时空数据的存储管理方式伴随着计算机技术的发展而不断变化和更新,在每一阶段都受到当前计算机软硬件水平、数据规模特征、实际应用需求等因素的影响,使得数据存储管理模式不断演化,从传统的集中文件存储/空间数据库,发展到以Hadoop HDFS为代表的分布式文件系统管理,再到当前的分布式NoSQL非关系型数据库。

4.1.1 集中式存储

传统的集中式数据存储将存储与应用分离,上层应用通过中间件连接,访问数据时需要通过网络对集中存储的数据进行快速访问。集中式存储中主要有基于文件和基于数据库这2种管理模式。

(1) 基于文件的管理模式。通常是基于单一文件系统(如FAT、NFS等)将数据以“文件”的形式存储在磁盘等可以直接访问的介质中,并提供了一系列存取、索引、更新等操作。这种管理模式存取高效、操作简单,但存储空间扩展性、数据结构兼容性、数据安全性都很差,且存储能力受到存储介质性能的极大影响。

(2) 基于数据库的管理模式。这是当前集中式存储管理的主流模式,利用成熟数据库技术来组织、存储和管理各类数据。传统的关系型数据库在处理结构化数据中有着很大优势,在容量上关系型数据库可采用分区技术对上亿级别的数据进行存储管理以提高访问性能;在并发访问能力上,关系型数据库相对于传统的文件系统来说,它能够从容地应对多用户的高并发访问场景(李绍俊等,2017)。20世纪90年代,基于关系型数据库的空间数据存储管理作为当时的主流应用模式,催生了众多成熟的空间数据库,常见的关系型空间数据库有Oracle Spatial、PostGIS、ArcSDE等。空间数据引擎为用户和空间数据库之间提供了一个开放性接口。基于空间数据引擎和关系型数据库在应用程度中结合的紧密程度,提出了内置、两层结构和三层结构这3种空间数据引擎结构。内置模式是直接在关系型数据库内核新建一个空间扩展模块,模块提供了针对空间数据的一系列操作,典型的有Ora-

cle Spatial、PostGIS等。两层结构模式是可以直接访问空间数据访问客户端和数据库服务端,典型的有SuperMap SDX+。三层结构模式在客户端和数据库服务端中间新增设空间数据应用服务器,并通过中间的服务层来对客户端的数据访问请求进行统一处理和响应,典型的有ESRI ArcSDE。

集中存储模式能够在一定程度上解决海量空间数据存储和管理所面临的问题。然而,在大数据时代下随着数据获取途径多样化、快速发展,数据逐渐呈现出非结构化、高时效性的特点,因此传统集中、单一的数据存储方式越来越不能满足大数据时代下非结构化数据存储和管理的实际应用需求,同时对多用户高并发访问能力提出了更高的要求。

4.1.2 分布式文件系统

2003年,Google研发出了谷歌文件系统GFS(Google file system)(Ghemawat et al, 2003)。GFS是专门针对Google计算机集群为Google的页面搜索数据存储进行优化的一个可扩展的分布式文件系统,集群中节点由众多廉价的服务器组成,主要面向大文件和读操作较多的场景。GFS中数据分块存储,采用了master-slave结构对海量数据按照一定的顺序进行高效存储(孟小峰等,2013)。对应开源的实现有Hadoop HDFS(HDFS, 2012)、Facebook专门针对海量小文件推出的Haystack(Beaver et al, 2010)等。

HDFS主要是面向大文件设计的,基于分布式集群架构实现结构化时空大数据的存储和管理,具有较好的可扩展性和容错性。王凯等(2015)从Hadoop不支持传统空间数据的问题出发,提出了一种针对矢量空间数据的存储格式,并在Hadoop环境下对GIS大数据进行处理,有效提高了GIS大数据的计算效率;尹芳等(2013)对Key/Value键值对数据模型进行了分析,为矢量数据能够在HDFS中进行高效存储而设计出了一种符合GeoJSON地理数据编码的矢量数据Key/Value文本格式,通过HDFS的数据自动分块功能将海量矢量数据自动分割成大量的小数据块,分别存储到不同节点上,以实现海量矢量数据的分布式高效存储。

分布式文件系统支持数据分块存储,具有高扩展性和高可靠性,然而随着用户访问数量激增,分布式文件系统无法提供统一的访问接口以及高效的数据查询检索能力。

4.1.3 分布式非关系型数据库

基于分布式数据库实现半结构化或非结构化时空大数据的存储与管理是当前数据库的重要发展趋势。NoSQL是指非关系型、分布式、不保证遵循关系型数据库ACID原则的数据库的统称(NoSQL, 2009), 可为时空大数据提供低成本、高扩展性、高通量I/O平台, 从而解决多用户高并发场景下海量、快速增长的半结构化和非结构化数据的高效、灵活的存储和管理问题(Ghemawat et al, 2003; 马林, 2009)。目前主流的开源NoSQL数据库主要可分为4类(Hecht et al, 2012):①面向Key-Value存储, 如Redis、Berkeley DB、MemcacheDB等;②面向列存储, 如HBase、Cassandra等;③面向文档存储, 如MongoDB、CouchDB等;④面向图存储, 如Neo4J、FlockDB等。分布式非关系型数据库提供了分布式I/O、索引结构、查询执行和优化等一系列高效管理操作。具体分类如表3所示。

