



计算机科学

COMPUTER SCIENCE

基于Inverted-B+树的海量三维地质块体模型高效索引方法

陈根深, 刘刚, 董洋, 范文遥, 易强, 姜子鑫

引用本文

陈根深, 刘刚, 董洋, 范文遥, 易强, 姜子鑫. 基于Inverted-B+树的海量三维地质块体模型高效索引方法[J]. 计算机科学, 2025, 52(8): 146-153.

CHEN Genshen, LIU Gang, DONG Yang, FAN Wenyao, YI Qiang, JIANG Zixin. Efficient Indexing Method for Massive 3D Geological Block Models Based on Inverted-B+ Tree [J]. Computer Science, 2025, 52(8): 146-153.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[面向海量空间数据的分布式距离连接算法](#)

Distributed Distance Join Algorithm for Massive Spatial Data

计算机科学, 2022, 49(1): 95-100. <https://doi.org/10.11896/jsjcx.210100060>

[基于MAP准则的红外图像小波域比例萎缩降噪和增强算法](#)

Denosing Algorithm of Proportional Shrinkage with Enhancement Based on the MAP Rule in Wavelet Domain for Infrared Image

计算机科学, 2010, 37(4): 274.

[非下采样轮廓波域红外与可见光图像配准算法](#)

Image Registration Algorithm for Infrared and Visible Light Based on Non-subsampled Contourlet Transform

计算机科学, 2016, 43(11): 313-316. <https://doi.org/10.11896/j.issn.1002-137X.2016.11.061>

[DPsIR⁺: 一种基于动态空间槽的分布式并行空间索引树](#)

计算机科学, 2006, 33(2): 121-126.

[DCPC: 基于能量保护的传感器网络分布式拓扑控制协议](#)

计算机科学, 2007, 34(4): 28-31.

基于 Inverted-B+ 树的海量三维地质块体模型高效索引方法

陈根深^{1,2,3,4,5} 刘刚^{1,2,3,4,5} 董洋^{3,4} 范文遥^{1,2,4,5} 易强^{1,2} 姜子鑫^{3,4}

1 中国地质大学(武汉)计算机学院 武汉 430074

2 智能地学信息处理湖北省重点实验室 武汉 430074

3 武汉地大坤迪科技有限公司 武汉 430074

4 贵州省战略矿产智慧勘查重点实验室 贵阳 550081

5 自然资源部城市国土资源监测与仿真重点实验室 广东 深圳 518000

(gschen@cug.edu.cn)

摘要 三维地质块体模型中大量的零值或空值使得基于 B+ 树的属性索引结构频繁分裂和调整,导致索引维护成本高;同时, B+ 树的单向链表结构加剧了大规模块体模型中数据顺序遍历和范围查询效率低下的问题。为此,提出了一种基于 Inverted-B+ 树 (IBT) 的索引方法。该方法通过构建 IBT 索引结构,在将重复键插入叶子节点时,为每个重复键创建倒排节点,从而有效减少了数据处理中的结构调整。通过在内部节点存储中间索引值来加速查询过程,并在叶子节点和倒排节点之间建立双向链表,实现了从任意叶子节点按顺序访问整个数据集从而进行高效的范围查询。利用三维地质结构模型经过体元剖分、插值和降维处理所得到的 6 个块体模型进行测试,结果表明:与传统 B+ 树相比, IBT 方法在索引构建时间、空间占用和查询性能方面均有显著提升,特别是在处理大规模数据集中,其索引构建效率提升了 71%,空间占用减少了 83%,查询效率得到了显著提升。

关键词: Inverted-B+ 树;规则块体;三维地质模型;空间数据管理;空间索引

中图分类号 TP391

Efficient Indexing Method for Massive 3D Geological Block Models Based on Inverted-B+ Tree

CHEN Genshen^{1,2,3,4,5}, LIU Gang^{1,2,3,4,5}, DONG Yang^{3,4}, FAN Wenyao^{1,2,4,5}, YI Qiang^{1,2} and JIANG Zixin^{3,4}

1 School of Computer Science, China University of Geosciences, Wuhan 430074, China

2 Hubei Key Laboratory of Intelligent Geo-Information Processing, Wuhan 430074, China

3 Wuhan Dida Quany Science and Technology Co., Ltd., Wuhan 430074, China

4 Guizhou Key Laboratory for Strategic Mineral Intelligent Exploration, Guiyang 550081, China

5 Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources, Shenzhen, Guangdong 518000, China

Abstract The prevalence of zero or null values in three-dimensional geological block models raises maintenance costs and efficiency issues due to frequent splitting and adjustment of the B+ tree-based attribute index structure. An indexing method based on Inverted-B+ Tree (IBT) is proposed. This method minimizes structural adjustments during data processing by constructing an IBT index structure, creating inverted nodes for duplicate keys inserted into leaf nodes. It accelerates queries by storing intermediate index values in internal nodes and establishing bidirectional links between leaves and inverted nodes, enabling efficient range queries via sequential access to the dataset from any leaves nodes. Six geological block models after voxelization, interpolation, and dimension reduction of geological structural models are mainly used in the experiment. Comparing the traditional B+ tree, results show that IBT has great performance in terms of index construction, spatial usage, and querying efficiencies. Especially for managing large amounts of data, index construction, and information query efficiency are improved by 71%, with 83% spatial usage

到稿日期:2024-07-22 返修日期:2024-09-06

基金项目:国家自然科学基金(42372345);自然资源部城市国土资源监测与仿真重点实验室开放基金(KF-2023-08-25);贵州磷、锰、铝优势资源成矿规律与快速高效智慧化勘查技术研究及示范项目(黔科合战略找矿[2022]ZD003, [2022]ZD004)

This work was supported by the National Natural Science Foundation of China(42372345), Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources(KF-2023-08-25) and Guizhou Strategic Prospecting Program([2022] ZD003 and [2022]ZD004).

