文章编号:1671-8860(2011)02-0140-04　　**文献标志码**:A

海量三维地质空间数据的自适应预调度方法

孙　卡[1]　吴冲龙[1]　刘　刚[1]　何珍文[1]

(1　中国地质大学(武汉)计算机学院国土资源信息系统研究所,武汉市鲁磨路 388 号,430074)

摘　要:针对大规模三维地质空间数据实时应用中的高效调度难题,采用空间聚类和空间插值理论,将缓存中的空间对象视为样品数据,将这些对象的命中率作为估值权值,将空间索引中的空间对象信息当作待估值数据,兼顾系统的内存容量和 CPU 的计算能力,设计实现了海量三维地质空间数据的自适应预调度算法。实验结果证明了该方法的正确性和有效性。

关键词:空间聚类;预调度;空间索引;自适应

中图法分类号:P208

在地质应用领域,由于地质空间数据(如地质体、地质现象和地质过程)具有不可直见性、非参数化、非结构化和非均质特征,这导致地质数据存在分布不均、疏密不同和随机性强的特点,在数据表达、数据存储、专题应用等方面与地表数据有很大的不同,因此已有的基于地表规则数据结构的数据调度技术不足以保证地质空间数据的高效调度。采用调度和预调度有机结合的策略来实现海量地质空间数据的调度是一条有效的途径,而其难点主要在于如何从海量的空间数据库中选择可被预先调度的对象。目前,2D 地学软件常使用基于扩展范围的预测模型,3D 地学软件常使用基于视点的预测模型。前者将与当前可见区域相邻的 8 个未可见区域的数据预调度出来;后者根据当前视点的位置、运动方向、运动速度、角速度等参数建立关于视点的预测函数,计算几个可能的视点位置,并根据视点进行数据的可见性判断,进而实现空间对象的选择,其典型应用如德国 Saarland 大学开发的用于大规模场景射线实时追踪的 K-D 树内外存一体化结构[1,2]。此外,Oracle 开发了基于对象关系图的预调度技术[3],该技术采用面向对象的思想,为存在继承、派生、联合、聚合关系的对象建立对象关系图,并沿此图实现预调度对象的追踪和加载。然而,地质空间数据的固有特征(结构信息不完全、参数信息不完全、关系信息不完全、演化信息不完全)导致已有的预调度模型不能完全满足其预调度的需求。本文结合地质空间数据的特点,兼顾其空间范围、空间相关性及调度频率,设计实现了一种新的预调度方法。

1　预调度算法设计

空间相关性在地学软件的应用(如地质分析和地质模拟)中占主导地位,与已调度的地质对象在空间上邻近的对象很有可能被调入内存。本文使用缓存来管理已调度到内存的地质对象,并对所有地质对象建立全局空间索引;使用空间聚类[4,5]来挖掘已调度地质对象之间的空间关系;使用调度频率作为衡量调度靶区的依据;使用自适应算法实现最优预调度对象的选择和加载。

1.1　样品树及热点调度区域

本文将已经被调度到缓存中的地质对象称作活动对象,尚在存储设备中的地质对象称作非活动对象。地质对象的包围盒可为矩形包围盒、有向包围盒[6]、包围球或其他任意类型的包围盒;将活动对象的包围盒或包围盒向外扩展一定比例构成的空间区域称作其影响区域,构建影响区域的目的是为了获得活动对象与其他对象之间粗略的空间关系,本文采用地质对象的矩形包围盒作为活动对象的影响区域;活动对象在缓存中的命中次数反映了该对象的重要程度,本文将其作为活动对象的影响因子;将满足筛选条件而被加载到

收稿日期:2010-12-15。

项目来源:国家 863 计划资助项目(2008AA121602)。

缓存中的对象称作预调度对象。在图 1 中,包围盒 1 至包围盒 11 为地质对象 1 至地质对象 11 的影响区域,假定对象 2、对象 3、对象 6 已被调度到缓存中,则它们是活动对象。与活动对象影响区域相交的对象,如对象 1、对象 4、对象 8、对象 10、对象 11 为预调度候选对象。本文主要研究如何根据活动对象高效合理地从非活动对象中确定预调度对象。