Redis作为高性能面向Key-Value存储的数据库, 数据吞吐量大, 可实现高速读写, 适合数据读写操作多的应用场景。然而Redis基于内存存储的特点对内存的消耗过大, 扩展性较差。在Redis的应用中, 张景云等(2013)为了提高元数据信息和矢量空间数据几何与属性信息的存储效率, 采用了基于Redis实现矢量空间数据按照库、集、层、要素4级结构进行存储的层次组织, 提高了矢量空间数据的管理组织效率; 闫密巧等(2017)基于Redis数据库为海量轨迹数据设计了针对轨迹数据特性的存储方案及数据存储结构, 从而提高了在线平台的轨迹数据实时存储效率。

HBase分布式数据库支持半结构化和非结构化时空数据存储与管理, 满足海量数据存取和时空查询的应用需求。在矢量数据存储方面, 郑坤等(2015)针对矢量空间数据设计了基于HBase的高效

存储模型, 实现了对矢量空间数据的直接存取和展示, 提高了矢量空间数据的存储效率; 在栅格数据存储方面, 张晓兵(2016)基于HBase的高扩展性设计了一个弹性可视化遥感影像存储系统, 解决了海量影像瓦片数据的快速存储问题。

MongoDB提供了多种类型的空间索引, 包括B-tree索引、GeoHash索引等, 从而更好地支持海量数据的分片存储。在矢量数据存储方面, 雷德龙等(2014)基于MongoDB和三层云存储架构的优势, 为海量矢量空间数据的高效存储管理与处理分析设计出了VectorDB; 在栅格数据存储方面, 田帅(2013)、张飞龙(2016)都将MongoDB和分布式文件系统结合起来设计了海量遥感数据存储管理系统, 其中采用基于MongoDB的高性能存储架构对遥感影像元数据进行高效存取, 针对遥感影像文件数据则采用了分布式文件系统进行存储, 此系统的混合存储策略实现了海量遥感影像数据的高效存取并提高了存储资源的利用率。

Neo4J是一个面向图操作的高性能、高可靠的开源图形数据库。马义松等(2016)基于Neo4J构建了一个电网的全景数据库, 基于该数据库对电网中具有分散、隔离特性的电力大数据进行了有序整合。廖理(2015)针对关系型数据库存储效率低、扩展性差等特点提出了一种基于Neo4J的时空数据存储模型, 该模型能够有效地将空间、时间和属性信息整合起来进行建模和存储, 提高时空数据存储效率。

分布式非关系型数据库侧重于提高半结构化和非结构化数据的存储效率, 可提供优于关系型数据库的低成本、高可扩展性、高并发能力和高通量I/O的海量数据存储平台。然而, 其在互联网领域的应用模式与传统的GIS领域还存在差异, 使得NoSQL数据库在数据操作方式、时空索引支持、查

表3 目前主流的NoSQL数据库分类
Tab.3 Classification of classic NoSQL database

分类	数据库	支持平台	存储性能	灵活性	复杂性	优势	不足
面向Key-Value	Redis, MemcacheDB等	Linux	高	高	无	内存数据库、可实现高速读写	内存消耗较大, 扩展性较差
面向列	HBase, Cassandra等	Linux, Windows	高	中等	低	数据压缩率高, 支持快速的OLAP	没有原生的二级索引
面向文档	MongoDB, CouchDB等	Linux, Mac, Windows	高	高	低	面向Document, 支持空间数据管理	不支持事务操作, 占用空间过大
面向图	Neo4J, FlockDB等	Linux, Mac, Windows	可变	高	高	高精度的图算法, 图查询迅速	没有分片存储机制, 图数据结构写入性能较差

询访问模式等方面仍存在较大的局限性。

4.2 时空大数据并行分析

时空大数据因其自身特点及存储模式的变化,使得传统的串行分析算法存在很大的局限性,不能充分利用和发挥当前新型硬件构架(单机多核、集群、集群/众核混合等)资源的优势,难以满足实际应用的规模与高效需求,因此时空大数据的并行分析已成为当前研究的热点。

传统GIS数据,主要以结构化的栅格影像和半结构化的矢量要素形式存储。对于这类空间数据的分析,经过长期的发展目前已形成较为成熟的串行算法库,并且对应的GIS软件平台非常成熟,如ArcGIS、SuperMap、GRASS等;然而这些串行算法及平台随着GIS数据量和计算规模的逐渐增大,难以满足实际的应用需求,因此并行空间分析算法受到越来越多的关注。作为GIS的2大基础数据结构,栅格和矢量具有不同的特征与优势,对于地理要素的表达形式不同,分析方法不同,因此栅格空间分析与矢量空间分析的并行化策略也各有侧重。

(1) 对于栅格空间分析来说,每一个像元上的计算形式相同且相对独立,单个像元计算任务复杂度低,因此多采取数据并行的策略,即在数据分块的基础上利用并行计算技术进行处理分析。过去,栅格空间分析往往采取CPU并行的方法,但随着GPU通用计算能力的发展以及在数据并行上的良好适应性,越来越多的栅格分析方法开始采取基于GPU的并行化策略。在图像分类、融合和滤波等方面,杨靖宇等(2010)从GPU的并行性和流式编程模型出发,为图像的高效处理分析设计了一种流水线并行处理模式,实现了影像光谱角匹配算法的并行化;卢俊等(2009)充分发挥GPU可编程渲染器和并行处理数据的优势,提出了基于GPU的遥感影像IHS融合算法,其将IHS算法映射到GPU的流式计算中,结果显示该算法的处理速度明显优于传统基于CPU的算法;杨洪余等(2017)利用CUDA编程模型的特性,提出了面向CPU/GPU异构环境的图像协同并行处理模型,结果显示该模型在灰度图像处理中处理速度得到了较大提升。在DEM地形分析方面,Do等(2011)为了提取DEM的排水网络以获取全局的流量累积,从而提出了一种并行生成树方法对集水盆地进行分层统计,结果显示该方法无需完整的DEM,分析效率高,扩展性强;Qin等(2012)提出了一种在GPU上兼容CUDA计算流量累积的