通信作者:刘刚(liugang@cug.edu.cn)

being reduced, which is relatively stable and scalable for information queries of 3D geological blocks.

Keywords Inverted-B+ tree, Regular blocks, 3D geological models, Spatial data management, Spatial index

1 引言

近年来,随着地质信息技术^[1-2]和三维地质建模技术的进步^[3-4],构建三维地质模型成为研究地下矿产资源分布的重要手段^[5]。其中,基于规则块体的三维地质模型(块体模型)^[6-7]能够有效综合地质要素的属性信息,精确表达地质结构内部的非均质性^[8-10],为地下矿产资源空间分布的模拟与预测提供了新的思路和方法,图1给出了一个示例。通过对块体模型的属性值建立索引,可以快速精确地定位和查询具备特定属性的体元,进而获取包括矿产类型、品位、储量等在内的重要信息,这些信息对于开展复杂地质分析和资源精确评估极为关键^[11-12]。其中,B+树作为一种自平衡的树结构,被广泛应用于数据库和文件系统的索引。其优点在于,能够优化数据的顺序访问和减少查询路径长度,从而加快查询速度。通过为每个属性构建B+树索引,能够实现块体模型属性数据的快速访问和高效查询。然而,由于块体模型数据具有海量、多维和异质的特点^[13-14],基于B+树的块体模型属性数据索引依然面临挑战,具体如下。

1)块体模型中大量重复的零值或空值使得基于B+树的属性索引频繁分裂与合并,增加了索引维护的成本,导致索引性能下降和资源浪费。

2)在大规模块体模型中进行范围查询时,B+树的单向链表结构限制了从任意位置进行反向或双向查询的能力,需要对节点依次顺序遍历,导致难以高效查询特定属性范围内的地质块体。

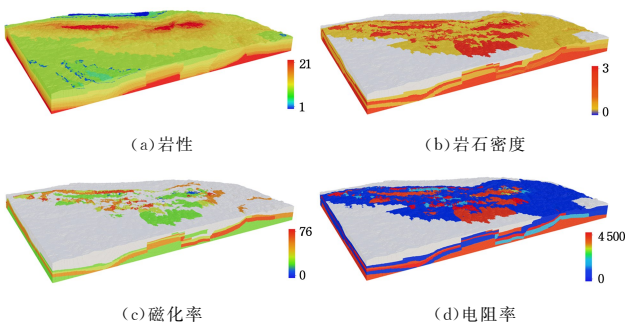


图1 三维地质块体模型示例

Fig.1 Examples of 3D geological block models

针对以上问题,本文提出一种基于Inverted-B+树(IBT)的三维地质块体模型属性索引方法。在B+树中引入倒排索引节点,通过聚集相同键值的数据指针,以减少B+树结构频繁分裂调整。在索引结构中建立了叶子节点与倒排节点间的双向链表,通过链表连接顺序访问整个数据集,以实现数据的高效顺序遍历与范围查询。此外,实现基于磁盘的IBT方法,并采用数据分页处理和缓存策略,以解决海量块体模型数据持久化的效率问题。通过实现单值查询和值域查询算法,以满足块体模型结构-属性的复杂联合查询的需求。最后,通过在6组不同规模和精度的块体模型数据集上进行验证,对

索引构建和数据查询的性能进行评估,验证了该方法的有效性和可扩展性。

2 相关工作

近年来,研究者提出了多种数据索引方法,包括基于空间分区、基于数据分区和基于映射的索引^[15]。

2.1 基于空间分区的索引

通过将空间对象转换成点、线、面或体等几何形状进行粗略索引^[16]。这种索引利用递归的方式将空间划分成更小的区块,直至每个区块内的空间对象数量适合索引节点管理。主要有两种策略^[17]:一种是采用分而治之的方法,如KD树、KDB树、BKD树、BSP树等;另一种是借鉴最小边界框的概念,将数据聚类到可能重叠的区域中,从而进行有效分区,常见的方法包括R树及其变种(R+树、R*树、Hilbert-R树、3D-R树),以及四叉树和八叉树等。这类索引及其变体在GIS、CAD和空间数据库等领域广泛应用^[18]。Cao等^[19]提出一种基于RHCA的Hilbert R树索引,用于地点数据的组织和查询。Cheng等^[20]将经过优化的希尔伯特R树与堆叠的LSTM模型相结合,实现了对大数据集的快速三维可视化,并保证了高质量渲染和流畅交互。Zhang等^[21]提出一种基于四叉树和R树的混合空间索引结构,以提高三维海洋场景中数据检索和渲染的效率。Liu等^[22]提出的HiIndex基于分块四叉树结构将全局地理范围递归划分,支持实时交互式可视化大规模矢量数据。Wang等^[23]提出一种面向GTM的混合空间索引结构3DOR*-树,将Octree的快速空间分区与3DR*-树对地质四面体模型的高效空间查询相结合。基于空间分区的索引方法优化了空间范围查询和最近邻搜索,但执行非空间属性查询时效率低下,因为这类数据不具备空间索引所依赖的地理特性,容易造成资源浪费。

2.2 基于数据分区的索引

基于数据分区的索引针对非空间属性的优化查询。其通过逻辑划分规则或物理存储分布,将数据组织到不同区域,以支持高效的检索、插入、更新和删除操作,常见结构有B树及其变体(B+树、B*树)和哈希索引^[24]等。Saface等^[25]提出了分布式B+树索引方法来处理流数据的范围查询,提高了可扩展性和效率。De Mol等^[26]提出间隔B+树(IBPT)索引技术用于处理可能性数值数据。该索引本质上是一种复合B+树,通过内部节点和叶子节点的特定结构设计,如叶子节点增加 P_{prev} 和 P'_{prev} 指针形成辅助双向链表,结合与选择原理,能够更高效地构建预选择集,从而实现模糊查询。这类索引能提高频繁执行等值查询、范围查询和排序操作的非空间数据的查询效率,但对空间数据的查询,如地理位置搜索和空间关系判断,效果通常不如专门设计的空间索引。

