GIS 中海量栅格数据的处理技术研究*

朱 雷[1]，潘 懋[1]，李丽勤[2]，吴焕萍[1]

(1. 北京大学 造山带与地壳演化教育部重点实验室，北京 100871；2. 北京市信息化促进中心，北京 100871)

摘 要：综合介绍了近些年来采用的针对 GIS 海量数据的处理策略及技术，指出了这些技术的优缺点；并在 GeoScene 软件的开发过程中得到了很好的应用。

关键词：海量数据；LOD；Out-of-core；内存池

中图法分类号：TP274 文献标识码：A 文章编号：1001-3695(2006)01-0066-03

Research on Massive Grid Data Treatment Techniques in GIS

ZHU Lei[1], PAM Mao[1], LI Li-qin[2], WU Huan-ping[1]

(1. *Key Laboratory of Orogenic Belts & Crustal Evolution, Ministry, Peking University, Beijing* 100871, *China;* 2. *Beijing Information Accelerate Center, Beijing* 100044, *China*)

Abstract: Spatial data are increasing in an unprecedented speed with the development of survey techniques. How to deal with the data is an important problem for the low capability of computer. Several ways have been proposed with different characteristics in recent years. Gives a general review on these methods.

Key words: Massive Grid Data; LOD; Out-of-core; Memory Pool

随着现代测绘技术的发展，GIS 中的空间数据正以前所未有的速度增长；如：我国 1.25 万的 DEM 数据(8.27GB)、七个波段的 Landsat TM 影像(135G)、全球 30m 分辨率的 DEM 数据(>12TB)。数据量的增长，在计算机分析处理上产生了很多问题，比如数据不可能一次完全被读入计算机的内存中进行处理。单纯依赖于硬件技术，并不能满足持续增长的数据的处理要求。因此需要在软件上找到处理海量数据的策略，并最终通过软硬件的结合完成对海量数据的处理。

针对这个问题近年来许多专家从不同侧面进行过研究。Lindstrom[1,2]在地形简化中使用了外存模型(Out-of-core)技术；钟正[4]采用了基于数据分块、动态调用的策略；汪国平等人[5]在研究使用高速网络进行三维海量地形数据的实时交互浏览中，采用了分块、多分辨率模板建立模型等方法。这些对海量数据的研究多数集中在各自系统的实现方式上，并没有对各种海量数据的策略以及技术进行综合性的研究。本文将从海量数据软件处理策略和软件处理技术两个角度对解决这个问题的主要方法进行讨论。

1 海量数据软件处理策略

在进行海量数据处理中，目前主要通过三种策略来进行海量数据的处理：分块策略、LOD(层次细节)模型和外存模型。

1.1 分块策略

分块的策略是将原始数据分割成若干大小一定的规则块状数据，使系统可以一块一块地处理数据。在数据文件块的分割与组织上，通常根据数据文件的访问方式：简单顺序访问方式和随机访问方式来组织数据。顺序访问方式在数据分块过程中保持原有数据的顺序；而对于随机访问方式，可以以任意的次序选择、修改文件，数据文件的划分较随意，但需要有索引文件。由于大部分文件访问动作需要用到这个索引，因此索引通常情况保存在主内存中。

对应分块大小的选择，Ueng 等人[6]提出了两种方法：

(1) 将各子块划分为互不相交的单元。由于数据集是不规则的，得到的子块通常也是不规则的。该方法的主要优点是没有数据冗余，缺点是很难确定块间的空间关系，如块与块的相邻关系等。

(2) 通过有序的结构来对数据分块。常用的结构包括规则格网、k-d 树、八叉树、BSP 树[7]等。该方法的主要优点是通过组织数据子集的有序结构，很容易获得块与块的空间关系。

此外在考虑分块的时候，一味地减小文件的尺寸会产生过多的文件块，过多的文件一方面不利于管理，另一方面对索引的设计提出了更高的要求。在索引上耗时过多同样不能提高系统的速度。

在 GIS(地理信息系统)中，栅格数据是一种主要的 DEM 地形数据。这些数据比较规则，往往也是顺序进行的，因此在数据分块上，可以考虑采用根据现有 DEM 分幅尺寸进行分块，同时对分块保存的数据文件建立一个简单的索引文件，在系统进行数据处理前，将这个索引文件读入内存中。

1.2 LOD(层次细节)模型

LOD(Level Of Detail)模型这种策略最早是由 Clark[11]在 1972 年提出的，它采用对同一个数据集中的数据使用具有不同细节的描述方法得到一组模型，供分析、绘制时选择使用。这样在进行宏观的场景观察时，仅仅需要将比较粗略的数据模型读入内存。