并行方法,对DEM数据预处理在GPU上进行了并行化实现,针对递归MFD算法的并行化提出了基于图论的并行化策略,结果显示该策略在流量累积计算中处理速度得到了很大的提升。然而,这些并行化方法都是针对单个空间分析算法的具体实现,属于细粒度的线程级并行。与之相对的粗粒度并行编程模型如MPI/MapReduce则是通过栅格数据分割实现分布式计算,有效利用集群资源。Xu等(2014)提出了一种基于MPI和MapReduce并行计算模式的栅格加权Voronoi图生成算法,该算法显著提高了利用大规模栅格数据生成Voronoi图的效率,并在城市公共绿地规划和最优路径规划中得到验证;程果等(2012)基于MPI并行计算模型提出了栅格地形分析中坡度坡向计算的并行化方法,有效降低了数据通信成本,充分利用了并行计算资源。更进一步,为了提高并行程序的可移植性,在栅格分析的通用并行框架研究方面,Qin等(2014)提出了一种面向栅格空间分析的并行框架PaRGO,该框架能够兼容OpenMP、CUDA以及MPI。

(2) 对于矢量空间分析来说,由于矢量数据是不定长结构,且实体之间存在着复杂的空间关系,尤其是空间实体之间的拓扑关系导致传统基于任务的数据划分方法会造成数据量存在很大差异,从而导致并行任务负载不均衡、数据通信成本高等种种问题。因此,目前的研究大体上可以分成3个方向:①从矢量数据划分策略入手,将其转化为数据并行,涉及索引构建、负载平衡、通信调度等。贾婷等(2010)从空间数据的拓扑特征出发,在矢量空间数据划分策略中采用了Hilbert空间填充曲线,并改进了K均值聚类算法对矢量空间数据进行均衡划分;刘润涛等(2009)从矢量空间数据的查询和索引结构出发,对K均值算法中基于R树和四叉树的空间索引结构进行优化,提高了矢量空间数据的查询和索引效率。②从矢量空间分析并行化算法本身入手,算法主要包括空间叠加分析、空间关系运算和网络分析等。在叠加分析方面,一般采用管道叠加、数据并行叠加和块式叠加这3种并行策略(Wilson, 1994; Langendoen, 1995),Qatawneh等(2009)在管道网络配置中并行实施Liang-Barsky限幅算法,实现了大规模的多边形裁剪。在空间关系的运算方面,朱效民等(2013)基于线段求交和点面叠加这2个基础空间分析算法,采用OpenMP进行了并行化实现,该方法利用数据排序及OpenMP的动态调度

特点优化了并行算法的内存管理,从而提高了并行算法的加速比;在网络分析方面,主要通过网络复制及网络分割实现并行化分析(隽志才等,2006;卢照等,2010);在大规模网络路径分析中,网络分割策略求解最短路径的性能更好,Lanthier等(2003)给出了三角不规则网络上欧氏和加权度量的最短路径算法的并行实现;在空间插值方面,王鸿琰等(2017)以薄板样条函数插值为例,提出了一种CPU/GPU协同并行的插值算法以加速海量LiDAR点云生成DEM,并行算法取得了19.6倍的加速比;Wu等(2011)面向亿级的激光雷达点数提出了基于多核平台的Delaunay三角网并行构建算法ParaStream,提高了数据吞吐量和降低了内存占用量。③从大规模集群的并行编程框架入手,主要是指基于MapReduce/Spark应用于海量非结构化或半结构化数据上的处理分析特点和优势,使得其在矢量空间分析并行化过程中被广泛应用。王凯等(2015)在Hadoop环境下对GIS大数据设计了一种更高效的并行处理模型,基于该模型进一步增强了Hadoop的空间计算能力;2013年,ESRI ArcGIS 10.2对Hadoop上的空间运算类库进行扩展,将GIS数据与Hadoop分析相结合,推出“GIS Tools for Hadoop”为大数据处理提供了基于Hadoop的并行分析环境,该框架充分利用了MapReduce进行并行数据计算,进一步提高了GIS空间计算效率和能力。未来,ESRI的分布式并行计算会将重心从MapReduce转移到Spark上,进一步挖掘Spark的优势。Cheng等(2016)基于超图模型,研究了海量空间数据分布式处理的调度策略的研究,并对I/O进行了优化,提升了海量数据的处理效率。

总的来说,GIS数据的空间分析并行化策略已日趋成熟,研究学者们对栅格和矢量这两类数据的空间分析算法并行化策略进行了大量的研究,卓有成效。然而,目前大部分的策略是针对单一算法的数据并行,未形成通用、成熟的空间分析算法并行框架,扩展性不强。因此,开发出自适应多种硬件平台、同时支持矢量/栅格空间分析功能的GIS并行框架是目前迫切需要解决的问题。

4.3 时空大数据挖掘

时空数据分析,侧重于利用已有数据集进行数据转换、处理以及简单的信息提取等操作,为用户决策提供依据。而相较于时空数据分析,时空数据挖掘综合了人工智能、机器学习、领域知识等交叉

方法,旨在从大规模数据集中发现高层次的模式和规律,揭示时空数据中具有丰富价值的知识,为对象的时空行为模式和内在规律探索提供支撑。目前,时空大数据挖掘作为一个新兴的研究方法,已在众多领域得到广泛应用,如交通监管、犯罪预测、环境监测、社交网络等。