2.3 基于映射的索引

通过映射机制将多维空间数据转换为一维索引键值,利用空间填充曲线建立数据顺序,实现了平衡空间和非空间查

询性能。空间填充曲线如 Z 值曲线和希尔伯特曲线,将多维空间映射到一维,广泛应用于不同领域。Jiang 等^[27]提出了 Q-HBML 的多级网格索引结构,增强了基于空间局部的矢量多边形数据的存储和访问。Liu 等^[28]提出 HGST 时空网格划分和编码方法,以提高海量轨迹数据时空距离查询的效率。Lei 等^[29]引入了 W-Hilbert 曲线用于索引和管理大量地理数据,在多尺度环境下提供了改进的数据聚类能力。基于映射的索引结构能支持空间数据的高效查询,但对于多属性组合查询效率不高,因为它需要分别查询每个属性并合并结果,增加了计算复杂度和时间成本。

以上各类索引方法展示了在不同数据查询环境下的优势和局限。针对海量块体模型属性数据,基于 Inverted-B+ 树的索引方法整合了 B+ 树的数据分区效率和倒排索引的查询速度,能够提升空间与非空间属性的索引性能;通过倒排功能快速定位特定属性的块体,并利用 B+ 树支持快速更新,使其能够适应地质数据的动态变化,从而提升地质信息系统的数据处理精度和应用效率。

3 IBT 索引结构在地质块体中的应用

地质块体模型是表达地球物质分布和结构特征的有效途径,IBT 可为其属性数据提供一种高效的索引结构。

3.1 IBT 的索引结构与基本原理

IBT 通过引入倒排索引节点,有效减少了由于大量重复数据造成的 B+ 树结构频繁调整的问题。IBT 索引结构如图 2 所示,结构总体上分为 B+ 树结构和倒排索引结构两个部分。在 B+ 树结构中,节点分为内部节点(包含根节点)和叶子节点。内部节点存储中间索引值以加速查询过程,叶子节点存储插入的实际数据,通过双向指针相连,形成有序链表。在插入重复键至叶子节点时,为每个重复 Key 创建一个倒排节点,并将新数据插入其中。基于磁盘的 IBT 索引设计有效减轻了地质块体属性数据存储和索引构建的负担,提高了数据查询的效率。

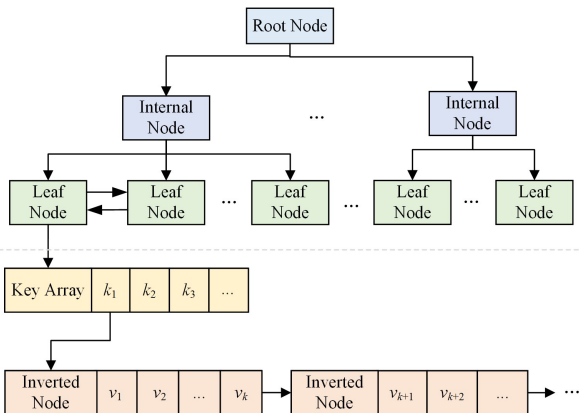


图 2 IBT 索引结构

Fig. 2 Structure of inverted B+ tree index

IBT 的设计中,内部节点主要用于加速查询,采用精简的存储策略。如图 3(a) 所示的内部节点,其页面起始处存储 2 字节标识节点类型和 4 字节记录键值对数量,其余部分存放 Key 及其子节点指针。叶子节点则作为 B+ 树与 Inverted 索

引的连接点,在页面起始处标识节点类型,后续 8 字节存储左右兄弟节点指针以支持双向链表,再记录 Key 数量,随后依次存放 Key 值、Value 和倒排索引头节点指针,如图 3(b) 所示。倒排节点用于存储所有相应 Key 的 Value,其结构如图 3(c) 所示,在页面起始处依次标识节点类型、左右兄弟节点的指针、Value 数量和所有 Value 数据,以支持快速遍历。

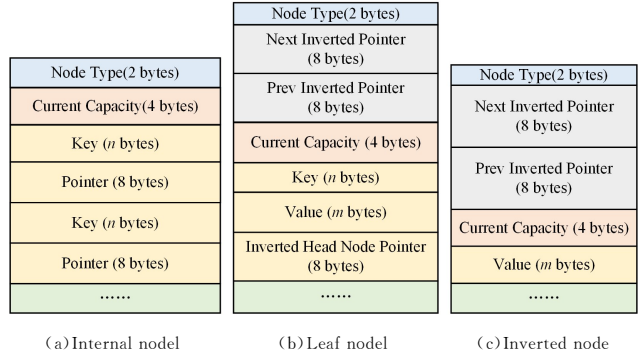


图 3 IBT 索引节点

Fig. 3 Nodes of inverted-B+ tree index

3.2 针对地质块体的 IBT 构建优化

IBT 方法可以通过分页存储管理和缓存预加载策略,在利用少量索引构建成本的前提下,显著提升属性数据的读取效率,并降低属性数据处理时的内存占用。这些技术特性使得海量地质块体模型的单机高效处理成为可能。

IBT 索引构建的核心是插入算法,如图 4 所示,常规 IBT 索引的插入需要频繁采用递归遍历,这将成为 IBT 索引构建的瓶颈。但是在海量块体模型属性数据中,连续型数据(如矿区伴生元素)通常严格依附于钻孔的空间位置分布,而离散型数据(如地层数据)多呈现层级化结构特征。由于两类数据均具有显著的空间局部性特征,因此可以通过叶子节点缓存等方式,快速构建 IBT 索引。在 IBT 的倒排索引节点中,关键字 Key 的插入通过单向链表的倒排节点进行,这些节点主要存储值(Value)。为提高插入效率,采用头插法插入数据和创建新的倒排节点。如果关键字 Key 与某个叶子节点值 k_1 相同,则将 Value 插入 k_1 的倒排索引中。若倒排索引的头节点已满,则通过头插法创建新的倒排节点作为新的头节点,并在此新节点中插入值 Value。