收稿日期：2005-01-25；修返日期：2005-03-18

基金项目：国家“863”计划资助项目(2001AA135180)

LOD 模型方法分为静态 LOD 模型和动态 LOD 模型。静态模型通过预先处理数据来生成不同的 LOD 模型层次; 动态模型在程序运行时动态地生成 LOD 模型层次。最简单的静态 LOD 模型的构建方法是使用金字塔模型, 如图 1 所示。

金字塔模型通过分层来保存不同细节的数据, 图中最下层的数据是原始数据集; 从底层开始, 在每层的基础上生成上一层金字塔数据, 新的金字塔中的数据量仅仅是上下一层数据的 1/4, 以此类推, 直到最上面一层的数据量达到了内存所允许的范围或者更小的数据尺寸。在进行宏观数据分析时可以考虑采用最低精度级别的数据, 随着对数据精度的要求逐渐提高, 可以提供更加精细的数据。就是 LOD 模型的核心思想。

为了提高系统访问海量数据的灵活性, 基于视点的 LOD 模型技术得到发展, 这种技术在同一个视窗内, 离视点近的数据比较精细, 而离视点远的数据比较粗略。这种 LOD 模型的建立是同分块策略结合进行的; 如图 1 所示, 根据视点不同可在不同层次上使用不同精度的数据。这样在访问海量数据的时候就可以随意跨越 LOD 的不同层次而产生灵活的分析结果。

1.3　外存模型

外存模型基本思想是按需读取, 只将当前必须处理的数据调入内存, 由于当前驻入内存中的只是相对很少的一部分数据, 这样就完全可以实现对海量数据进行处理。外存模型技术的计算模型分为两种: 批处理方法和在线处理方法。批处理方法是数据不经过预处理, 只是将数据文件分为小的足够内存处理的块, 并按一定方式将当前计算需要的数据块驻入内存; 这种方法类似于分块算法, 适用于顺序文件的访问。在线处理方法对外存中的原始数据进行预处理, 按照一定的结构组织以便于高效查询, 在数据处理时通过一系列的查询将部分数据驻入内存[8]。外存模型在进行数据组织及建立索引的过程中的一个最大的特点是, 它将分析数据访问的频度。根据不同的访问频度, 将数据设置不同的优先度。有些访问频度较高的数据可以始终保存在系统内存中, 而不被交换到磁盘上去。

在外存模型技术中树结构作为中间结构被广泛用于数据的组织。此外 Pascucci 采用 Lebesgue 空间填充曲线来组织和索引数据[9]; Varadhan 用图来组织外存模型技术所需的数据[10]; Vitter 给出了基于外存算法的数据层次设计方案[3], 在现在的计算机体系结构中的各级存储器(CPU 快速缓存、系统内存、硬盘等) 的有效使用, 对提高性能会起到很大的作用。

总之, 在海量数据的处理策略中, 通过分块策略系统使用一定大小的内存顺序或者随机访问整个数据文件中的数据; 通过建立 LOD 及多分辨率模型具体分析应用要求, 同样可以使用一定大小的内存空间, 来完成不同的具体任务。而 Out-of-core 技术则是一种综合的解决方案, 外存模型技术通过将计算机的各种存储设备和数据处理的需要度结合起来, 改变了传统内存算法(In-core) 把所有数据调入内存处理的模式, 整个数据存放在次级外部存储设备上, 经过组织可以部分地被装入内存, 缓解了内存资源的相对不足和 I/O 瓶颈、减少 I/O 时间、提高内存中数据的复用度, 可以提高外存模型算法的效率。外存模型技术已被用于海量数据的生成、组织和可视化[1,2,4]。目前这种方法已经渐渐成为处理海量数据的主流方法。

2　海量数据软件处理技术

在海量数据的处理上除了采取上述策略外, 在程序设计上同样需采用一定的技术, 使用这些方法可以提高程序编写效率, 如内存映射方法、多线程技术、内存池技术。

2.1　内存映射文件技术

内存映射文件技术是操作系统内存管理模块的一部分, 它提供了一个统一的内存管理方法, 使应用程序能够通过内存指针像访问动态内存一样对磁盘上的文件进行访问。在程序运行中的任一时刻, 整个程序都可以由执行代码页面和应用程序资源页面组成, 这些页面可以保存在内存中, 也可以保存在磁盘上; 根据需要由操作系统把这些页面交换进内存或换出内存。通过内存映射技术, 对文件内容的访问就如同在该地址区域内直接对指针取值一样简单。