传统的数据挖掘算法多基于小型数据集,研究更关注模型精度和规律识别能力,往往忽略模型执行效率和数据吞吐能力。然而,面对TB级别甚至是PB级别的非结构化海量数据时,基于单机处理的传统数据挖掘算法,数据处理效率低,其计算能力难以满足时空大数据挖掘的需求,因此如何在时空大数据时代下进行高效的数据挖掘是目前面临的一大难题。云计算技术的出现和迅速发展,为研究大数据时代下的高性能数据挖掘方法提供了更高的可能性,利用云平台动态资源分配以及并行计算的能力,将传统的数据挖掘串行算法转化成并行算法,并有效移植到云平台中,从而实现数据挖掘的并行化和高效化。

(1) 面向地理环境的时空大数据挖掘

随着遥感成像方式的多样化以及遥感数据获取能力的增强,遥感领域进入了“大数据”时代。然而现有的遥感影像分析和处理技术与“遥感大数据”之间的不匹配,难以采用已有技术对遥感大数据进行充分挖掘,因此基于遥感大数据的特点进一步促进数据挖掘技术的发展和优化是目前遥感领域的前沿问题之一。

遥感大数据挖掘应用广泛,不仅可以发现不同尺度下的地理空间演变规律,还被应用于反映人类社会活动的社会经济估算、环境污染监测、城市化监测等方面。Li等(2014)基于叙利亚2008–2014年间的夜间灯光数据进行分析,结果显示各区域内流离失所者的数量与夜间灯光损失呈线性相关,相关系数达0.5以上,说明了夜光遥感数据分析能对叙利亚战争危机进行有效监测。Zhang等(2018)基于Gaofen-1 160 m空间分辨率的AOD数据、大气模式模拟数据,提出了嵌套线性混合效应模型,预测了超高分辨率的PM_{2.5}日均浓度。Zhao等(2018)利用东南亚国家1992–2013年间的DMSP/OLS夜间灯光数据,基于像素级夜间灯光亮度与东南亚城市的相应空间梯度之间的二次关系,划分成低、中、高、极高4种类型的夜间照明区域,对城市化发展进行动态分析,结果显示,不同夜间照明区域之间的过

渡模式描绘出城市化发展的不同模式。除此上述典型案例外,李德仁等(2013)和李德仁等(2014)提出了基于“OpenRS Cloud”的遥感大数据挖掘平台,充分利用分布式计算的优势对多源、海量遥感数据进行存储、分析等,实现了遥感大数据的高效存取,进而利用机器学习、人工神经网络、云模型等方法逐步探索遥感大数据间蕴藏的内在联系及知识,进一步实现从遥感数据到知识的转变。

(2) 面向人类社会活动的时空大数据挖掘

面向人类社会活动的时空大数据挖掘中最具代表性有移动轨迹大数据挖掘和社交媒体大数据挖掘。

移动轨迹数据是在时空维度下对运动物体的移动过程进行数据收集所获得的数据信息,具有规模大、种类多、变化快、价值高但密度低、精度低等特点,包括浮动车轨迹数据、人类出行轨迹数据等。这些数据刻画了个体和群体在时空环境下的时空动态性,蕴含着人类、车辆等对象的移动模式和行为特征,对城市空间规划、交通状况预测、个性化服务等应用具有重要的价值。现阶段,云计算等新兴数据处理方法为移动轨迹大数据的高效分析、深度挖掘提供了新的发展方向。在时空关联规则挖掘方面,Chester等(2011)采用了MapReduce并行计算模型对FP-Growth算法进行了并行化实现,同时还优化了数据挖掘并行策略,从而提高了时空挖掘的处理速度和能力。谢欢(2015)基于Spark框架实现了传统协同过滤算法和时空关联规则FP-Growth算法的并行化优化,并行化后的算法进一步提高了数据处理效率和数据挖掘能力。夏大文(2016)基于Hadoop中的MapReduce并行处理模型搭建了面向移动轨迹大数据的分布式计算平台,提出了基于MapReduce的并行频繁模式增长算法MR-PFPh和交通流量预测分布式建模通用框架MF-TFF,能有效挖掘出交通时空特征、实时交通流量等数据背后蕴藏的有价值信息。

社交媒体与地理位置服务的结合与应用,产生了包含丰富的空间、时间和语义等信息的海量社交媒体数据,如微博签到数据、点评数据等,具有数据量大、产生速度快、现势性高但数据质量参差不齐等特点。由于社交媒体数据与人类生活息息相关,社交媒体大数据挖掘受到越来越多的关注,探索用户的时空行为模式成为研究热点之一。罗俊(2016)提出了基于LFM的局部敏感哈希的MapReduce并

行化k-means聚类的协同过滤算法,并应用于用户个性化推荐系统中,从而减少了传统k-means算法处理海量高维数据时的计算量和迭代次数,提高了用户个性化推荐系统的可拓展性和实时性。廉捷(2013)针对网络爬虫数据大、效率低等问题,采用了Fetch-List索引模式对网络爬虫进行并行化优化,使得优化后的网络爬虫系统能同时多线程获取感兴趣的数据,减少了数据获取时间;同时基于爬虫获取的数据采用SVM模型权重优化算法进行并行化处理和分析,提高了利用网络数据预测信息的准确度,用于预测信息传播方向。