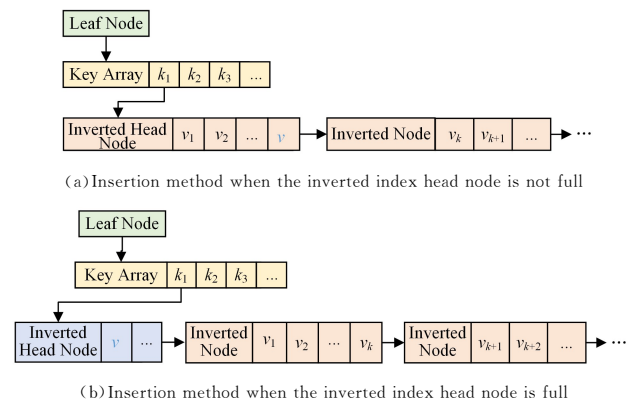


图 4 IBT 的插入算法

Fig. 4 Insertion algorithm of inverted-B+ tree

3.3 基于 IBT 的地质块体查询方法

IBT 支持单值检索和值域查询两种方式,这在地质块体模型的数据管理中极为关键。单值检索从根节点开始,通过二分搜索递归定位到目标叶子节点,如算法 1 所示。如果在叶子节点找到匹配的关键字 Key,则其对应的值和倒排索引节点中的所有值将被汇总到结果集中。如果当前节点为非叶子节点,那么查询将继续递归到对应的子节点 childNode 进行。

算法 1 Search Key

Input: The node to be queried and the key for the search

Output: return the set of query results

```

1. i ← binarySearch(key, node)
2. result ← newEmptyList()
3. if node is a leaf node, then
4.   if getKey(i) == key then
5.     add the value at position i to the result
6.   if the inverted head node at position i is not empty then
7.     invertedNode ← getInvertedHeadNode(node, i)
8.     do
9.       add all values from invertedNode to the result
10.      invertedNode ← getNextInvertedNode(invertedNode)
11.    while invertedNode is not empty
12. else
13.  childNode ← getChildNode(n, i)
14.  return searchKey(childNode, k)
15. return result

```

在 IBT 中,叶子节点通过双向链表连接,便于执行值域查询。如算法 2 所示,先用单值检索算法确定查询范围的下界位置。如果下界存在,则从该节点向右遍历至上界;若下界不存在,则从上界的节点向左遍历。过程中,若遇到倒排索引节点,将继续遍历这些节点以完善查询结果。

算法 2 Range Search

Input: The upper and lower bounds of the query range, maxKey and minKey, along with the inverted B+Tree object to be queried

Output: return the set of query results

```

1. root ← getRoot(tree)
2. result ← newEmptyList()
3. minleaf, minI ← searchKey(minKey, root)
4. if leaf node is not empty then
5.   while getKey(minleaf, minI) ≤ maxKey do
6.     add the value at position i to result
7.   if the inverted head node at position i is not empty then
8.     traverse all inverted nodes and add all values within the
       nodes to result
9.   minI ← minI + 1
10.  if minI == getCurrentCapacity(minleaf)
11.    minleaf ← getNextLeafNode(minleaf)
12.    minI ← 0
13. else
14.  maxleaf, maxI ← searchKey(maxKey, root)
15.  while getKey(maxleaf, maxI) ≥ minKey do

```

```

16.   add the value at position i to result
17.   if the inverted head node at position i is not empty then
18.     traverse all inverted nodes and add all values within the
       nodes to result
19.   maxI ← maxI - 1
20.   if maxI < 0
21.     maxleaf ← getPrevLeafNode(maxleaf)
22.     maxI ← getCurrentCapacity(maxleaf)
23. return results

```

4 实验与评估

实验代码通过 Java 实现;编译环境为 jdk1.8;运行环境为处理器 Intel^(R) Xeon^(R) W-2133 CPU 3.60 GHz, 12 核, RAM 64GB 和 NVIDIA Quadro P6000 GPU;操作系统为 Windows 10。

为了验证所提出方法的有效性和性能,将其与传统 B+树算法进行对比实验,从索引构建时间和空间、单值查询时间和值域查询时间 4 个方面,验证 IBT 方法的可行性和先进性。通过对西南某研究区的 1:25000 三维地质结构模型进行体元剖分、插值和空间降维处理,得到 6 个不同尺度的规则块体数据集,实验数据如表 1 所列。数据包括离散型属性(如地层和断层,代表地质结构的分类信息)和连续型属性(如 Pb, Ts 和 Ag 等矿物元素的品位)。

表 1 实验数据

Table 1 Experimental data

Datasets	Data scale	Valid blocks	File size/kB
D32	32×32×32	16406	673
D64	64×64×64	126067	5377
D128	128×128×128	983021	43009
D256	256×256×256	7743278	344065
D512	512×512×512	61452258	2752513
D1024	1024×1024×1024	489555817	22020097

4.1 索引构建性能评估

实验目的是比较 IBT 与 B+树在索引构建性能上的差异,涵盖构建时间和索引大小。实验在相同硬件条件下,使用 D32 至 D1024 不同规模的测试数据,针对 Stratum, Fault, Pb, TS 和 Ag 等属性进行索引构建,以检验方法在处理离散和连续属性数据上的效能。性能比较采用比率或差值,通过比率式(1)来展示 IBT 相对于 B+树的构建性能优势。

$$ratio = \frac{(P_{B+} - P_{Inverted-B+})}{P_{B+}} \quad (1)$$

图 5 展示了两种方法在不同数据规模下的索引构建时间对比。在较小数据规模(如 D32, D64, D128)下,两种方法的构建时间较短,但随着数据规模的增加,索引构建时间均明显增长。在所有测试数据规模下,IBT 的索引构建时间均短于 B+树,平均提升率为 71%,结果显示,IBT 在处理大规模数据时具有更高的效率。