这样, 向文件中写入数据就可以直接对指针进行赋值, 如 * pMem = 1; 同样, 从文件的某个特定位置读取数据也一样非常的简单, 如 nTokenLen = * pMem; 事实上, 当内存映射文件提供了对文件某个特定位置的直接读写时, 真正对磁盘文件的读写是由底层实现的。

使用内存映射文件 I/O 的一个好处是速度快。系统对所有数据的传输都是使用一定尺寸的数据页面(如 4KB) 来实现的。因为虚拟内存管理器以统一的方式处理所有磁盘 I/O, 一次以一个页面为单位读写内存, 这种优化使它能够以足够快的速度处理内存操作。限制所有的磁盘的读写操作以 4K 大小的页面为单位进行, 意味着一些小的 I/O 操作将被缓冲入一次大的操作之中, 也就是说一次 I/O 操作以后的其他操作所需的数据已经被前一次的页面操作读入到内存, 从而无须再进行一次磁盘 I/O, 这样将大幅度地减少硬盘读写头的移动, 从而提高了系统的性能。

在地理信息系统软件开发中使用这种技术将会获益很大。但是这种技术同操作系统密切相关, 因此如果进行跨平台软件的设计, 在移植的时候将会比较困难。

2.2　多线程处理技术

在处理海量数据时, 计算机的资源被大量占用。为了提高程序的可交互性, 在程序设计的时候往往采用多线程技术。多线程技术是一种成熟的技术, 它通过使用多个程序计数器并发执行程序中不同位置的指令来加速程序的执行过程。每个线程严格地按照顺序执行, 拥有自己的程序计数器和栈, 线程之间可以像进程那样分时共享 CPU。线程也可以像进程一样产生自己的子线程, 也可以被阻塞并等待系统调用返回, 一个线程被阻塞后, 其他线程可以照常运行。

多线程技术可以应用在数据生成、外存模型技术、可视化、LOD 模型层次切换等方面。特别是处理已经分块的数据上, 很有优势。

图 2 显示的是在进行海量数据处理过程中的利用多线程技术的一个例子: 整个数据根据一定的方法被分割成小块, 图中共显示了 16 块数据, 每一次处理其中的一块数据。在内存中共保存九块数据(A ～I) 用深色表示。当前处理的数据块 E 用浅黄表示, 随着数据处理的进行, 当数据处理的边界离开 E 块向 A, B, D 块中移动的时候, 需要判断目前数据块的左上角

所处的位置。这里我们将 1 号块再虚拟分成四个小区域: a, b, c, d。当落在 d 区域内,内存中还应该保证有 A ~I,九块数据在内存中。随着数据处理的边界继续向左上方移动,当数据块的边界越过数据块 A 的中心时,系统开始进行内存数据的导入、导出操作,一个新的工作线程被创建,该线程完成 1, 2, 3, 5, 6 数据块的读入,同时将处理后的 C, F, I, G, H 数据块导出系统。这样保证系统中有九块数据可以进行分析。采用多线程的技术,可以使系统同磁盘的数据交换过程尽量的分散,在最大数据交换情况下需要交换五块内存数据块,如果数据处理仅仅是在一个方向上越过边界,需要交换的内存块数仅有三块。这种多线程的分步加载策略对系统运行的流畅非常有利,在数据读取工作线程进行数据读取的同时,分析线程可以继续进行数据的分析;在进行漫游操作的过程中,可以有效地避免由于数据块的导入、导出造成的界面停止响应的问题。

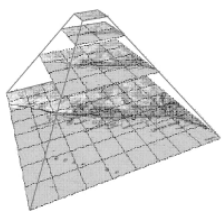

图 1 LOD 数据组织中的金字塔模型

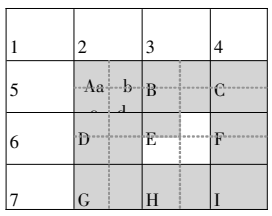

图 2 内存中的数据块示意图

除了进行数据块的交换外,多线程技术还可以在海量地形显示处理中用来进行“遮挡判断”,在网络地形浏览中进行数据流传输等。该技术目前已经被广泛地应用在现有的 GIS 系统中;在进行海量数据处理中多线程技术可以充分利用计算机的资源,有效地提高系统的效率,减少停顿等现象的出现。