时空大数据挖掘在众多领域中得到广泛应用,基于云计算框架的分布式数据挖掘算法研究已成为热点。然而,数据挖掘算法并行化仍然受算法自身的限制较多,如挖掘任务数据使用模式、任务之间的相关性、任务执行流程等。而且,如今人工智能算法发展如火如荼,给大数据挖掘算法提供了新的模型和方法。人工智能算法多为“黑箱”模型,将底层数据挖掘过程隐藏起来,使大数据挖掘更方便,但也同时使得挖掘算法扩展性变差、并行难度加大。

5 当前研究发展趋势

综上,正是由于时空大数据相较过去在存储管理、处理分析、智能挖掘等方面日益增长的性能需求,时空大数据并行处理、分析、挖掘在面临上述问题的同时,也涌现了新的发展趋势。这些并行化发展趋势可总结为以下3点:

(1) 异构计算逐渐成为计算主流

大数据时代计算需求的多元化促进了CPU/GPU异构计算的快速发展(卢风顺等,2011)。由于CPU和GPU在硬件设备和计算方式上具有显著差异,CPU主要是面向复杂多任务逻辑,GPU则具有更高的通用计算能力,更适合用于海量结构化数据的并行化处理和分析。因此,异构计算平台具有很大的发展潜力,综合CPU和GPU两者的优势制定一个高效合理的协同方式,保证CPU和GPU之间的计算负载均衡,降低两者由于数据处理方式不同而带来的额外成本,促进双方资源的合理配置,使异构计算平台在时空数据处理分析和挖掘中发挥出最佳性能。

(2) 时空计算与时空大数据存储融合一体化

时空大数据处理面临的挑战本质上是计算平

台的处理能力与大数据处理的问题规模之间的矛盾。传统以计算为中心的数据处理模式将存储与计算相互分离,在实际处理操作中常常面临着内存容量有限、I/O压力过大等难题,使得数据处理体系难以达到最佳性能。目前在时空大数据环境下,面对数据量成指数级别增长的趋势,研究学者已经认识到数据的存储、传输成为制约算法/模型性能提升的另一个关键因素,因此将计算与存储相互融合是计算技术发展的重要方向,具有良好的发展前景(殷进勇等,2015)。

(3) 面向时空智能分析框架的成熟化

随着云计算框架的不断发展更新,分布式数据挖掘算法研究日趋成熟;同时如今人工智能发展迅速,人工智能与时空数据挖掘也逐步融合。然而,一方面,目前时空大数据的处理分析和挖掘是一个离散、迭代的过程,通常需要针对不同的应用需求场景提出不同的分析和挖掘算法并进行优化。因此,迫切需要一个集成众多时空分析和数据挖掘功能的通用时空智能分析框架,兼容多类模型和多源时空数据,从而实现时空大数据分析应用的统一化和高效化。另一方面,目前人工智能算法多为“黑箱模型”,算法的不透明性导致难以对其进行扩展和优化,并行难度进一步加大。因此,时空数据处理分析算法的并行与底层人工智能算法的并行需要进一步统一和融合。

参考文献(References)

- 程果,景宁,陈萃,等.2012.栅格数据处理中邻域型算法的并行优化方法[J].国防科技大学学报,34(4): 114-119.
[Cheng G, Jing N, Chen L, et al. 2012. Parallel optimization methods for raster data processing algorithms of neighborhood-scope[J]. Journal of National University of Defense Technology, 34(4): 114-119.]
- 杜江,张铮,张杰鑫,等.2015. MapReduce 并行编程模型研究综述[J].计算机科学,42(S1): 537-541. [Du J, Zhang Z, Zhang J X. 2015. Survey of MapReduce parallel programming model[J]. Computer Science, 42(S1): 537-541.]
- 贾婷,魏祖宽,唐曙光,等.2010.一种面向并行空间查询的数据划分方法[J].计算机科学,37(8): 198-200. [Jia T, Wei Z K, Tang S G, et al. 2010. New spatial data partition approach for spatial data query[J]. Computer Science, 37 (8): 198-200.]
- 隽志才,倪安宁,贾洪飞,等.2006.两种策略下的最短路径并行算法研究与实现[J].系统工程理论方法应用,15(2): 123-127. [Jun Z C, Ni A N, Jia H F, et al. 2006. Study and implement of shortest path parallel algorithms with two strategies[J]. System Engineering- Theory Methodology Applications, 15(2): 123-127.]
- 雷德龙,郭殿升,陈崇成.2014.基于MongoDB的矢量空间数据云存储与处理系统[J].地球信息科学学报,16(4): 507-516. [Lei D L, Guo D S, Chen C C. 2014. Vector spatial data cloud storage and processing based on MongoDB [J]. Journal of Geo-information Science, 16(4): 507-516.]
- 李德仁,马军,邵振峰.2015.论时空大数据及其应用[J].卫星应用, (9): 7-11. [Li D R, Ma J, Shao Z F. 2015. The application of spatial temporal big data[J]. Satellite Application, (9): 7-11.]
- 李德仁,王树良,李德毅.2013.空间数据挖掘理论与应用[M].北京:科学出版社. [Li D R, Wang S L, Li D Y. 2013. Spatial data mining theories and applications[M]. Beijing, China: Science Press.]
- 李德仁,张良培,夏桂松.2014.遥感大数据自动分析与数据挖掘[J].测绘学报,43(12): 1211-1216. [Li D R, Zhang L P, Xia G S. 2014. Automatic analysis and mining of remote sensing big data[J]. Acta Geodaetica et Cartographica Sinica, 43(12): 1211-1216.]
- 李建江,崔健,王聃,等.2011. MapReduce 并行编程模型研究综述[J].电子学报,39(11): 2635-2642. [Li J J, Cui J, Wang D, et al. 2011. Survey of MapReduce parallel programming model[J]. Acta Electronica Sinica, 39(11): 2635-2642.]
- 李绍俊,杨海军,黄耀欢,等.2017.基于NoSQL数据库的空间大数据分布式存储策略与实践[J].武汉大学学报:信息科学版,42(2): 163-169. [Li S J, Yang H J, Huang Y H, et al. 2017. Geo-spatial big data storage based on NoSQL database[J]. Geomatics and Information Science of Wuhan University, 42(2): 163-169.]
- 廉捷.2013.基于用户特征的社交网络数据挖掘研究[D].北京:北京交通大学. [Lian J. 2013. Research on user features based data mining in social networks[D]. Beijing, China: Beijing Jiaotong University.]
- 廖理.2015.基于Neo4J图数据库的时空数据存储[J].信息安全与技术,6(8): 43-45. [Liao L. 2015. Application research of Neo4J in spatio-temporal data storage[J]. Information Security and Technology, 6(8): 43-45.]
- 刘润涛,安晓华,高晓爽.2009.一种基于R-树的空间索引结构[J].计算机工程,35(23): 32-34. [Liu R T, An X H, Gao X S. 2009. Spatial index structure based on R-tree[J]. Computer Engineering, 35(23): 32-34.]
- 卢风顺,宋君强,银福康,等.2011.CPU/GPU协同并行计算研究综述[J].计算机科学,38(3): 5-10. [Lu F S, Song J Q, Yin F K, et al. 2011. Survey of CPU/GPU synergistic parallel computing[J]. Computer Science, 38(3): 5-10.]
- 卢俊,张保明,黄薇,等.2009.基于GPU的遥感像数据融合