缓存中活动对象影响因子的值是不同的,若按影响因子为关键字对所有活动对象排序,以其影响区域为查询条件、以相交和包含为判别规则来检索全局空间索引,进而确定预调度对象,这样势必会增加对空间索引的检索次数,更无法体现活动对象的空间相关性。本文借鉴聚类分析的思想,提出了样品树的概念。样品树是按活动对象的空间距离而建立的具有不同粒度级别的聚类结构。该树具有深度的概念,处于同一层次上的聚类对象具有相同的聚类粒度。图 2 中,Root 为样品树的根结点,活动对象 2 和活动对象 6 构成聚类对象 C,对象 3 的聚类对象为其自身。假定对象 2、对象 3、对象 6 的影响因子和影响区域分别为 R_2、R_3、R_6 和 V_2、V_3、V_6,对于聚类对象 C,其影响因子可按式(1)计算:

$$R_C = (R_2 * V_2 + R_6 * V_6)/V_C \qquad (1)$$

以递归的方式自下而上计算,可求得样品树中每个聚类对象的影响因子,其流程如图 2 中的实线方向所示。

如果 $R_C \geqslant R_3$,则说明聚类对象 C 影响区域中的空间对象被用户频繁地调度,该区域为热点调度区域。本文采用相交和包含判别原则,优先从热点调度区域中选择预调度对象,因此,图 1 中预调度候选对象 1 和 10 的调度优先级要高于其他预调度候选对象,预调度对象的过滤流程为非活动对象、预调度候选对象、预调度对象。

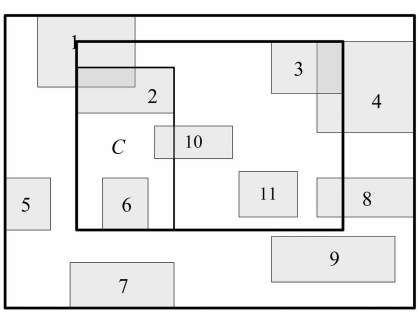

图 1　活动对象及非活动对象空间位置图

Fig. 1　Spatial Position About Active and Inactive Objects

对于式(1),若活动对象的包围盒较大,则说明它在空间中的延展较广,它与其他空间对象发生空间关系的几率也较大;若活动对象的影响因子较大,则说明其被频繁地进行空间调度,与它发生空间关系的空间对象被调度的几率也比较大。另外,空间索引仅记录非活动对象包围盒的大小,其详细的空间分布信息是未知的,系统只能粗略地判断它与活动对象之间的空间关系,因此,将聚类对象的影响因子作为均值处理是合理的。

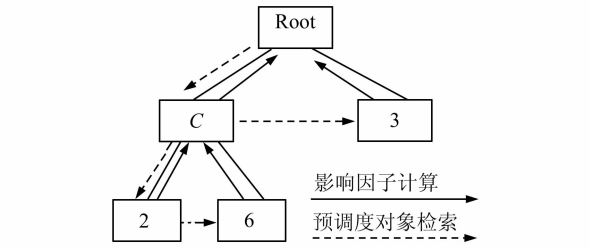

图 2　活动对象的样品树

Fig. 2　Sample Tree of Active Objects

1.2　算法的自适应特点

自适应由硬件自适应和软件自适应组成。前者是指系统根据硬件(内存、CPU)的使用情况自动地选择预调度的对象,并适时地启动和关闭预调度程序;后者是指预调度程序采用面向对象技术和统一访问接口技术实现,本算法中的空间索引算法、缓存管理算法、聚类分析算法、影响因子的确定算法等在发生改变时,算法的总体流程和功能也发生改变。

由于样品树及聚类对象的影响因子已被确定,系统可从样品树的根结点开始,以其影响区域为检索条件对全局索引进行检索,其流程如图 3 所示。

1) 若当前结点检索到的预调度候选对象的 ID 为(1,2,3,…,n),检测到的供预调度对象使用的内存空间为 K_{pre},空间索引项中存储的单个空间对象占用的内存为 K_i,若 $\sum_{1}^{n} K_i \leqslant K_{\text{pre}}$,则执行步骤 4)、5),否则执行步骤 2)、3);