图 6 展示了 IBT 与 B+树在不同数据规模下的空间占用差异。IBT 在所有数据规模下的空间占用均显著小于

B+树,特别是在大数据规模下差异更明显。尽管这两种方法的空间占用随数据规模增加而上升,但IBT的增长速度较慢。如图6(f)所示,IBT的空间占用提升率约稳定在83%,表明其在节省空间方面具有明显优势。IBT通过在倒排节点中单独存储重复Key对应的Value,有效减少了节点分裂,显著提升了索引构建和查询效率。此设计使得

B+树节点中仅存储唯一的Key,减少了存储空间需求。在属性索引实验中,地层和断裂等离散型属性因重复Key多而表现出高效率;尽管连续型属性如Pb, Ts和Ag的Key重复率较低,效率及空间利用提升不如离散型,但总体而言,IBT在性能上相比B+树有显著优势,如图5(f)和图6(f)所示。

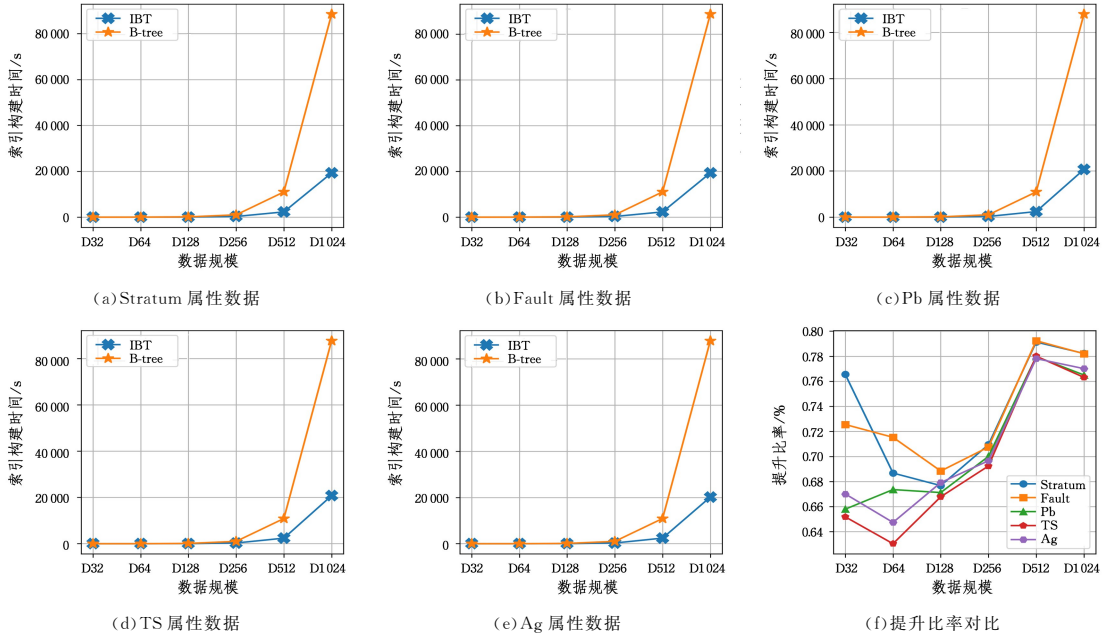


图5 索引构建时间对比

Fig. 5 Comparison of index construction time

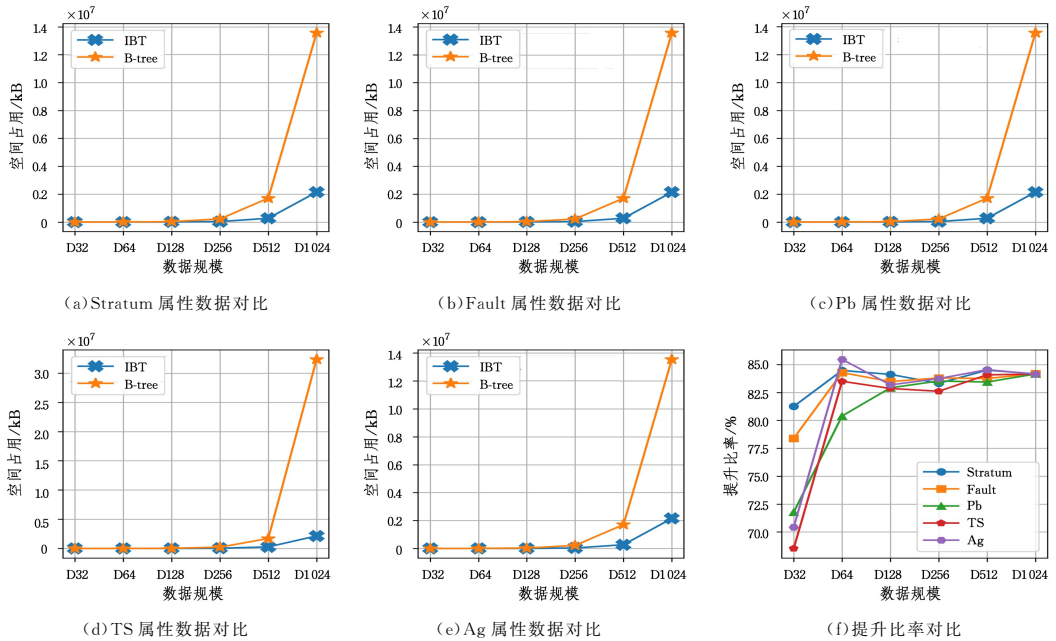


图6 索引构建空间占用对比

Fig. 6 Comparison of space occupation in index construction

4.2 单值查询算法性能评估

为每个属性生成了1000个随机数来构建查询集,并记录查询时间的均值。为避免浮点数类型的数据在进行单值精确查询时存在精度匹配问题,采用了模糊查询方法。图7显示了IBT与B+树在单值查询性能上的比较结果,其中IBT