- IHS 变换算法[J]. 计算机工程, 35(7): 261-263. [Lu J, Zhang B M, Huang W, et al. 2009. IHS transform algorithm of remote sensing image data fusion based on GPU [J]. Computer Engineering, 35(7): 261-263.]
- 卢照, 师军. 2010. 并行最短路径搜索算法的设计与实现[J]. 计算机工程与应用, 46(3): 69-71. [Lu Z, Shi J. 2010. Design and implementation of parallel shortest path search algorithm [J]. Computer Engineering and Applications, 46(3): 69-71.]
- 罗俊. 2016. 数据挖掘算法的并行化研究及其应用[D]. 青岛: 青岛大学, [Luo J. 2016. Research on parallelization of data mining algorithm and application[D]. Qingdao, China: Qingdao University.]
- 马林. 2009. 数据重现: 文件系统原理精解与数据恢复最佳实践[M]. 北京: 清华大学出版社. [Ma L. 2009. Shuju chongxian: Wenjian xitong yuanli jingjie yu shuju huifu zuijia shijian[M]. Beijing, China: Tsinghua University Press.]
- 马义松, 武志刚. 2016. 基于Neo4J的电力大数据建模及分析[J]. 电工电能新技术, 35(2): 24-29. [Ma Y S, Wu Z G. 2016. Modeling and analysis of big data for power grid based on Neo4J[J]. Advanced Technology of Electrical Engineering and Energy, 35(2): 24-29.]
- 孟小峰, 慈祥. 2013. 大数据管理: 概念、技术与挑战[J]. 计算机研究与发展, 50(1): 146-169. [Meng X F, Ci X. 2013. Big data management: Concepts, techniques and challenges [J]. Journal of Computer Research and Development, 50 (1): 146-169.]
- 彭晓明, 郭浩然, 庞建民. 2012. 多核处理器: 技术、趋势和挑战[J]. 计算机科学, 39(Z3): 320-326. [Peng X M, Guo H R, Pang J M. 2012. Mutil-core processor: Technology, tendency and challenge[J]. Computer Science, 39(Z3): 320-326.]
- 田帅. 2013. 一种基于MongoDB和HDFS的大规模遥感数据存储系统的设计与实现[D]. 杭州: 浙江大学. [Tian S. 2013. A design and implementation of large-scale remote sensing data storage system based on MongoDB and HDFS [D]. Hangzhou, China: Zhejiang University.]
- 王鸿琰, 关雪峰, 吴华意. 2017. 一种面向CPU/GPU异构环境的协同并行空间插值算法[J]. 武汉大学学报: 信息科学版, 42(12): 1688-1695. [Wang H Y, Guan X F, Wu H Y. 2017. A collaborative parallel spatial interpolation algorithm on oriented towards the heterogeneous CPU/GPU system[J]. Geomatics and Information Science of Wuhan University, 42(12): 1688-1695.]
- 王凯, 曹建成, 王乃生, 等. 2015. Hadoop 支持下的地理信息大数据处理技术初探[J]. 测绘通报, (10): 114-117. [Wang K, Cao J C, Wang N S, et al. 2015. Research on GIS big data computing technologies based on Hadoop[J]. Bulletin of Surveying and Mapping, (10): 114-117.]
- 夏大文. 2016. 基于MapReduce的移动轨迹大数据挖掘方法与应用研究[D]. 重庆: 西南大学. [Xia D W. 2016. MaReduce: Based methodologies of mobile trajectory big data mining and its applications[D]. Chongqing, China: Southwest University.]
- 谢欢. 2015. 大数据挖掘中的并行算法研究及应用[D]. 成都: 电子科技大学. [Xie H. 2015. Research and application on the parallel algorithm in big data mining[D]. Chengdu, China: University of Electronic Science and Technology of China.]
- 闫密巧, 王古宏, 王志宇. 2017. 基于Redis的海量轨迹数据存储模型研究[J]. 微型电脑应用, 33(4): 9-11. [Yan M Q, Wang Z H, Wang Z Y. 2017. Large-scale trajectory data storage model based on Redis[J]. Microcomputer Applications, 33(4): 9-11.]
- 杨洪余, 李成明, 王小平, 等. 2017. CPU/GPU 异构环境下图像协同并行处理模型[J]. 集成技术, 6(5): 8-18. [Yang H Y, Li C M, Wang X P, et al. 2017. Image cooperative parallel processing model in CPU/GPU heterogeneous environment[J]. Journal of Integration Technology, 6(5): 8-18.]
- 杨靖宇, 张永生, 董广军. 2010. 基于GPU的遥感影像SAM 分类算法并行化研究[J]. 测绘科学, 35(3): 9-11. [Yang J Y, Zhang Y S, Dong G J. 2010. Investigation of parallel method of RS image SAM algorithmic based on GPU[J]. Science of Surveying and Mapping, 35(3): 9-11.]
- 殷进勇, 杨阳, 徐振朋, 等. 2015. 计算存储融合: 从高性能计算到大数据[J]. 指挥控制与仿真, 37(3): 1-7. [Yin J Y, Yang Y, Xu Z P, et al. 2015. The fusion of computing and storage: From HPC to big data[J]. Command Control & Simulation, 37(3): 1-7.]
- 尹芳, 冯敏, 诸云强, 等. 2013. 基于开源Hadoop的矢量空间数据分布式处理研究[J]. 计算机工程与应用, 49(16): 25-29. [Yin F, Feng M, Zhu Y Q, et al. 2013. Research on vector spatial data distributed computing using Hadoop projects[J]. Computer Engineering and Applications, 49(16): 25-29.]
- 张飞龙. 2016. 基于MongoDB遥感数据存储管理策略的研究[D]. 开封: 河南大学. [Zhang F L. 2016. Research on the storage management strategy of remote sensing data base on MongoDB[D]. Kaifeng, China: Henan University.]
- 张景云. 2013. 基于Redis的矢量数据组织研究[D]. 南京: 南京师范大学. [Zhang J Y. 2013. Vector data organization research based on Redis[D]. Nanjing, China: Nanjing Normal University.]
- 张晓兵. 2016. 基于HBase的弹性可视化遥感影像存储系统[D]. 杭州: 浙江大学. [Zhang X B. 2016. An HBase based remote sensing elastic visualization storage system[D]. Hangzhou, China: Zhejiang University.]
- 赵永华, 迟学斌. 2005. 基于SMP集群的MPI + OpenMP混