2.3 内存池与缓冲技术

对原始数据采用分块策略可以减少系统对内存的需求量,但在分析海量数据的过程中如果不进行很好的设计的话,程序很快就会消耗掉系统中的内存,系统响应将逐渐减慢并最终让人难以接受。为了保证数据的处理仍然能够正常进行而不是简单的提示用户内存不足,在控制内存的使用中可以采用限制内存的方法,这种方法也称为内存池技术。根据系统硬件配置的不同情况,将海量数据计算时能够使用的内存限制在一定的数量之内。使用这些内存一次只处理一部分数据,在处理结束后把数据循环写回辅助存储设备,直到再次使用它们为止。这样系统对内存需求会限制在一个可以预期的尺寸内,不会因为无限制的使用而导致系统的交互性下降。

此外在处理海量数据的过程中,由于数据导入导出,系统不断地动态分配、释放内存,也会使得系统堆变得支离破碎,程序效率逐渐下降。在一些情形下,即使没有出现任何内存泄漏,应用程序不久也会出现崩溃。如图 3 所示,这个图解代表了系统中的一个堆: 0 表示是自由块, 1 表示是分配块(即正在使用中的存储区域) 。

虽然图 3a 中总共有 10 个自由块,但试着从这个有太多碎片的堆中分配出五个单位的连续自由块(即五个 0) ,肯定会失败。图 3b 中仅仅有七个自由块,但是没有碎片,有足够的内存使分配操作得以正确执行。

这个问题除了可以通过内存池技术解决,还可以通过缓冲技术来解决。缓冲技术是同内存池相联系的,其原理是为对象创建两个内存池,但是每次只使用其中的一个。在分配对象时,系统简单地在活动的内存池中指定下一个可用空间。如果这个内存池填满了,系统把它们从当前的内存池里复制到另一个非活动的内存池里,然后将原来这个原本非活动的内存池切换成活动的。由于这种方法运行的时候,应用程序会停下来,导致程序的波动。往往在程序设计的时候使用三个内存池,其中的一个填满以后,程序使用第二个内存池进行新的内存分配;同时将第一个内存池中的数据整理到第三个中,在复制完成后,再将三号内存池激活,如此重复使用。

使用内存池技术、缓冲技术可以很好地解决程序运行过程中的逐渐减慢,以及运行的颠簸问题,提高了程序的稳定性。

3 结论与展望

就处理 GIS 中海量栅格数据的问题中主要使用的处理策略进行了一个较为综合性的探讨。在实验室开发的海量数据 GIS 软件 GeoScene 中使用了这些技术,并取得了很好的效果。此外这些技术不单单在地学 GIS 这一特定的领域中使用,同样还可以在具有相似情况的其他数据的计算机处理过程应用中。

参考文献:

[1] indstrom P, Pascucci V. Visualization of Large Terrains Made Easy [C]. Proceedings of IEEE Visualization, 2001. 363-370.

[2] Lindstrom P, Pascucci V. Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-core Visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2002, 8(3): 239-254.

[3] Vitter J S. External Memory Algorithms and Data Structures: Dealing with Massive Data[J]. ACM Computing Surveys, 2001, 33(2): 209-271.

[4] 钟正, 朱庆. 一种基于海量数据库的 DEM 动态可视化方法[J]. 海洋测绘, 2003, 23(2): 9-13.

[5] 汪国平, 吴学礼, 陈斌, 等. 高速网上三维海量地形数据的实时交互浏览的实现[J]. 测绘学报, 2002, 31(1): 34-38.

[6] Ueng S, Sikorski K, Ma Kwan-Liu. Out-of-core Streamline Visualization on Large Unstructured Meshes[J]. IEEE Transactions on Visualization and Computer Graphics, 1997, 3(4): 370-380.

[7] Wald I, Slusallek P, Benthin C. Interactive Distributed Ray Tracing of Highly Complex Models[C]. Rendering Techniques, 2001. 277-288.

[8] Isenburg M. Out-of-core Compression for Gigantic Polygon Meshes [EB/OL]. http://www.cs.unc.edu/~isenburg/oocc/, 2003.

[9] Pascucci V, Frank R. Global Static Indexing for Real-time Exploration of Very Large Regular Grids[C]. Proc. of SC2001, High Performance Networking and Computing, 2001.

[10] Gokul Varadhan, Dinesh Manocha. Out-of-core Rendering of Massive Geometric Datasets[C]. Proceedings of IEEE Visualization, 2002. 69-76.

作者简介:

朱雷(1973-) , 男, 博士生, 研究方向为 GIS; 潘懋(1954-) , 男, 教授, 博士生导师, 研究方向为 GIS、信息地质; 李丽勤(1973-) , 女, 副研究员, 博士, 研究方向为信息化应用; 吴焕萍(1976-) , 男, 博士生, 研究方向为 GIS。