在所有数据规模上都展现出了性能优势。特别是随着数据集规模的增大,IBT的性能扩展性更为显著,查询响应时间的增长较B+树更缓慢,这在图7(a)~图7(e)中有直观的体现,并在图7(f)中的性能改善中位数统计中得到了进一步证实。

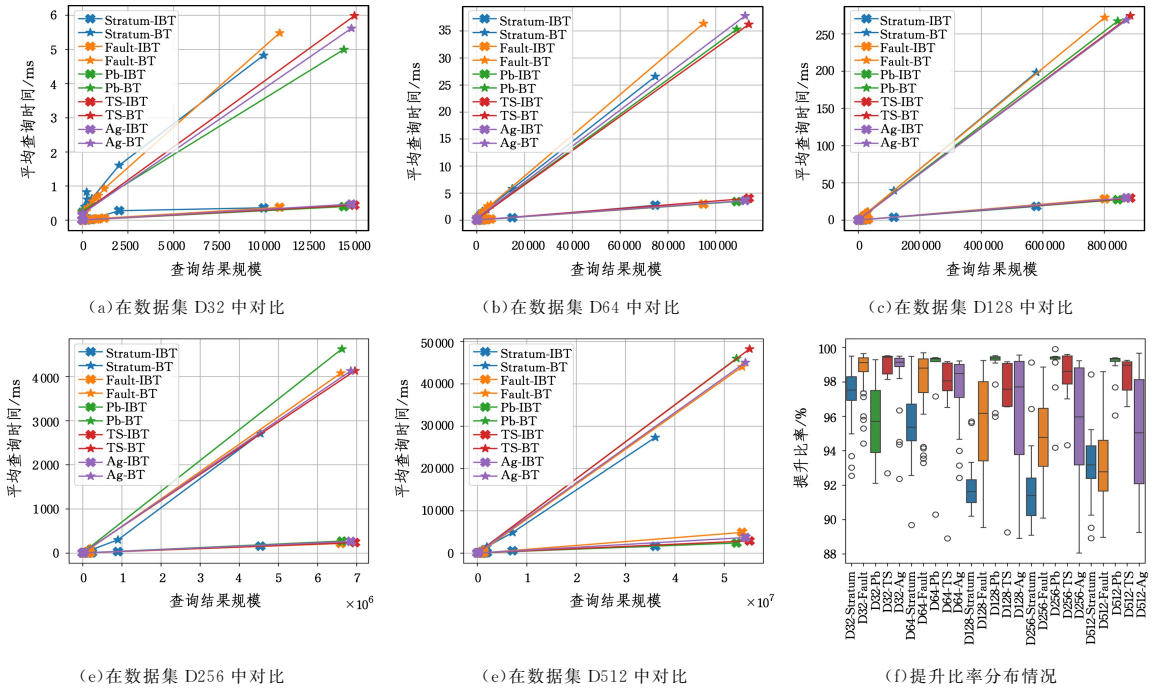


图 7 单值索引性能对比

Fig. 7 Performance comparison of single-value query

4.3 值域查询算法性能评估

图 8 对比了 IBT 和传统 B+树处理不同规模数据的值域查询性能。随着数据规模的增大,查询时间的波动和异常值增多,这一点在连续型属性查询中更为明显,这主要是数据密度不均匀分布和查询区间宽度的不一致性导致的。例如,在

更大的数据规模下,相同比例的查询区间覆盖的数据条目更多,导致查询时间增长。图 8(f)展示了不同数据规模下的值域查询性能提升比率分布,曲线的峰值和宽度分别表示最常见的性能提升比率和提升的波动程度,曲线越窄意味着性能提升越稳定。

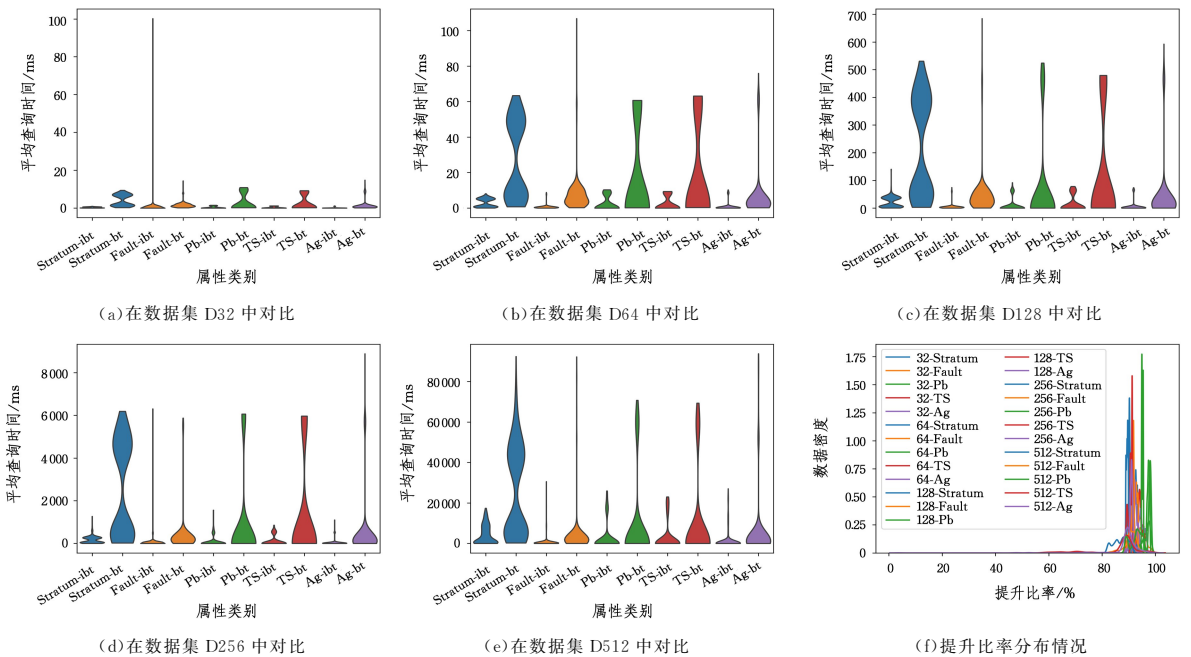


图 8 值域查询性能对比

Fig. 8 Performance comparison of range-value queries

5 应用案例

案例 1 地层索引

以某矿区地层属性为例,该矿区地层形成过程只受少量

地质构造作用影响。由于地层多为层状结构(见图 9),在索引构建时,能够按照地层由新到老的顺序自上而下遍历块体模型快速构建 IBT 索引,实现该块体模型地层属性的分页存储。按照顺序存储的数据更为集中,极大地缩短了块体模型