- 合编程模型及有效实现[J]. 微电子学与计算机, 22(10): 7-11. [Zhao Y H, Chi X B. 2005. MPI + OpenMP hybrid paradigms and efficient implementation base on SMP clusters[J]. Microelectronics & Computer, 22(10): 7-11.]
- 郑坤, 付艳丽. 2015. 基于 HBase 和 GeoTools 的矢量空间数据存储模型研究[J]. 计算机应用与软件, 32(3): 23-26. [Zheng K, Fu Y L. 2015. Research on vector spatial data storage model based on HBase and GeoTools[J]. Computer Applications and Software, 32(3): 23-26.]
- 朱效民, 潘景山, 孙占全, 等. 2013. 基于 OpenMP 的两个地学基础空间分析算法的并行实现及优化[J]. 计算机科学, 40(2): 8-11. [Zhu X M, Pan J S, Sun Z Q. 2013. Parallel implementation and optimization of two basic geospatial-analysis algorithms based on OpenMP[J]. Computer Science, 40(2): 8-11.]
- Beaver D, Kumar S, Li H C, et al. 2010. Finding a needle in haystack: Facebook's photo storage[C]//Usenix conference on operating systems design and implementation. USENIX Association: 47-60.
- Chang F, Dean J, Ghemawat S, et al. 2008. Bigtable: A distributed storage system for structured data[J]. ACM Transactions on Computer System, 26(2): 1-26.
- Cheng B, Guan X F, Wu H Y, et al. 2016. Hypergraph+: An improved hypergraph-based task-scheduling algorithm for massive spatial data processing on master-slave platforms [J]. ISPRS International Journal of Geo-Information, 5(8): 141-157.
- Chester S, Crowe J. Explorations of parallel fp_growth[EB/OL]. 2011-08-13[2018-8-31]. <http://webhome.csc.uvic.ca/schester/>.
- Dagum L, Menon R. 1998. OpenMP: An industry standard API for shared-memory programming[J]. IEEE Computational Science & Engineering, 5(1): 46-55.
- Dean J, Ghemawat S. 2004. MapReduce: Simplified data processing on large clusters[J]. Sixth Symposium on Operating System Design and Implementation, 51(1): 137-150.
- Dinan J, Balaji P, Buntinas D, et al. 2016. An implementation and evaluation of the MPI 3.0 one-sided communication interface[J]. Concurrency and Computation: Practice and Experience, 28(17): 4385-4404.
- Do H-T, Limet S, Melin E. 2011. Parallel computing flow accumulation in large digital elevation models[J]. Procedia Computer Science, 4(4): 2277-2286.
- Garland M, Grand S L, Nickolls J, et al. 2008. Parallel computing experiences with CUDA[J]. IEEE Micro, 28(4): 13-27.
- Ghemawat S, Gobioff H, Leung S T. 2003. The Google file system[J]. Proceedings of SOSP 2003, Operating Systems Review, 37(5): 29-43.
- HDFS. 2012. HDFS architecture guide[EB/OL]. 2012-10-02 [2018-08-31]. http://hadoop.apache.org/docs/hdfs/r0.22.0/hdfs_design.html.
- Hecht R, Jablonski S. 2012. NoSQL evaluation: A use case oriented survey[C]//International conference on cloud and service computing (ICSC). IEEE, 336-341.
- Hong S, Oguntebi T, Olukotun K. 2011. Efficient parallel graph exploration on multi-core CPU and GPU[C]//International conference on parallel architectures and compilation techniques. IEEE Computer Society: 78-88.
- Javier D, Camelia M-C, Alfonso N. 2012. A survey of parallel programming models and tools in the multi and many-core era[J]. IEEE Transactions on Parallel and Distributed System, 23(8): 1369-1386.
- Langendoen H F. 1995. Parallelizing the polygon overlay problem using Orca[D]. Amsterdam, Holland: Vrije Universiteit Amsterdam.
- Lanthier M, Nussbaum D, Sack J R. 2003. Parallel implementation of geometric shortest path algorithms[J]. Parallel Computing, 29(10): 1445-1479.
- Li X, Li D R. 2014. Can night-time light images play a role in evaluating the Syrian Crisis[J]. International Journal of Remote Sensing, 35(18): 6648-6661.
- Manyika J, Chui M, Brown B, et al. 2011. Big data: The next frontier for innovation, competition, and productivity[R]. Chicago, IL: The McKinsey Global Institute: 1-156.
- Nickolls J, Dally W J. 2010. The GPU computing era[J]. IEEE Micro, 30(2): 56-69.
- NoSQL. 2009. NoSQL definition: Next generation databases mostly addressing some of the points: Being non-relational, distributed, open-source and horizontally scalable[EB/OL]. 2009-06-11[2018-08-31]. <http://nosql-database.org/>.
- NVIDIA. 2017. NVIDIA Tesla V100 GPU architecture: The world's most advanced data center GPU[J/OL]. 2017-08-30 [2018-08-31]. [https://images.nvidia.com/content/volta-architecture-whitepaper.pdf](https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf).
- Qatawneh M, Sleit A, Almobaideen W. 2009. Parallel implementation of polygons clipping using transputer[J]. American Journal of Applied Sciences, 6(2): 214-218.
- Qin C Z, Zhan L J. 2012. Parallelizing flow-accumulation calculations on graphics processing units: From iterative DEM preprocessing algorithm to recursive multiple-flow-direction algorithm[J]. Computers & Geosciences, 43(6): 7-16.
- Qin C Z, Zhan L J, Zhu A X, et al. 2014. A strategy for raster-based geocomputation under different parallel computing platforms[J]. International Journal of Geographical Information Science, 28(11): 2127-2144.
- Waldrop M. 2008. Big data: Wikiomics[J]. Nature, 455: 22-25.
- Wilson G V. 1994. Assessing the usability of parallel program-