2) 对其孩子结点以影响因子为关键字进行排序,以热点调度区域为首顺次建立兄弟关系;

3) 跳转到孩子结点所在层的热点调度区域中,执行步骤 1);

4) 将检索到的对象 ID 放入预调度对象 ID 链表;

5) 跳转到其兄弟结点中,若兄弟结点存在,则执行步骤 1),否则结束。

该流程优先检索热点调度区域,如图 2 中的虚线方向所示。由于以上操作在内存中进行,不

涉及到地质数据的物理加载,其速度是比较快的,地质数据的加载在检索流程结束后统一进行。本文采用触发器及多线程相关机制实现CPU的自适应:触发器实时监控CPU的计算能力,当CPU的剩余计算能力满足设定条件时,启动预调度进程,以最大程度地利用CPU;当CPU的使用率超过一定值时,则停止预调度进程,以保证其他操作的正常进行;多线程则确保系统在执行其他操作时,可同时执行预调度进程。

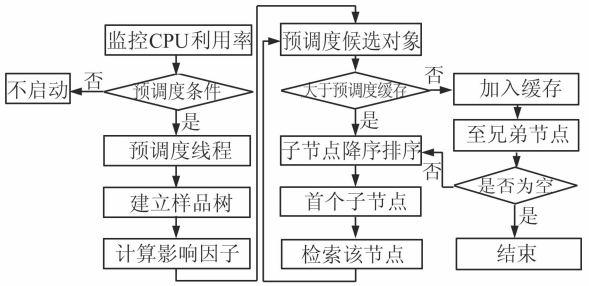

图3 预调度的流程

Fig. 3 Flow Chart of Pre-Load

2 实 验

本文采用Guttman提出的R-Tree来建立全局空间索引,该索引是目前最有前途的真三维空间索引算法之一[7];样品树体现了不同聚类粒度级别上空间对象间的相关性,实现了影响因子的计算,并确定了热点调度区域,为选择预调度对象提供了依据;基于频率的替换算法LRU实现了缓存管理功能,它将缓存中最近最少使用的活动对象替换出缓存,并提供活动对象命中率的访问接口。

本文采用4个金属矿山的三维可视化模型数据作为测试数据,其数据量及相关参数如表1所示。其中,露天开采模型采用TIN数据结构表达;地层边界及矿体边界数据采用B-Rep数据结构表达;非均质的地层及矿体内部数据采用基于八叉树的规则块体结构表达;地下巷道模型采用结构实体几何(CSG)和B-Rep混合数据结构表达。

实验流程如下:触发器实时监控系统的运行情况,若系统处于调度间歇期(编辑状态或分析状态),且CPU的利用率满足条件,则启动预调度进程,并将调度出的对象放在预调度缓存中;若系统处于调度进程,则停止预调度进程,并遍历预调度缓存,将满足调度查询条件的预调度对象转移到调度缓存中,然后执行剩余空间对象的调度。由于预调度进程和调度进程是交替执行的,因此网络及服务器的使用率得到提高,并最大程度地避免了网络阻塞。本实验中通过设定具体的间歇时间来模拟系统的编辑和分析操作,以降低手工操作的干扰。

实验参数如下,服务器内存4 G,CPU 3.8 GHz;客户端内存4 G,CPU 3.6 GHz;数据库软件为Oracle 11g;网络为千兆网卡;调度缓存为1 G;预调度缓存为500 M。实验数据如表1所示,间歇时间均为2 000 ms。图4分别展示了实验3至实验6调度过程中数据量不断增加的效果。

表1 预调度的实验数据及效率对比

Tab. 1 Experimental Data and Efficiency Contrast About Pre-load

编号	数据总量/G	活动对象数	调度数据量/M	无预调度/ms	预调度/ms	效率提升/%
1	4.2	234	1.05	326	229	29.75
2	4.2	702	3.74	1 105	794	28.14
3	6.7	1 734	7.38	2 351	1 743	25.86
4	6.7	2 017	9.47	2 746	2 205	19.70
5	6.7	2 548	11.32	3 324	2 597	21.87
6	6.7	3 394	13.82	4 213	3 451	18.09
7	9.4	4 562	19.27	5 718	4 365	23.66
8	9.4	5 492	23.37	6 754	5 440	19.46
9	14.6	7 563	30.24	8 518	6 987	17.97
10	14.6	8 946	35.98	11 015	8 995	18.34