数据加载和渲染的时间。

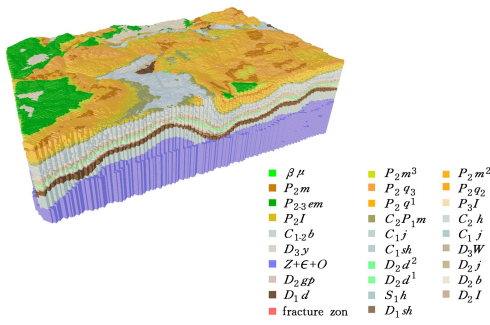


图9 研究区地层数据加载与渲染

Fig. 9 Loading and rendering of stratigraphic data in the research area

当需要显示某地层数据时,通过索引结构快速定位磁盘位置,快速读入该地层数据,图 10 展示了地层代号为 P_{2-3em} 的地层块体。IBT 索引支持该交互过程能快速加载所需数据,从而减小单机工作负载。

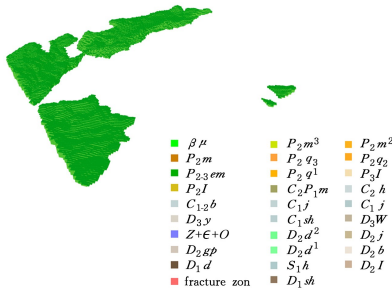
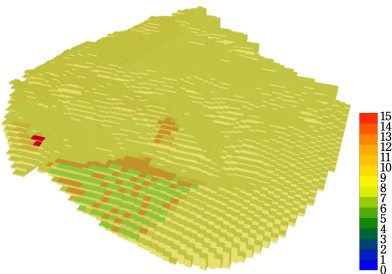


图 10 单个地层数据加载与渲染

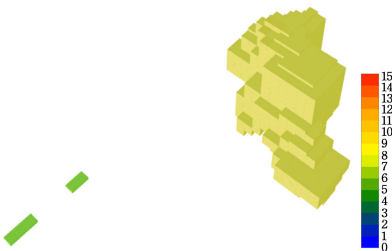
Fig. 10 Single stratum data loading and rendering

案例 2 属性索引

以某矿区矿体的 TS 元素品位查询为例,在构建 TS 属性数据索引过程中,获得该元素的品位区间。查询研究区的 TS 元素品位区间(6,15)的结果如图 11(a)所示。



(a) 矿体 TS 品位值区间为(6,15)的块体



(b) 地层中矿体 TS 品位值区间为(6,15)的块体

图 11 属性查询

Fig. 11 Attributes queries

地层是矿体依附的基本单元,在矿体形成过程中,与地层体的空间关系呈现多种形态。而在矿产开发过程中,设计掘进路线需要根据地层的岩性和走向等信息,预测矿体的空间分布情况并科学规划开挖方量。因此,根据地层的岩性信息与矿体品位信息查询区块的空间位置对于制定科学的掘进计划至关重要。图 11(b)展示了地层代号为 D_1d 的地层中 TS 品位区间为(6,15)的块体信息。

结束语 本文提出了一种基于 IBT 的三维地质块体模型高效索引方法,实现了海量多维异质块体模型数据的高效检索。主要贡献如下:

1)提出基于 IBT 结构的三维地质块体模型属性数据索引方法,通过结合 B+树的有序性与高效查询,以及倒排索引处理重复数据的优势,避免了构建块体属性数据索引结构频繁调整问题,实现了块体模型属性数据索引的性能优化。

2)针对连续型属性数据带来的查询粒度问题,通过区间划分将连续型数据进行离散化,并在索引结构中建立叶子节点与倒排节点间的双向链表,每个叶子节点存储数据的键值及对应的块体指针,倒排节点存储指向相同数据值的所有块体的指针。这使得从任一叶子节点出发,可以通过链表链接顺序访问整个属性数据集,实现了块体连续性属性数据的高效顺序遍历与精确范围查询。

3)通过实现基于磁盘的 IBT 索引结构,并利用缓存池技术和数据分页策略,将索引节点视为独立的磁盘页面进行处理,大幅降低了磁盘访问次数。通过属性类别构建的 IBT 索引,相当于将大规模的三维地质块体模型数据进行拆分,增加了数据访问与数据更新的灵活性,从而能够提高大规模块体模型数据的持久化效率。

实验结果验证了 IBT 在索引构建时间、空间占用、单值查询和值域查询性能上的优势。特别是在大规模数据集处理中,IBT 提高了 71%的效率并减少了约 83%的空间需求。IBT 在实际应用案例中展现了其在三维地质块体模型属性数据管理中具备一定的可扩展性和稳定性。

参考文献

[1] WU C L, LIU G, ZHOU Q, et al. Fundamental problems of integrated application of big data in geoscience[J]. Bulletin of Geological Science and Technology, 2020, 39(4): 1-11.

[2] LIU G, WU C L, HE Z W, et al. Data model for geological spatio-temporal big data expression and storage management[J]. Bulletin of Geological Science and Technology, 2020, 39(1), 164-174.

[3] ZHOU Q, WU C L, HU X Y, et al. A new metallogenic model for the giant manganese deposits in northeastern Guizhou, China [J]. Ore Geology Reviews, 2022, 149: 105070.