- ming systems: The Cowichan problems[M]//Decker K M, Rehmann R M. Programming environments for massively parallel distributed systems. Basal, Switzerland: Birkhäuser: 183-193.
- Wu H Y, Guan X F, Gong J Y. 2011. ParaStream: A parallel streaming delaunay triangulation algorithm for lidar points on multicore architectures[J]. Computers & Geosciences, 37(9): 1355-1363.
- Xu G H. 1999. Pay much attention to the digital earth by the social[J]. Science News Weekly, (1): 7-8.
- Xu M, Cao H, Wang C Y. 2014. Raster-based parallel multiplicatively weighted voronoi diagrams algorithm with MapReduce [M]//Cao B Y, Ma S Q, Cao H H. Ecosystem assessment and fuzzy systems management. New York: Springer International Publishing: 177-188.
- Zaharia M, Xin R S, Wendell P, et al. 2016. Apache spark: A unified engine for big data processing[J]. Communications of the ACM, 59(11): 56-65.
- Zhang T H, Zhu Z M, Gong W, et al. 2018. Estimation of ultra-high resolution PM_{2.5} concentrations in urban areas using 160 m Gaofen-1 AOD retrievals[J]. Remote Sensing of Environment, 216(10): 91-104.
- Zhao M, Cheng W M, Zhou C H, et al. 2018. Assessing spatio-temporal characteristics of urbanization dynamics in Southeast Asia using time series of DMSP/OLS nighttime light data[J]. Remote Sensing, 10(1): 47- 66.

Research progress and trends of parallel processing, analysis, and mining of big spatiotemporal data

GUAN Xuefeng, ZENG Yumei*

(State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing,
Wuhan University, Wuhan 430079, China)

Abstract: With the rapid development of the Internet, Internet of things, and cloud computing technology, data with geographical location and time tag are accumulated in an explosive way, and this indicates that we are in the era of big spatiotemporal data. In addition to the typical "4V" characteristics, big spatiotemporal data also contain rich semantic information and dynamic spatiotemporal patterns. Although massive spatiotemporal data have promoted the evolution of various cross-disciplinary studies, traditional methods of data processing and analysis would no longer meet the requirements of efficient storage and real-time analysis of such data. Therefore, it is of great importance to integrate big spatiotemporal data with high-performance computing/cloud computing. To address this problem, this article begins with the concept and origin of big spatiotemporal data, and introduces its unique characteristics. Then, the performance requirements generated by current big data applications are analyzed, and the status quo of the underlying hardware and software is summarized. Furthermore, the article comprehensively reviews parallel processing, analysis, and mining methods for big spatiotemporal data. Finally, we conclude with the challenges and opportunities of storage, management, and parallel processing analysis of big spatiotemporal data.

Key words: big spatiotemporal data; high- performance computing; parallel spatial analysis; data mining; progress and trends