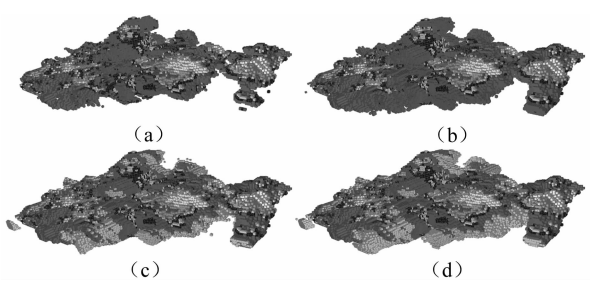

图4 针对地质数据的调度结果

Fig. 4 Load Result About Geological Data

3 结 语

本文设计出了满足地质空间数据调度需要并兼顾计算机的实时处理能力的预调度算法。实验证明,该算法可提升20%左右的调度效率。由于自适应技术的采用,本算法中的空间索引算法、缓存管理算法、聚类分析算法、影响区域、影响因子的确定算法可以使用其他算法进行替换,因此,该算法在理论和应用上具有一定的普适性。本算法尤其适合储量估算、油气模拟、地质分析等需对海量地质数据进行快速调度的应用。在处理河流、地下管线等空间上延展较广的线状地物时,活动对象的影响区域可进行分裂处理,以更好地反映

其空间形态。另外,基于格网的影响因子确定方法以及基于服务器端的预调度策略有待于进一步研究。

参 考 文 献

[1] Stefan P, Johannes G, Hans P, et al. KD-Tree Traversal for High Performance GPU Ray Tracing [J]. Computer Graphics Forum, 2007, 26(3):415-424

[2] Gerd M, Roman B, HeikoF, et al. Accelerated and Extended Building of Implicit Kd-Trees for Volume Ray Tracing[C]. The 11th International Fall Workshop-Vision, Modeling, and Visualization (VMV), Aachen, Germany, 2006

[3] Wenny R, David T, Eric P. Object-oriented Oracle [M]. USA:IMR Press, 2005

[4] 李德仁,王树良,李德毅,等. 论空间数据挖掘和知识发现的理论与方法[J]. 武汉大学学报·信息科学版,2002,27(3):221-233

[5] 田扬戈,边馥苓. 基于概念聚类和面向属性归纳的区划分析[J]. 武汉大学学报·信息科学版,2005,30(1):86-88

[6] Gottschalk S, Lin M C, Manocha D. OBB-Tree: A Hierarchical Structure for Rapid Interference Detection[C]. ACM SIGGRAPH'96, New Orleans, USA, 1996

[7] 朱庆,龚俊. 一种改进的真三维索引方法[J]. 武汉大学学报·信息科学版,2006,31(4):340-343

第一作者简介:孙卡,博士生,现主要从事地质信息技术、空间数据库的研发。

E-mail:sunka1982@163.com

Self-Adaptive Pre-Load Method for Massive 3D Geological Data

SUN Ka[1]　*WU Chonglong*[1]　*LIU Gang*[1]　*HE Zhenwen*[1]

(1 Faculty of Earth Resources, School of Computer, China University of Geosciences, 388 Lumo Road, Wuhan 430074, China)

Abstract: Aiming at the difficult problems of massive geological data load, the theoretical basis is spatial clustering and spatial interpolation. The spatial objects in cache are regarded as sample data, the hit-ratio of these objects are thought as interpolation weight, and the object information in spatial index are look upon as estimated data, and the RAM and the computing capabilities of CPU are also took into account. The self-adaptive pre-load method on massive 3D spatial data in geological space are designed, and the experimental result shows that our proposed method is correct and efficiency.

Key words: spatial clustering; pre-load; spatial index; self-adaptive

About the first author: SUN Ka, Ph.D candidate, majors in geological information technology and spatial database.
E-mail: sunka1982@163.com

下期主要内容预告