[4] CHEN G S, GUO X L, LIU G, et al. 3D geological modeling in the typical area of karst basin on the south bank of the Yangtze river in Yichang city [J]. Safety and Environmental Engineering, 2019, 26(2): 1-8.

[5] LIU G, CHEN G S, CHEN Q Y, et al. Key technologies for efficient visualization of large scale 3D spatial models based on web [J]. Geospatial Information, 2023, 21(12): 8-13.

[6] LIU L, SUN Y, JI M, et al. Efficient Construction of Voxel

- Models for Ore Bodies Using an Improved Winding Number Algorithm and CUDA Parallel Computing[J]. *ISPRS International Journal of Geo-Information*, 2023, 12(12): 473.
- [7] LI J, LIU P R, LIANG Z X, et al. Three-dimensional geological modeling method of regular voxel splitting based on multi-source data fusion[J]. *Rock and Soil Mechanics*, 2021, 42(4): 1170-1177.
- [8] SMIRNOVA A D, CHEN S J, MIKHAYLOVA T V. Geological Mathematical Block Modelling in Kuzbass Mining Industry [C]//2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). IEEE, 2022: 1970-1973.
- [9] CUI Z S, CHEN Q Y, LIU G. A two-stage downscaling hydrological modeling approach via convolutional conditional neural process and geostatistical bias correction[J]. *Journal of Hydrology*, 2023, 620: 129498.
- [10] WANG Z Y, ZUO R G, YANG F F. Geological mapping using direct sampling and a convolutional neural network based on geochemical survey data[J]. *Mathematical Geosciences*, 2023, 55(7): 1035-1058.
- [11] JIN X, WANG G W, TANG P, et al. 3D geological modeling and uncertainty analysis for 3D targeting in Shanggong gold deposit (China) [J]. *Journal of Geochemical Exploration*, 2020, 210: 106442.
- [12] WELLMANN F, CAUMON G. 3-D Structural geological models: Concepts, methods, and uncertainties [M]//Advances in geophysics. Elsevier, 2018: 1-121.
- [13] LIU Y Y, ZHANG X W, GUO W, et al. Research Status of and Trends in 3D Geological Property Modeling Methods: A Review [J]. *Applied Sciences*, 2022, 12(11): 5648.
- [14] HOULDING S. 3D geoscience modeling: computer techniques for geological characterization[M]. Springer Science & Business Media, 2012.
- [15] LI M X, WANG H C, DAI H P, et al. A Survey of Multi-Dimensional Indexes: Past and Future Trends[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2024, 36(8): 3635-3655.
- [16] MAMOULIS N. Spatial data management[M]. Beijing: Springer Nature, 2022.
- [17] PAN J J, WANG J, LI G. Survey of vector database management systems[J]. *The VLDB Journal*, 2024, 33(5): 1591-1615.
- [18] CHEN G S, LIU G, CHEN Q, et al. Key technologies and application of 3D spatial data management system for mining based on multi-source data integration and WebGIS [J]. *Software Guide*, 2023, 22(11): 18-28.
- [19] CAO K Y, LIU Y F, DONG Y K, et al. Hilbert R-tree Space Indexing Based on RHCA Clustering[C]// International Conference on Smart Technologies and Systems for Internet of Things. Singapore: Springer Nature Singapore, 2021: 205-211.
- [20] CHENG H, XIE K, WEN C, et al. Fast visualization of 3D massive data based on improved Hilbert R-tree and stacked LSTM models[J]. *IEEE Access*, 2021, 9: 16266-16278.
- [21] ZHANG Y N, ZHANG A M, GAO M, et al. Research on Three-Dimensional Electronic Navigation Chart Hybrid Spatial Index Structure Based on Quadtree and R-Tree[J]. *ISPRS International Journal of Geo-Information*, 2022, 11(5): 319.
- [22] LIU Z B, CHEN L, YANG A R, et al. Hiindex: An efficient spatial index for rapid visualization of large-scale geographic vector data[J]. *ISPRS International Journal of Geo-Information*, 2021, 10(10): 647.
- [23] WANG Y Z, LYU H, MA Y Q. Geological tetrahedral model-oriented hybrid spatial indexing structure based on Octree and 3D R*-tree[J]. *Arabian Journal of Geosciences*, 2020, 13(15): 728.
- [24] CHAUDHRY N, YOUSAF M M. A hash-based index for processing frequent updates and continuous location-based range queries[J]. *Knowledge and Information Systems*, 2023, 65(10): 4233-4271.
- [25] SAFACE S, MIRABI M, RAHMANI A M, et al. A distributed B+ Tree indexing method for processing range queries over streaming data[J]. *Cluster Computing*, 2023, 27: 1251-1274.
- [26] DE MOL R, BARRANCO C D, DE TRÉ G. Indexing possibilistic numerical data using interval B+-trees[J]. *Fuzzy Sets and Systems*, 2021, 413: 138-154.
- [27] JIANG H, KANG J F, DU Z H, et al. Vector spatial big data storage and optimized query based on the multi-level Hilbert grid index in HBase[J]. *Information*, 2018, 9(5): 116.
- [28] LIU H, YAN J N, WANG J L, et al. HGST: A Hilbert-GeoSOT Spatio-Temporal Meshing and Coding Method for Efficient Spatio-Temporal Range Query on Massive Trajectory Data[J]. *ISPRS International Journal of Geo-Information*, 2023, 12(3): 113.
- [29] LEI Y, TONG X C, WANG D L, et al. W-Hilbert: A W-shaped Hilbert curve and coding method for multiscale geospatial data index[J]. *International Journal of Applied Earth Observation and Geoinformation*, 2023, 118: 103298.



CHEN Genshen, born in 1993, Ph.D, is a member of CCF(No. J5928G). His main research interests include geoscience big data, GIS, and geo-information visualization.



LIU Gang, born in 1967, Ph.D, professor, Ph.D supervisor, is a member of CCF(No. 24497S). His main research interests include geoscience big data, intelligent geoinformation processing, and visualization and visual analysis.